

Proceedings of ICCM 2023

21st International Conference on Cognitive Modelling¹

Edited by

Catherine Sibert

¹ Co-located with the 56th Annual Meeting of the Society for Mathematical Psychology

Preface

The International Conference on Cognitive Modelling (ICCM) is the premier conference for research on computational models and computation-based theories of human cognition. ICCM is a forum for presenting and discussing the complete spectrum of cognitive modelling approaches, including connectionism, symbolic modeling, dynamical systems, Bayesian modeling, and cognitive architectures. Research topics can range from low-level perception to high-level reasoning. In 2023, ICCM was jointly held with MathPsych – the annual meeting of the Society for Mathematical Psychology. The conference was held at the University of Amsterdam from July 18th to July 21st. An additional, virtual conference was held online from June 19th to June 23rd. Submissions from both the in-person and virtual conferences are included in these proceedings.

Acknowledgements

We would like to thank the Society for Mathematical Psychology (SMP) for their ongoing commitment to the collaboration between our societies. In particular, the MathPsych Conference Chair (Joachim Vandekerckhove) maintained the conference website, and the officers of the SMP (especially Leslie Blaha) gave much needed logistical support.

Papers in this volume may be cited as:

Lastname, A., Lastname, B., & Lastname, C. (2023). Title of the paper. In Sibert, C. (Ed.). *Proceedings of the 21st International Conference on Cognitive Modelling* (pp. 6-12). University Park, PA: Applied Cognitive Science Lab, Penn State.

ISBN-13: 978-0-9985082-7-6, published by the Applied Cognitive Science Lab, Penn State.

(C) Copyright 2023 retained by the authors

Conference Committees

General and Program Chairs

Catherine Sibert	University of Groningen
Marieke van Vugt	University of Groningen

Program Committee

Madeleine Bartlett	University of Waterloo
Leslie Blaha	Air Force Research Laboratory
Michael Byrne	Rice University
Yiyang Chen	University of Kansas
Edward Cranford	Carnegie Mellon University
Taylor Curley	Air Force Research Laboratory
Chris Dancy	Pennsylvania State University
Anna Geuzebroek	University College Dublin
Wouter Kruijnen	University of Groningen
Othalia Larue	Wright State University
Christian Lebiere	Carnegie Mellon University
Can Serif Mekik	University of Toronto
Junya Morita	Shizuoka University
Bruno Nicenboim	University of Tilberg
David Peebles	University of Huddersfield
Stefan T. Radev	Rensselaer Polytechnic Institute
Sarah Ricupero	Pennsylvania State University
Frank Ritter	Pennsylvania State University
Sönke Steffen	University of Groningen
Terrence C. Stewart	National Research Council Canada
Andrea Stocco	University of Washington
Anastasia Stoop	University of Illinois/UC
Farnaz Tehrani	Pennsylvania State University
James Treyens	University of Washington
Robert L. West	Carleton University
Arkady Zgonnikov	TU Delft

Table of Contents

Resource demands of an implementationist approach to cognition.....1	
<i>Federico Adolfi, Iris van Rooij</i>	
Expressing psychological distance using Sharma-Mittal divergence.....10	
<i>Mikaela Akrenius</i>	
From knowledge graph to cognitive model: A method for identifying task skills.....17	
<i>Ivana D.M. Akrum, Niels A. Taatgen</i>	
A pipeline for analyzing decision-making processes in a binary choice task.....24	
<i>Amirreza Bagherzadehkhosravi, Farnaz Tehrani</i>	
Cognitively and Linguistically Motivated Part of Speech Tagging: Quantitative Assessment of a Near Human-Scale Computational Cognitive Model.....31	
<i>Jerry T. Ball, Stuart M. Rodgers</i>	
Role Stability and Team Performance in a 4-Player Cooperative Cooking Game.....38	
<i>Sounak Banerjee, Wayne D. Gray</i>	
Improving Reinforcement Learning with Biologically Motivated Continuous State Representations.....44	
<i>Madeleine Bartlett, Kathryn Simone, Nicole Sandra-Yaffa Dumont, P. Michael Furlong, Chris Eliasmith, Jeff Orchard, Terrence C. Stewart</i>	
Quantifying performance in magnitude comparison tasks using a drift-diffusion model.....51	
<i>Mark Bensilum, Richard P. Cooper</i>	
Uncovering iconic patterns of syllogistic reasoning: A clustering analysis.....57	
<i>Daniel Brand, Nicolas Riesterer, Marco Ragni</i>	
Modeling Change Points and Performance Variability in Large-Scale Naturalistic Data.....64	
<i>Michael Collins, Florian Sense, Michael Krusmark, Tiffany Jasstrzemski</i>	
Metacognitive threshold: a computational account.....70	
<i>Brendan Conway-Smith, Robert L. West</i>	
Extending counterfactual reasoning models to capture unconstrained social explanations...76	
<i>Stephanie Droop, Neil Bramley</i>	

Modeling A Human-AI Cooperation Task In ACT-R.....	86
<i>Tanishca Sanjay Dwivedi, Christopher L. Dancy</i>	
Comparing Classical and Quantum Probability Accounts of the Interference Effect in Decision Making.....	88
<i>Christopher R. Fisher, Lorraine Borghetti, Joseph W. Houpt, Christopher Stevens, Leslie M. Blaha</i>	
Using neural networks to create fast and reusable approximate likelihood functions for ACT-R.....	95
<i>Christopher R. Fisher, Taylor Curley, Christopher Stevens</i>	
Tetrad Fit Index for Factor Analysis Models.....	101
<i>Vithor Rosa Franco, Rafael Valdece Sousa Bastos, Marcos Jiménez</i>	
Single Neuron Distribution Modelling for Anomaly Detection and Evidence Integration..	107
<i>P. Michael Furlong, Madeleine Bartlett, Terrence C. Stewart, Chris Eliasmith</i>	
Illuminating Individual Learning Dynamics Within a Task: A Computational Model Analysis.....	114
<i>Theodros Haile, Chantel Prat, Andrea Stocco</i>	
An ACT-R Observer Model for Anticipatory Assistive Robots.....	121
<i>Chenxu Hao, Colin Halupczok, Winfried Ilg, Daniel Haeufle, Phillipp Beckerle, Nele Russwinkel</i>	
GPT-Jass : A Text-to-model Pipeline for ACT-R Models.....	126
<i>Anthony M. Harrison, Laura M. Hiatt, Greg Trafton</i>	
Cognitive Modelling of Intention Recognition in Cocktail Mixing.....	128
<i>Linda Heimisch, Janice Jansen, Nele Russwinkel</i>	
A Straightforward Implementation of Sensorimotor Abstraction in a Two-Layer Architecture for Dynamic Decision-Making.....	135
<i>Nils Heinrich, Annika Österdiekhoff, Stefan Knopp, Nele Russwinkel</i>	
Using cognitive models to test interventions against mind-wandering during driving.....	140
<i>Moritz Helf, Andreea Minculescu, Jochem Rieger, Jelmer Borst</i>	
An Initial Cognitive Model of a Radar Detection Task.....	143
<i>Alexander R. Hough, Christopher Stevens, Elizabeth Fox, Christopher Myers</i>	
Integrated Cognitive Model Framework for Analogical Reasoning.....	150
<i>Alexander R. Hough, Othalia Larue, Christopher Myers, Olivia Leung</i>	
Seeing What You Believe: Cognitive Mechanisms of Flexible Integration of Priors in Visual Decisions.....	153
<i>Gabriela Iwama, Randolph Helfrich</i>	

Predicting Human Interleaving Time in Semi-Automated Vehicles.....	156
<i>Christian P. Janssen, Leonard Praetorius, Jelmer P. Borst</i>	
Modelling the role of Hanja in the Korean mental lexicon: A second tier of spreading activation.....	158
<i>Stephen Jones, Yoolim Kim</i>	
Errors Are The Stepping Stones to Learning: Trial-by-Trial Modeling Reveals Overwhelming Evidence for Mediator Retrievals of Previous Errors in Memory Consolidation.....	160
<i>Bridget Leonard, Andrea Stocco</i>	
Cognitive modeling of category learning and reversal learning.....	167
<i>Marcel Lommerzheim, André Brechmann, Nele Russwinkel</i>	
An integrative model of human response processes in Raven's Matrices.....	174
<i>Can Serif Mekik, Ron Sun, David Yun Dai</i>	
ACT-R Modeling of Rapid Motor Learning Based on Schema Construction.....	180
<i>Kazuma Nagashima, Jumpei Nishikawa, Ryo Yoneda, Junya Morita</i>	
The CoFI Reader: A Continuous Flow of Information approach to modeling reading.....	187
<i>Bruno Nicenboim</i>	
Improving Visuomotor Control of a Cognitive Architecture.....	194
<i>Grace Roessling, Tim Halverson, Christopher Myers</i>	
A Cognitive Model of a Temporal Binding Task.....	201
<i>Laura Saaad, Alexander R. Hough, Leslie M. Blaha, Christian Lebiere</i>	
Moral Judgements as the Combination of Distributed Language Representation and Memory Activation Mechanism.....	208
<i>Kenya Sasaki, Jumpei Nishikawa, Kazuma Nagashima, Junya Morita</i>	
Generating body images from distributed word representation.....	210
<i>Kosuke Sasaki, Jumpei Nishikawa, Junya Morita</i>	
Relative Attention Across Features Predicts that Common Features Increase Geometric Similarity.....	217
<i>Florian I. Seitz</i>	
Comparing Model Variants Across Experimental and Naturalistic Data Sets.....	223
<i>Florian Sense, Michael Collins, Michael Krusmark, Tiffany Jastrzembski</i>	
Estimating Individual Property in a Simple Memory Task.....	230
<i>Kohei Simbori, Jumpei Nishikawa, Kazuma Nagashima, Junya Morita</i>	

Leveraging Large-Scale Brain Connectivity Data to Explore and Expand the Common Model of Cognition.....	232
<i>Sönke Steffen, Catherine Sibert</i>	
Novelty Detection, Insect Olfaction, Mismatch Negativity, and the Representation of Probability in the Brain.....	233
<i>Terrence C. Stewart, P. Michael Furlong, Kathryn Simone, Madeleine Bartlett, Jeff Orchard</i>	
Efficient Memory Encoding Explains the Interactions Between Hippocampus Size, Individual Experience, and Clinical Outcomes: A Computational Model.....	239
<i>Andrea Stocco, Briana M. Smith, Bridget Leonard, Holly Sue Hake</i>	
A Cognitive Model of the Effects of Workload on Perceptual Span.....	246
<i>Garrett Swan, Christopher A. Stevens, Samantha Klosterman</i>	
Preferred Mental Models in Syllogistic Reasoning.....	252
<i>Sara Todorovikj, Daniel Brand, Marco Ragni</i>	
Modelling the Effects of ACT-R Working Memory Demands on Accuracy Rates of Relational Reasoning Problems.....	259
<i>Nico V. Turcas, Jim Davies, Robert L. West</i>	
Alleviating 4 Million Cold Starts in Adaptive Fact Learning.....	266
<i>Maarten van der Velde, Florian Sense, Jelmer Borst, Hedderik van Rijn</i>	
A neural network simulation of event-related potentials in response to syntactic violations in second-language learning.....	268
<i>Stephan Verwijmeren, Stefan L. Frank, Hartmut Fitz, Yung Han Khoe</i>	
Exploring Errors Towards a More Realistic Strategy Model.....	275
<i>Shan N. Wang, Frank E. Ritter</i>	
Modeling a Strategy with Learning in a Complex Task.....	277
<i>Shan N. Wang, Frank E. Ritter</i>	
Cognitive and Meta-cognitive Signatures of Memory Retrieval Performance in Spoken Word Learning.....	279
<i>Thomas Wilschut, Florian Sense, Hedderik van Rijn</i>	
Long Road Ahead: Lessons Learned from the (soon to be) Longest Running Cognitive Model.....	281
<i>Siyu Wu, Amir Bagherzadeh, Frank E. Ritter, Farnaz Tehranchi</i>	
The Cognitive Substrates of Model-Based Learning: An Integrative Declarative-Procedural Model.....	288
<i>Yuxue C. Yang, Andrea Stocco</i>	

A diffusion model decomposition of the unit-decade compatibility effect in two-digit number comparison.....	295
<i>Bella E. Zapata, Thomas J. Faulkenberry</i>	

Resource Demands of an Implementationist Approach to Cognition

Federico Adolfi (fedeadolfi@bristol.co.uk)

School of Psychological Science, University of Bristol, UK
Ernst Strüngmann Institute for Neuroscience in Cooperation with Max-Planck Society, Germany

Iris van Rooij (iris.vanrooij@donders.ru.nl)

Donders Institute for Brain, Cognition, and Behaviour, Radboud University, The Netherlands
School of Artificial Intelligence, Radboud University, The Netherlands
Department of Linguistics, Cognitive Science, and Semiotics & Interacting Minds Centre, Aarhus University, Denmark

Abstract

A core inferential problem in the study of natural and artificial systems is the following: given access to a neural network, a stimulus and behaviour of interest, and a method of systematic experimentation, figure out which circuit suffices to generate the behaviour in response to the stimulus. It is often assumed that the main obstacles to this “circuit cracking” are incomplete maps (e.g., connectomes), observability and perturbability. Here we show through complexity-theoretic proofs that even if all these and many other obstacles are removed, an intrinsic and irreducible *computational hardness* remains. While this may seem to leave open the possibility that the researcher may in practice resort to approximation, we prove the task is *inapproximable*. We discuss the implications of these findings for implementationist versus functionalist debates on how to approach the study of cognitive systems.

Keywords: Computational Complexity; Meta-theory; Neural Networks; Neuroscience; Artificial Intelligence (AI); Circuit Understanding; Implementationism; Functionalism

Introduction

Whether cognition is best studied from a functional or implementational perspective¹ is a longstanding debate that has defined and divided researchers in the cognitive sciences. The main conflict of intuitions is that some think we can make faster progress by taking a function-first approach (Egan, 2018; van Rooij & Baggio, 2021), whereas the implementation-first approach believes that cracking neural primitives is a necessary first step on the path to faster and better understanding (P. S. Churchland & Sejnowski, 1999). Function-first approaches are certainly no silver bullet and face intractability challenges (Rich, de Haan, Wareham, & van Rooij, 2021). Hence, the implementation-first approach may seem to have the high ground. It may thus seem wise to put all eggs in that implementation-first basket and to answer calls to create complete maps of the territory (e.g., connectomes) and build tools for full observability and full perturbability to probe how neural circuits support behaviors (cf. Bargmann & Marder, 2013). While it seems intuitive that scientific discovery is sure to be sped up by realizing these ideals in practice, in this paper we show this intuition is mistaken.

Our case study is the *circuit cracking* problem. Understanding how internal circuits support behavior in natural and

artificial systems is a central concern in the cognitive and brain sciences. Cracking a circuit — identifying which circuits² are involved in producing a target behavior — is regarded across disciplines as a basic facet of achieving such understanding (cf. Voss, Goh, et al., 2021; Voss, Cammarata, et al., 2021). It is often assumed that the main sources of complexity in circuit cracking have to do with data-analytic strategies, experimental design and control, statistical inference, observability, perturbability, and access to connectomes.

We study the computational resource demands of the circuit cracking problem using formal concepts and proof techniques from computational complexity theory (van Rooij, Blokpoel, Kwisthout, & Wareham, 2019; Garey & Johnson, 1979). We build on and extend the work of Ramaswamy (2019), who provided initial definitions and a proof sketch of computational hardness. We formalize and flesh out crucial details, and prove that the *circuit cracking* problem is not only intractable to solve exactly, but even to solve approximately. This holds even when all aforementioned assumed sources of complexity are removed. We explain how these proofs challenge the conviction that implementation-first approaches should be prioritized over function-first approaches.

The remainder of this paper is organized as follows. First we *situate* the circuit cracking problem in the fields of neuropsychology, neuroscience and artificial intelligence. Next we *formalize* the circuit cracking problem to make it amenable to computational complexity analysis. We then *present proofs* of hardness and inapproximability. Finally, we *discuss* the implications of these complexity-theoretic findings for meta-theoretical commitments and the allocation of research resources in the cognitive sciences.

Situating Circuit Cracking

Circuit cracking is closely related to problems which have been the focus of continued efforts in neuropsychology, neuroscience, artificial intelligence, and more recently in subfields at their intersection. Neuropsychology has used lesion studies to infer functional specialization of brain regions (e.g., Milner, 2005). Cognitive neuroscience has used functional neuroimaging to identify “modules” for specific functions such as face perception (e.g. Kanwisher & Yovel, 2006), and invasive recording and perturbation techniques to crack cir-

¹In the literature, terms like ‘top-down’ versus ‘bottom-up’, or ‘behaviour-first’ versus ‘brain-first’ are also used. We adopt the ‘functional’ versus ‘implementational’ terminology both as an umbrella distinction and because these terms are more meaningful from a computational perspective.

²We use the term “circuit” for both natural and artificial circuits.

cuits such as that for speech perception (Hamilton, Oganian, Hall, & Chang, 2021). Systems neuroscience has used connectivity mapping, direct recordings, and perturbation techniques, to identify crucial circuits supporting animal behaviors, and canonical circuits (cf. Bargmann & Marder, 2013; Olsen & Wilson, 2008). AI is increasingly concerned with mechanistic interpretability and model compression with the aim to understand which circuits in a learned system support certain aspects of machine outputs (Geva, Schuster, Berant, & Levy, 2021; Geva, Caciularu, Wang, & Goldberg, 2022). At the intersection of these fields, it has been of interest to use artificial neural networks to perform simulations of neural damage to circuits (for a meta-theoretical treatment of this subfield, see Guest, Caso, & Cooper, 2020).

These efforts are consistent with causal theories of explanation in philosophy of science (cf. Thompson, 2021). When applied at the intersection of computation and neuroscience, these meta-theories yield desiderata where circuit cracking fits naturally as an initial subgoal. More generally, a shared goalpost, which sometimes appears as a separate level of analysis (e.g. Marr & Poggio, 1976), is to find “mechanistic primitives” at early stages of inquiry (Hartley & Poeppel, 2020) to provide a foundation for and to facilitate the non-trivial business of discovering circuit-behavior mappings (cf. Rust & LeDoux, 2023).

Assumed sources of complexity

In pursuit of these non-trivial goals researchers have allocated substantial resources to remove what they believe to be primary roadblocks holding back circuit cracking. One roadblock was construed as the lack of complete “maps” of the territory; the so-called “-omes” (e.g., connectomes). To address this, researchers undertake, for instance, connectomic circuit mapping (e.g., Schmidt et al., 2017) in specific regions of an organism’s brain thought to subserve behaviors of interest. Another roadblock was the lack of precise measurement and causal intervention tools to enable full observability and perturbability. Engineering advances bring this possibility closer for natural systems (cf. Juavinett, Bekheet, & Churchland, 2019), and current software tools provide these capabilities for artificial systems (cf. Lindsay, 2022).

As ideals, complete maps, full observability, and full perturbability do not come without conceptual problems. A rarely questioned conviction is that, since we do not know what matters, every detail matters — that to understand, we must measure fully (Gomez-Marin, 2021; Niv, 2021). The promise seems to be that after achieving these first steps, deeper understanding will surely follow.

As the conceptual issues turn into roadblocks in practice, researchers acknowledge additional obstacles in the scaling up of circuit cracking efforts (e.g., A. K. Churchland & Kiani, 2016; Urai, Doiron, Leifer, & Churchland, 2022). These mostly have to do with data-analytic strategies, experimental control, and statistical inference. Where theoretical challenges are acknowledged, they are usually subservient to the development of data models. Generally, there is a preoccu-

pation with the possibility that, as we scale up to higher dimensional circuits and behaviors, common data-analytic tools might not identify the phenomena of interest efficiently (e.g., Lindsay, 2022). Critics of reductive views focus more on the challenges related to behavioral experiment designs (e.g., Niv, 2021). Together, these views provide a more nuanced perspective on what stands in the way of understanding.

However, we argue in this paper that even this updated list is missing an important element. What researchers generally fail to acknowledge are the computational constraints on notions of understanding and their associated real-world epistemic processes (i.e. the computational complexity of the tasks researchers set up for themselves in pursuit of understanding how neural circuits support behavior). We next set the formal foundations to address this gap.

Formalizing Circuit Cracking

We formalize the experimental problem facing a scientist who wants to discover a minimal sufficient neural circuit for eliciting a type of behaviour (see Fig. 1). We present definitions of problem components (i.e., circuit connectivity, neural dynamics, experimental apparatus) inspired by Ramaswamy (2019), and then put them together in precise problem definitions. By formally modeling the scientist’s problem at a high level of abstraction it becomes a transparent theoretical tool and amenable to formal analysis (cf. Morrison & Morgan, 1999; Guest & Martin, 2021; van Rooij & Blokpoel, 2020). Our analyses will be without loss of generality because the simplifications and idealisations we choose ensure that these analyses yield lower-bounds on real-world complexity.

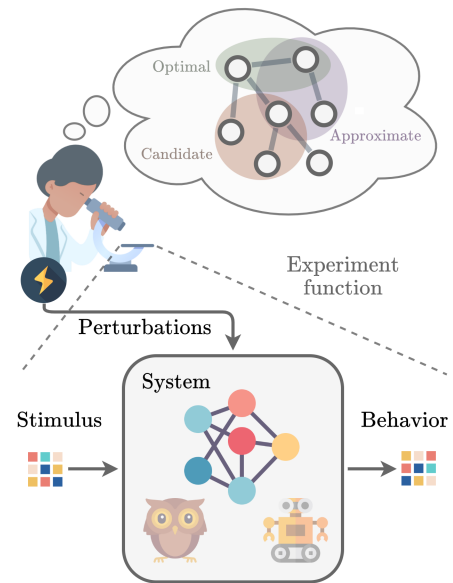


Figure 1: Schematic of the SUFFICIENT CIRCUIT problem.

Circuit Connectivity and Dynamics

Network models are ubiquitous in neuroscience, where they are used to study both small- and large-scale neural circuits.

They combine a network architecture with the dynamics that occurs at each node (cf. [Bassett, Zurn, & Gold, 2018](#)).

Definition 1 (Circuit Connectivity). A graph $G = (V, E)$ whose vertices and directed edges represent neurons and their connections, respectively, encodes the *connectivity* of a *neural circuit*. Given such a graph, a subgraph encodes a (sub)circuit.

Definition 2 (State Vector). A boolean vector $\mathbf{r} \in \mathcal{R} = \{0, 1\}^n$ that encodes the state of a circuit of size n is called a *state vector*, where r_i represents the state of the i^{th} neuron; active ($r_i = 1$) or inactive ($r_i = 0$).

Definition 3 (Circuit Parameters and Rules). The dynamics of neuronal spiking in a circuit can be expressed using *spiking rules* that specify under what conditions a neuron spikes. An example is the following:

$$g(\mathbf{q}^{(t)}, \mathbf{q}^{(t-1)}) = \begin{cases} 1 & \sum_{i=0}^1 \sum_j q_j^{(t-i)} \geq T \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{q}^{(t)}$ is a vector of inputs at time t , with q_j encoding the input from the j^{th} neuron, and T is some neuron- or neuron-type-specific firing threshold. In the example, the neuron whose spiking rule is captured by g integrates all its inputs over 2 time steps and compares the sum to a threshold to determine whether to spike at the current time step.

Experimental Apparatus

Definition 4 (Inhibition/Stimulation Matrix). A boolean matrix \mathbf{I} is an *inhibition matrix* if it encodes a spatio-temporal pattern of artificial neuronal inhibition of a circuit. The value at row i , column j indicates whether the i^{th} neuron at the j^{th} timepoint in an experiment is artificially silenced ($\mathbf{I}_{i,j} = 1$) or not ($\mathbf{I}_{i,j} = 0$). A *stimulation matrix* \mathbf{S} is defined similarly for artificial stimulation of circuits.

Definition 5 (Activity Matrix). A boolean matrix \mathbf{A} is an *activity matrix* if it encodes a spatio-temporal pattern of activation of a neural circuit. The value at row i , column j indicates whether the i^{th} neuron at the j^{th} timepoint in an experiment is active ($\mathbf{A}_{i,j} = 1$) or not ($\mathbf{A}_{i,j} = 0$).

Definition 6 (Behavioral readout). The external detection of an organism behavior is a *behavioral readout* with respect to a target circuit and it is encoded as a boolean variable b whose value, which is accessible to the experimenter (via an *experiment function*), indicates whether the behavior has been elicited ($b = 1$) or not ($b = 0$).

Definition 7 (Experiment Function). A computable function³ $F : \mathcal{R} \times \mathcal{I} \times \mathcal{S} \rightarrow \mathcal{A} \times \mathcal{B}$ that maps a state vector $\mathbf{r} \in \mathcal{R}$ for a circuit and perturbation (i.e., stimulation and inhibition) matrices $\mathbf{I} \in \mathcal{I}, \mathbf{S} \in \mathcal{S}$ to an activity matrix $\mathbf{A} \in \mathcal{A}$ and a behavioral readout $b \in \mathcal{B} = \{0, 1\}$ is an *experiment func-*

tion $F(\mathbf{r}, \mathbf{I}, \mathbf{S}) = (\mathbf{A}, b)$ that encodes the ability to conduct experiments on neural circuits by presenting the external stimulus, performing perturbations on the input circuit, and recording circuit activity and system behavior.⁴

Computational Problem

Notions of *cracking* in the context of neural circuits generally include the following tasks as necessary steps before asking more intricate questions: “(i) describing a behavior whose neural circuit mechanisms we seek to understand, (ii) identifying which neurons are involved...” ([Olsen & Wilson, 2008](#)).

Definition 8 (Sufficient Circuit). Given a neural circuit $G = (V, E)$, a subset $C \subseteq V$ of neurons is called a *sufficient circuit* for a behavior if the latter can be elicited when all neurons $v \in V \setminus C$ are excluded from the circuit. A *minimal sufficient circuit* C obtains when it is not possible to do the same with any proper subset $C' \subset C$. A *minimum sufficient circuit* is smallest among all sufficient circuits.

Problem 1. (MINIMAL / MINIMUM) SUFFICIENT CIRCUIT (search, optimization, and decision versions)

Input: a neural circuit $G = (V, E)$ with unknown parameters and rules and an experiment function F . (An integer k is part of the input in the decision version).

Output: a minimal (minimum) sufficient circuit $C \subseteq V$ for the behavior $b = 1$. (For the decision version, *Question:* is there a sufficient circuit $C \subseteq V$ of size $|C| = k$?)

Hardness and Inapproximability of Circuit Cracking

Our proofs of hardness and inapproximability build on concepts and techniques from computational complexity theory ([Garey & Johnson, 1979](#); [van Rooij et al., 2019](#); for definitions and remarks, see [Appendix](#)). We demonstrate a polynomial-time reduction from CLIQUE (a known NP-hard problem) to SUFFICIENT CIRCUIT, which proves hardness. This proof builds on a proof sketch provided in [Ramaswamy \(2019\)](#). We fill in formal details which are necessary to argue the correctness of any such proof and to provide a foundation to build subsequent proofs on. Later we use this construction to prove hardness of approximation.

Hardness

We next present a proof that for every instance of CLIQUE (decision) there exists a corresponding instance of SUFFICIENT CIRCUIT (decision) such that there is a $k + 2$ -sized sufficient circuit in the latter if and only if there is a k -clique in the former. Furthermore, we show that in each case the instance of SUFFICIENT CIRCUIT can be constructed from the instance of CLIQUE in polynomial time. Because CLIQUE is NP-hard, it follows that SUFFICIENT CIRCUIT is too.

⁴An experiment function $F_{[O, S, B]}$ is specific to a particular organism O (and therefore to the input circuit C with its parameters and rules), external stimulus S , and external behavior B . Since the input circuit and experiment function are paired together in every instance of the problem we will drop the subindices when referring to F .

³The function might also be an oracle (i.e., an imaginary device that given the input to F computes the correct output in a single step) or a tractable function, and our proofs would apply as well.

Theorem 1. (MINIMAL / MINIMUM) SUFFICIENT CIRCUIT is NP-hard.

Proof. Given an instance of CLIQUE, an undirected graph $G = (V, E)$ and an integer k , we construct an instance of SUFFICIENT CIRCUIT (decision), a directed graph $G' = (V', E')$ and integer k' , as follows (see Fig. 2 for a schematic example). Let $V' = V \cup \{s, m\}$, where s and m are new nodes. For exposition, we informally call s “sensory neuron”, m “motor neuron”, and vertices in V' corresponding to V “interneurons”, but formally these are not labelled vertices. Next, for each undirected edge $(u, v) \in E$ we construct both directional edges (u, v) and (v, u) between the corresponding interneurons $V \cap V'$ of G' (i.e., $(u, v), (v, u) \in E'$). Neuron s is connected with one directed edge (s, v) to each “interneuron” $v \in V \cap V'$, and m is connected with one directed edge (v, m) from each “interneuron” $v \in V \cap V'$. This transformation can be completed in polynomial time (i.e., $O(|V'| + |E'|) \sim O(|V| + |E|) \sim O(|V|^2)$).

Next, we construct the unknown parameters and rules (see Definition 3) which govern circuit dynamics. We add an experiment function F for the circuit (see Definition 7), which behaves according to the following rules. All conduction delays between neuron pair types are the same (e.g., all interneuron-motor delays are the same). The spiking rules are constructed as follows. The sensory neuron signals the arrival of the external stimulus which is controlled by the experimenter. Input from s or alternatively from at least $k - 1$ interneurons is enough to make interneurons spike.

$$g_{\text{inter}}(\mathbf{q}^{(t)}) = \begin{cases} 1 & [q_{\text{sensory}}^{(t)} = 1] \vee [\sum_i q_i^{(t)} \geq T = k - 1] \\ 0 & \text{otherwise} \end{cases}$$

The motor neuron spikes if it receives at least 2 subsequent inputs of at least k spikes, or 1 input of size $|V|$.

$$g_{\text{motor}}(\mathbf{q}^{(t)}, \mathbf{q}^{(t-1)}) = \begin{cases} 1 & [\sum_i q_i^{(t)} = |V|] \vee \\ & [\sum_i q_i^{(t)} \geq T \wedge \sum_i q_i^{(t-1)} \geq T], \\ & T = k - 1. \\ 0 & \text{otherwise} \end{cases}$$

We set the activity of the motor neuron m to correlate perfectly with the elicitation (or not) of the external behavior.

$$b = \begin{cases} 1 & m \text{ spikes} \\ 0 & \text{otherwise} \end{cases}$$

This completes the SUFFICIENT CIRCUIT instance.

We next consider the properties of the constructed instance. The entire resulting circuit is a *sufficient circuit* for the behavior. This can be seen by unfolding the dynamics of the circuit according to the parameters and spiking rules, without excluding any neurons. Starting from a quiet circuit, the sensory neuron spikes at time t which will elicit a spike from $|V|$ interneurons at time $t + 1$. According to the circuit’s spiking rules, this makes the motor neuron spike at time $t + 2$, so the behavior is elicited (as F will show upon evaluation).

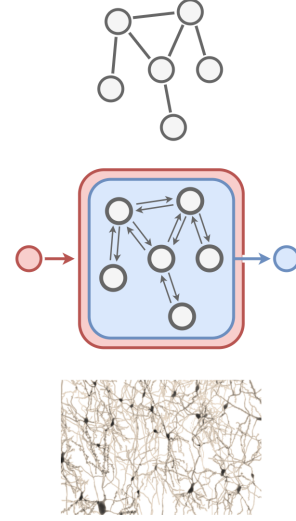


Figure 2: Schematic of an instance of SUFFICIENT CIRCUIT (middle) constructed from an instance of CLIQUE (top) in the hardness proof (Theorem 1). Circles, lines, and arrows indicate vertices, undirected edges, and directed edges. An arrow and a contour enclosing a set of vertices indicates all enclosed vertices are connected. The colouring is shown here for illustration purposes only (i.e., it is not formally part of the input). In this example, there is a 3-clique in the CLIQUE instance, and hence a $(3 + 2)$ -clique in the SUFFICIENT CIRCUIT instance. Note how the model circuit (middle) is much simpler, smaller and highly idealized compared to real neural circuits (bottom). This is why our analyses give a lower bound on real-world complexity.

Consider now the case where we exclude from the circuit (e.g., artificially silence) all interneurons in V except for a set $X \subset V$ of size $|X| = k$. (Sensory and motor neurons cannot be silenced without abolishing the behavior). At time $t + 1$, only $|X|$ interneurons are allowed to spike in response to incoming input from the sensory neuron. At time $t + 2$, the motor neuron receives $|X| < |V|$ spikes (insufficient to elicit a spike) and at most $|X|$ interneurons receive input spikes from $|X| - 1$ other interneurons. Then, at time $t + 3$, the motor neuron receives at most $|X|$ spikes from interneurons. If X is not a k -clique, at $t + 2$ fewer than k of the interneurons X will have received input from at least $k - 1$ other interneurons, hence at time $t + 3$ the motor neuron will receive fewer than k spikes, which is insufficient to spike. If, on the other hand, it receives $|X| = k$ spikes, the spiking rule dictates it will fire. This happens if and only if X is a k -clique. In this case, the sensory and motor neuron, and X , together form a circuit which qualifies as a $(k + 2)$ -sized *minimum sufficient circuit* for the behavior. ■

Inapproximability

We next prove inapproximability of MINIMUM SUFFICIENT CIRCUIT by combining ideas from the hardness proof above with a variant on a classic proof technique described in Garey and Johnson (1979; see also van Rooij & Wareham, 2012).

The key idea is that we can construct a larger instance of MINIMUM SUFFICIENT CIRCUIT by creating multiple copies of a given (smaller) instance and connecting them such that obtaining an approximate solution for the larger instance would imply obtaining an optimal solution for the smaller instance. This implies that if MINIMUM SUFFICIENT CIRCUIT were tractably approximable, then the problem would also be tractably solvable exactly. As this contradicts [Theorem 1](#), we can conclude that no such approximation is possible.

Theorem 2. MINIMUM SUFFICIENT CIRCUIT cannot be d -approximated tractably.

Proof. We prove by contradiction. Given a graph $G = (V, E)$ and an integer k , create circuit $G' = (V', E')$ and set $k' = k + 2$ for an instance of SUFFICIENT CIRCUIT with all parameters and spiking rules as in the hardness proof, except that the original sensory and motor neurons now drop their direct dependence on the stimulus and behavior, respectively, to become part of an interneuron circuit. Create d copies of G' such that we obtain $G'_1, G'_2, \dots, G'_{d+1}$ disconnected subgraphs. Add 2 new neurons named (informally) “sensory” and “motor”. Connect the sensory neuron unidirectionally towards the original sensory neurons, and the motor neuron unidirectionally from the original motor neurons (see [Fig. 3](#) for a schematic).

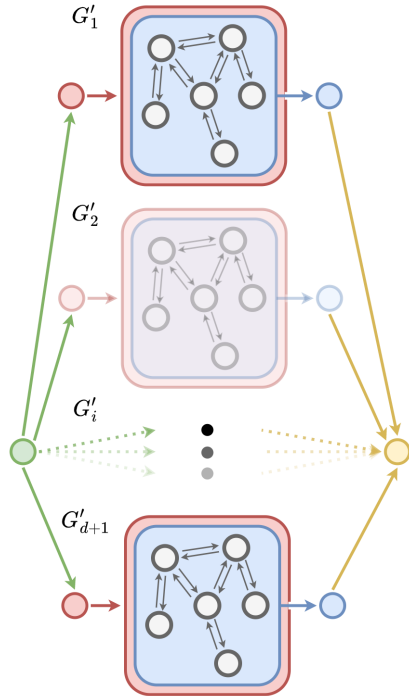


Figure 3: A schematic of the padded-instance created in the inapproximability proof ([Theorem 2](#)).

Next we define an experiment function F , parameters, and rules for the circuit, exactly as in the hardness proof. As in the proof of [Theorem 1](#), for each copy G' the “original” motor neuron (blue in [Fig. 3](#)) will spike if and only if G has a clique of size k . The motor neuron (yellow in [Fig. 3](#)) spikes if and

only if it receives concurrent input from all $d + 1$ neurons that were originally motor neurons in the non-padded instance.

$$g_{motor}(\mathbf{q}^{(t)}) = \begin{cases} 1 & \sum_i q_i^{(t)} \geq T = d + 1 \\ 0 & \text{otherwise} \end{cases}$$

Assume for a contradiction that there exists an approximation algorithm A for MINIMUM SUFFICIENT CIRCUIT capable of computing a solution that is at most a distance $d \in \mathbb{N}$ away from the optimal solution $optsol(\cdot)$. That is, the size of the subcircuit solution, in number of neurons, is such that

$$|optsol(I_{msc})| \leq |A(I_{msc})| \leq |optsol(I_{msc})| + d$$

Since there are $d + 1$ copies in the padded instance, and the approximate solution is at most d away from optimal, at least one copy must have an optimal solution to its non-padded instance. We can find this special copy in polynomial time by checking each copy for the minimum number of chosen neurons that overlap with the solution set for the padded instance. This, in turn, allows to compute an answer to the CLIQUE problem, and we have thus produced the desired contradiction. We conclude there cannot exist a d -approximation algorithm for MINIMUM SUFFICIENT CIRCUIT as defined above. ■

Discussion

Discovery of explanations of natural and artificial cognition can be approached in multiple ways. A major issue of contention has been whether implementation-first approaches may have a claim to primacy. Here we studied the demands of a core scientific problem in the implementationist research agenda: *circuit cracking*. We verify that this problem is intractable to solve exactly ([Theorem 1](#)), building directly on previous work ([Ramaswamy, 2019](#)). It is conceivable, however, that researchers might be able to leverage approximate answers, if these could be obtained tractably. Here we have begun to ask and answer this open question formally. Our proof of inapproximability shows it is not possible ([Theorem 2](#)). Inapproximability holds even when the scientist has perfect experimental control, observability and perturbability, and even when the behaviors are simple.

Our results speak to a fundamental and irreducible computational barrier that is qualitatively distinct from commonly assumed challenges. These include, for instance, coming up with clever behavioral experiment designs ([Niv, 2021](#); [Juavinett, Erlich, & Churchland, 2018](#)), establishing causality under naturalistic conditions ([Marinescu, Lawlor, & Kording, 2018](#)), the impossibility of inferring cognition from correlation ([Guest & Martin, 2023](#)), choosing and interpreting replication experiments ([Devezer & Buzbas, 2021](#)), and eliciting ecological behaviors and neural activity using ecological stimuli to avoid phenomena outside the environment as experienced by the organism ([Testard, Tremblay, & Platt, 2021](#)). While these are legitimate, non-trivial obstacles, removing them does not make a dent to computational intractability.

This constraint should be part of discussions on the merits of different approaches and the challenges each pose.

Our analyses should similarly inform discussions on scaling up circuit understanding efforts in light of new technical capabilities. There is an ongoing preoccupation with the suitability of current data-analytic tools to tackle increasingly larger and higher-dimensional circuits and datasets (Lindsay, 2022). However, the kinds of barriers we uncover here are rarely⁵ acknowledged as a source of concern, especially as the field aims to scale up analyses to larger networks.⁶ In the notable cases where they are acknowledged, the assumption seems to be that better technology and data-analytic insights are crucial to mitigate them. However, because the hardness and inapproximability shown here are intrinsic in nature, better tools cannot possibly make a difference (their indispensability to surpass other obstacles notwithstanding).

The counterintuitive disconnect between complexity barriers and the fast pace of data-analytic developments should give researchers pause. There are two potential consequences of ignoring it. First, even with well-honed tools we are at risk of compounding errors that waste years of scientific resources. Second, by ignoring computational complexity and letting our data-analytic tools do the driving, we perpetuate a pattern where our tooling plays a disproportionate role in deciding what ideas succeed and which fail (cf. Hooker, 2020).

More concretely, consider what a complexity-theoretic barrier looks like from the researcher’s point of view. An intuitive scenario is the attempt, by individual researchers, to use data-analytic tools (e.g., analysis software) to solve intractable problems. These obstacles are comparatively easy to spot, as one quickly finds that analysis pipelines take an infeasible amount of time to return results. A more slippery but no less forgiving form of the barrier can be found in the distributed practices of individuals involved in parallel efforts. In this case our intuitions about complexity-theoretic obstacles are more prone to fail us. Consider two scenarios describing what intractability might look like for the collective search for a target object (e.g., minimal circuits for a given behavior). In scenario (a) the search does not terminate. That is, a research community engages in the search for evidence but fails to produce research output because the chosen procedures can only be inefficient to give an answer one way or the other. In scenario (b) the search is forced to terminate prematurely due to practical constraints. For instance, researchers might be pressured, by forces better described in a sociology of science frame (Field & Derksen, 2020), to produce intermediate outputs and to draw conclusions. These outputs and interpretations, however, are likely to contain severe errors, as procedures which might in principle return correct answers if

given enough time are cut short. For many (perhaps most) fields, scenario (b) seems like the most plausible. Crucially, this scenario accumulates and compounds errors. Our hardness of approximation result is relevant here because it shows that these errors cannot be contained to be “small”.

One upshot of our work is that theories in the cognitive sciences should be allowed to speak about the mechanisms behind behavior at higher levels of abstraction even when finding or empirically verifying some of the associated implementation parts may not be feasible. To get a grasp of the issue, consider a hypothetical example involving the “gnostic neurons” (or “grandmother cells”) in neuroscience, psychology, and artificial intelligence (cf. Barwich, 2019; Thomas & French, 2017; Gale, Martin, Blything, Nguyen, & Bowlers, 2020). We have here an explanation which puts forward an empirical object: X-selective neurons, where X is some complex but specific concept. Finding some form of such units within a circuit, if the conjecture is true, could be done tractably by algorithmic variants on brute-force search⁷. Now consider a different theory whose associated implementation part happens to be intractable to find (an example might involve the minimum circuits for the instances we construct in our proofs). If we mostly let our data-analytic tools do the appraisal for us, only theories which happen to posit tractably discoverable empirical objects will be selected for. Those theories that fail this test would be discarded independent of their explanatory merits. The hypothetical scenario we describe is arguably the reality in many subfields. If we factor in the widespread belief among researchers that most questions (including explanation appraisal) are “empirical questions”, it becomes clear that the situation could not be more delicately poised. It is only a matter of when and how far, not whether, a field will wander off course as researchers “let the data speak” and choose epistemic paths based on what empirical objects are revealed in the data’s utterances. Safeguards in the form of non-empirical theoretical appraisal, among others, are therefore indispensable. This implies that we need to actively seek this level of abstraction and to tolerate the state of uncertainty that it comes with.

The above considerations do not imply that the implementationist research lines cannot be fruitful. They may well lead to new discoveries. But they do not have the primacy that is often ascribed to them (cf.⁸ Poeppel & Adolfs, 2020). This is important because a more balanced distribution over functional and implementation approaches to explaining cognition is more likely to make discoveries than an effort fixated on any single approach (cf. Rich et al., 2021). Our work suggests that if we postpone functional theory until we have an implementationist grasp, we may end up getting neither.

⁵Even though neuro-scientists routinely express complexity intuitions, these are rarely examined formally. Oftentimes they can be shown wrong upon closer formal analysis (e.g. Adolfs, Wareham, & van Rooij, 2022; Woensdregt et al., 2021; Zeppi & Blokpoel, 2017).

⁶A non-polynomial time algorithm (no faster method can exist as per our proofs) running, e.g., in time $O(2^n)$ may not yet pose severe problems for circuits of size $n = 10$ but for $n = 100$ it would take millennia to run (see Ramaswamy, 2019 for an analysis of runtimes).

⁷This might depend on definitional differences which are generally important but not relevant to our point here.

⁸The reader is cautioned against buying into the broader framing Poeppel & Adolfs (2020) adopt (i.e., accidentally reinforcing “great man theorizing”; cf. Guest, 2023) — an unintentional (albeit, foreseeable and preventable) consequence of using critique of a single person’s claims to counteract widespread misconceptions.

Acknowledgments

We thank Todd Wareham for feedback on the final version of the manuscript, and anonymous reviewers for their comments. FA thanks David Poeppel for support and discussions on the maps and mapping problems in cognitive neuroscience, and Jeff Bowers for discussions on circuit understanding in neural networks.

References

- Adolfi, F., Wareham, T., & van Rooij, I. (2022). A computational complexity perspective on segmentation as a cognitive subcomputation. *Topics in Cognitive Science*.
- Ausiello, G., Marchetti-Spaccamela, A., Crescenzi, P., Gambosi, G., Protasi, M., & Kann, V. (1999). *Complexity and Approximation*. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Bargmann, C. I., & Marder, E. (2013). From the connectome to brain function. *Nature Methods*, 483–490.
- Barwich, A.-S. (2019). The value of failure in science: The story of Grandmother Cells in neuroscience. *Frontiers in Neuroscience*, 13, 1121.
- Bassett, D. S., Zurn, P., & Gold, J. I. (2018). On the nature and use of models in network neuroscience. *Nature Reviews Neuroscience*, 566–578.
- Churchland, A. K., & Kiani, R. (2016). Three challenges for connecting model to mechanism in decision-making. *Current Opinion in Behavioral Sciences*, 74–80.
- Churchland, P. S., & Sejnowski, T. (1999). *The computational brain* (5. print ed.). Cambridge, Mass.: MIT Press.
- Devezer, B., & Buzbas, E. (2021). *Minimum viable experiment to replicate* [Preprint]. Retrieved from <http://philsci-archive.pitt.edu/21475/>
- Egan, F. (2018). Function-theoretic explanation and the search for neural mechanisms. In *Explanation and integration in mind and brain science* (pp. 145–163). Oxford University Press.
- Field, S. M., & Derksen, M. (2020). Experimenter as automaton; experimenter as human: Exploring the position of the researcher in scientific research. *European Journal for Philosophy of Science*, 11.
- Fortnow, L. (2009). The status of the P versus NP problem. *Communications of the ACM*, 52(9), 78–86.
- Gale, E. M., Martin, N., Blything, R., Nguyen, A., & Bowers, J. S. (2020). Are there any ‘object detectors’ in the hidden layers of CNNs trained to identify objects or scenes? *Vision Research*, 60–71.
- Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability*. Freeman: San Francisco.
- Geva, M., Caciularu, A., Wang, K. R., & Goldberg, Y. (2022). *Transformer Feed-Forward Layers Build Predictions by Promoting Concepts in the Vocabulary Space*. arXiv.
- Geva, M., Schuster, R., Berant, J., & Levy, O. (2021). *Transformer feed-forward layers are key-value memories*. arXiv.
- Gomez-Marin, A. (2021). Promisomics and the short-circuiting of mind. *eNeuro*, 8(2).
- Guest, O. (2023). *What makes a good theory, and how do we make a theory good?* PsyArXiv.
- Guest, O., Caso, A., & Cooper, R. P. (2020). On Simulating Neural Damage in Connectionist Networks. *Computational Brain & Behavior*, 289–321.
- Guest, O., & Martin, A. E. (2021). How computational modeling can force theory building in psychological science. *Perspectives on Psychological Science*, 16(4), 789–802.
- Guest, O., & Martin, A. E. (2023). On logical inference over brains, behaviour, and artificial neural networks. *Computational Brain & Behavior*, 1–15.
- Hamilton, L. S., Oganian, Y., Hall, J., & Chang, E. F. (2021). Parallel and distributed encoding of speech across human auditory cortex. *Cell*, 4626–4639.e13.
- Hartley, C. A., & Poeppel, D. (2020). Beyond the Stimulus: A Neurohumanities Approach to Language, Music, and Emotion. *Neuron*, 597–599.
- Hooker, S. (2020). *The Hardware Lottery*. arXiv.
- Juavinett, A. L., Bekheet, G., & Churchland, A. K. (2019). Chronically implanted Neuropixels probes enable high-yield recordings in freely moving mice. *eLife*, e47188.
- Juavinett, A. L., Erlich, J. C., & Churchland, A. K. (2018). Decision-making behaviors: Weighing ethology, complexity, and sensorimotor compatibility. *Current Opinion in Neurobiology*, 42–50.
- Kanwisher, N., & Yovel, G. (2006). The fusiform face area: A cortical region specialized for the perception of faces. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 2109–2128.
- Lindsay, G. W. (2022). *Testing the tools of systems neuroscience on artificial neural networks*. arXiv.
- Marinescu, I. E., Lawlor, P. N., & Kording, K. P. (2018). Quasi-experimental causality in neuroscience and behavioural research. *Nature Human Behaviour*, 891–898.
- Marr, D., & Poggio, T. (1976). From understanding computation to understanding neural circuitry. *A.I. Memo*, 357(3), 1–22.
- Milner, B. (2005). The Medial Temporal-Lobe Amnesic Syndrome. *Psychiatric Clinics of North America*, 599–611.
- Morrison, M., & Morgan, M. S. (1999). Models as mediating instruments. In *Models as mediators: Perspectives on natural and social science* (pp. 10–37). Cambridge University Press.
- Niv, Y. (2021). The primacy of behavioral research for understanding the brain. *Behavioral Neuroscience*, 601–609.
- Olsen, S. R., & Wilson, R. I. (2008). Cracking neural circuits in a tiny brain: New approaches for understanding the neural circuitry of *Drosophila*. *Trends in Neurosciences*, 512–520.
- Poeppel, D., & Adolfi, F. (2020). Against the Epistemological Primacy of the Hardware: The Brain from Inside Out, Turned Upside Down. *eNeuro*, ENEURO.0215-20.2020.
- Ramaswamy, V. (2019). An algorithmic barrier to neural circuit understanding. *BioRxiv*, 639724.
- Rich, P., de Haan, R., Wareham, T., & van Rooij, I. (2021).

- How hard is cognitive science? *Proceedings of the Annual Meeting of the Cognitive Science Society*.
- Rust, N. C., & LeDoux, J. E. (2023). The tricky business of defining brain functions. *Trends in Neurosciences*, 3–4.
- Schmidt, H., Gour, A., Straehle, J., Boergens, K. M., Brecht, M., & Helmstaedter, M. (2017). Axonal synapse sorting in medial entorhinal cortex. *Nature*, 469–475.
- Testard, C., Tremblay, S., & Platt, M. (2021). The value of failure in science: the story of grandmother cells in neuroscience. *Current Opinion in Neurobiology*, 76–83.
- Thomas, E., & French, R. (2017). Grandmother cells: Much ado about nothing. *Language, Cognition and Neuroscience*, 342–349.
- Thompson, J. A. F. (2021). Forms of explanation and understanding for neuroscience and artificial intelligence. *Journal of Neurophysiology*, 1860–1874.
- Urai, A. E., Doiron, B., Leifer, A. M., & Churchland, A. K. (2022). Large-scale neural recordings call for new insights to link brain and behavior. *Nature Neuroscience*, 11–19.
- van Rooij, I., Blokpoel, M., Kwisthout, J., & Wareham, T. (2019). *Cognition and intractability: A guide to classical and parameterized complexity analysis*. Cambridge University Press.
- van Rooij, I., & Baggio, G. (2021). Theory before the test: How to build high-verisimilitude explanatory theories in psychological science. *Perspectives on Psychological Science*, 16(4), 682–697.
- van Rooij, I., & Blokpoel, M. (2020). Formalizing verbal theories. *Social Psychology*, 51(5), 285–298.
- van Rooij, I., & Wareham, T. (2012). Intractability and approximation of optimization theories of cognition. *Journal of Mathematical Psychology*, 56(4), 232–247.
- Voss, C., Cammarata, N., Goh, G., Petrov, M., Schubert, L., Egan, B., ... Olah, C. (2021). Visualizing Weights. *Distill*, e00024.007.
- Voss, C., Goh, G., Cammarata, N., Petrov, M., Schubert, L., & Olah, C. (2021). Branch Specialization. *Distill*, e00024.008.
- Woensdregt, M. S., Spike, M., de Haan, R., Wareham, T., van Rooij, I., & Blokpoel, M. (2021). Why is scaling up models of language evolution hard? In *Proceedings of the annual meeting of the cognitive science society* (Vol. 43).
- Zeppi, A., & Blokpoel, M. (2017). Mindshaping the world can make mindreading tractable: Bridging the gap between philosophy and computational complexity analysis. In *Proceedings of the 39th annual meeting of the cognitive science society*.

Appendix

Computational Complexity Analysis

Our proofs of hardness and inapproximability build on concepts and techniques from computational complexity theory (van Rooij et al., 2019; Garey & Johnson, 1979). We give a set of definitions that are used in our proofs and/or their interpretation.

Definition 9 (Polynomial-time). An algorithm is said to run in *polynomial-time* if the number of steps it performs is on the order of n^c (also denoted $O(n^c)$), where n is a measure of the input size and c is some constant.

Definition 10 (Polynomial-time reducibility). Let \mathcal{P} and \mathcal{Q} be computational problems. We say \mathcal{P} is *polynomial-time reducible* to \mathcal{Q} if there exists a polynomial-time algorithm (called *reduction*) that transforms instances of \mathcal{P} into instances of \mathcal{Q} such that solutions for \mathcal{Q} can be transformed in polynomial-time into solutions for \mathcal{P} .

Definition 11 (NP-hard). NP-hard problems are the *hardest* problems in the class NP.⁹ A problem is said to be *NP-hard* if all problems in NP can be polynomial-time reduced to it.

Definition 12 (Intractability). NP-hard problems are considered *intractable*, i.e., unsolvable with a realistic amount of resources (e.g. time and space), because they cannot be solved in polynomial-time (unless $P = NP$)¹⁰.

Note that it follows from the above definitions that a problem \mathcal{Q} can be shown to be intractable by polynomial-time reducing a known NP-hard problem \mathcal{P} to \mathcal{Q} .

The proofs construct reductions from the known NP-hard problem CLIQUE (Garey & Johnson, 1979; van Rooij et al., 2019)

Definition 13 (Clique). Given a graph $G = (V, E)$, a subset of vertices $V' \subseteq V$ is called a *clique* if all $v \in V'$ are adjacent to each other under G (i.e., it is a complete subgraph).

Problem 2. CLIQUE (decision version)

Input: An undirected graph $G = (V, E)$ and an integer k .

Question: Is there a clique $V' \subseteq V$ of size $|V'| \geq k$?

In order to prove intractability for non-trivial scenarios we assume the experimenter only sets up tractable tasks for their experimental subjects. Furthermore, even if the experimenter had perfect knowledge of F , and it was tractable, our results would hold.

Problems that are intractable may still be tractable to approximate.¹¹ We analyse the complexity of approximating circuit cracking in the following sense.

Definition 14 (Approximation¹²). Given an optimization

problem \mathcal{P} and an approximation algorithm \mathcal{A} for \mathcal{P} , we call \mathcal{A} an *approximation algorithm* if there exists a constant d such that for all instances x of \mathcal{P} the absolute error between the value $m(\cdot)$ of an optimal solution $optsol(x)$ and the output $\mathcal{A}[x]$ is such that $|m(optsol(x)) - m(\mathcal{A}[x])| \leq d$.

Hardness and Inapproximability of Circuit Cracking

Here we spell out some aspects of the inapproximability proof.

Inapproximability Consider the properties of the constructed instance. The entire circuit is a candidate solution, i.e. a *sufficient circuit* for the behavior, because if the sensory neuron spikes, $(d + 1)$ of the original sensory neurons will spike, subsequently $(d + 1)$ copies of $|V|$ interneurons will receive enough input to spike, which is enough to make each of the original motor neurons spike, and this in turn is a sufficient condition for the motor neuron to spike. Now consider what happens when some neurons are excluded from the circuit. The sensory and motor neuron of the padded instance, as well as the original sensory and motor neurons of the non-padded instances, cannot be excluded without abolishing the behavior. This is because excluding any of these would preclude the possibility of concurrent spikes from all original motor neurons arriving at the motor neuron. We can therefore consider next the case where we exclude none of the neurons just mentioned and all but $(d + 1)k$ of the original interneurons V_1, \dots, V_{d+1} . If even a single one of the $d + 1$ copies has less than k included interneurons, then its corresponding original motor neuron will not spike, hence the motor neuron of the padded instance cannot spike due to receiving strictly less than $d + 1$ concurrent inputs. This implies that k interneurons in each of $d + 1$ copies must be chosen to be in the solution set. Each of the original motor neurons will spike if and only if the k chosen interneurons in the corresponding copy form a k -clique, by the same reasoning as in the hardness proof.

⁹Here NP stands for *non-deterministic polynomial time*.

¹⁰Here P stands for (deterministic) *polynomial time*. It is widely conjectured that $P \neq NP$ (Fortnow, 2009).

¹¹It is generally overestimated how often this is possible for relevant notions of approximation (van Rooij & Wareham, 2012).

¹²See also Ausiello et al., 1999.

Expressing Psychological Distance Using Sharma-Mittal Divergence

Mikaela Akrenius (makreniu@indiana.edu)

Cognitive Science Program, 1001 E. 10th Street, Bloomington, IN 47405 USA

Abstract

Psychological distance spaces are the building block of many cognitive models, such as the generalized context model (Nosofsky, 1986) and the similarity-choice model (Luce, 1963; Shepard, 1957). The distance between two stimuli is typically computed based on a multidimensional scaling solution using the Minkowski power metric. This paper proposes a novel method for computing pairwise dissimilarities between stimulus representations that is based on the Kullback-Leibler divergence of response distributions. The method is extended with Sharma-Mittal divergence, and its application and properties are illustrated using a classic set of perceptual identification and categorization data.

Keywords: multidimensional scaling; identification; categorization; information theory; Kullback-Leibler divergence; Sharma-Mittal divergence; generalized context model; similarity-choice model

Introduction

Multidimensional scaling (MDS) is a powerful tool for data analysis, visualization, and dimensionality reduction that is widely applied across the sciences. Given a set of pairwise distances between data points, MDS finds coordinates in a Cartesian space that best fit the distances.

The work of Torgerson (1952), Shepard (1962a, 1962b), Kruskal (1964a, 1964b), and others helped popularize the use of MDS within cognitive psychology, leading to the development of models that use multidimensional scaling to postulate a mental space that defines stimulus locations relative to two or more feature dimensions. Examples of such models include the MDS-choice model, an extension of the similarity-choice model (Luce, 1963; Shepard, 1957), which translates distances in the representational space into similarities that determine response rates in identification experiments, and the generalized context model (Nosofsky, 1986), which applies the same principle to categorization.

A key feature of these models is that information about pairwise connections between stimuli, such as similarity ratings or confusions between their respective responses, is indicative of the number of features that the stimuli are represented upon and of the locations of stimuli in this representation. For example, similarity ratings between different kinds of cars could be used to construct a mental space in which each car is represented as a location relative to various dimensions (e.g., type, brand, color). Cars that are closer to each other in the representational space would be harder to identify (confused more often), and cars that are further from each other would be easier to identify.

In most of these applications – as in applications of MDS in general – the distance between stimuli is defined using the Minkowski (1896, see e.g., Borg & Lingoes, 1987, Borg & Groenen, 2005) power metric. The Minkowski metric

generalizes several distance measures (e.g., Euclidean, Manhattan) and has been shown to capture task- and feature-specific variation in the structure of the similarity space (e.g., Shepard, 1964; Garner, 1974; Wiener-Ehrlich, 1978; Dunn, 1983; Melara, Marks, & Lesko, 1992). In addition, it has been shown that selective attention can drive the structure of the similarity space by stretching or shrinking it along one or more of its coordinate axes, and that learning can increase distances between stimuli, which can be modeled by complementing the Minkowski metric with additional parameters (e.g., Nosofsky, 1986, 1987).

The purpose of this project is to explore whether distance measures based on Kullback-Leibler (1961) divergence, Sharma-Mittal (1975) divergence, or other information-theoretic constructs could yield results that are comparable to the Minkowski metric and/or provide new mathematical properties that would be meaningful considering psychological data.

We will begin by introducing the Minkowski metric, Kullback-Leibler (KL) divergence, and Sharma-Mittal (SM) divergence. After that, we will propose a way to apply KL and SM divergence to extract pairwise dissimilarities between representations of individual stimuli from confusion matrix data. Finally, we will run MDS on these dissimilarities and illustrate their properties using existing identification and categorization data.

Definitions: Minkowski Distance, Kullback-Leibler (KL) Divergence, and Sharma-Mittal (SM) Divergence

The Minkowski (1896, see e.g., Beckenbach & Bellman, 1961) distance of order r between points $x = (x_1, \dots, x_n) \in \mathbb{R}$ and $y = (y_1, \dots, y_n) \in \mathbb{R}$ in an n -dimensional space is

$$D_M(x, y) = [\sum_{i=1}^n |x_i - y_i|^r]^{1/r} \quad (1)$$

where $r \in [-\infty, \infty]$ determines the concavity of the distance metric used. For Euclidean distance, $r = 2$, whereas for Manhattan (city-block) distance $r = 1$. The lower the value of r is, the more weight smaller component (unidimensional) distances are given when computing overall (multidimensional) distance, resulting in larger overall distance when r is smaller for pairs of points that are not parallel to the coordinate axes.

The Kullback-Leibler (KL, 1961) divergence between the probability distributions $p = (p_1, \dots, p_n) \in [0, 1]$ and $q = (q_1, \dots, q_n) \in [0, 1]$, $\sum p_i = 1$, of a discrete random variable X (or two discrete random variables X and Y) with n outcomes is

$$D_{KL}(p||q) = \sum_{i=1}^n p_i \ln \left(\frac{p_i}{q_i} \right) \quad (2)$$

where p_i and q_i denote probabilities associated with corresponding outcomes.¹ Unlike the Minkowski metric, KL divergence is not symmetric ($D_{KL}(p||q) \neq D_{KL}(q||p)$) and it violates the triangle inequality, due to which it is not considered a distance metric. KL divergence can, however, be made symmetric by taking the average of $D_{KL}(p||q)$ and $D_{KL}(q||p)$ (Jeffreys, 1948, divergence) or by applying some other transformation, e.g., Jensen-Shannon divergence (Wong & You, 1985; Lin & Wong, 1990; Lin, 1991; see e.g., Nielsen, 2019, for alternatives).

The Sharma-Mittal (1975) family of entropies is a generalization of Rényi and Tsallis entropies and gives rise to Sharma-Mittal (SM) divergence of order α and degree β

$$D_{SM}(p||q) = \frac{1}{\beta-1} \left[\left(\sum_{i=1}^n p_i^\alpha q_i^{1-\alpha} \right)^{\frac{1-\beta}{1-\alpha}} - 1 \right] \quad (3)$$

where $\alpha > 0$, $\alpha \neq 1$, and $\beta \neq 1$. When $\alpha \rightarrow 1$ and $\beta \rightarrow 1$, SM divergence corresponds to KL divergence (i.e., Shannon entropy), whereas when $\alpha \rightarrow 1$ and $\beta \rightarrow \alpha$ Rényi and Tsallis entropies are recovered, respectively (see e.g., Nielsen & Nock, 2011, for formal properties, and Crupi et al., 2018, for an application in cognitive science). Figure 1 illustrates the influence of variation in α and β on the concavity of the entropy function, and Figure 2 illustrates their influence on the convexity of the divergence function when comparing a distribution with two outcomes to the uniform distribution.

Constructing Dissimilarities from Confusion Matrix Data using KL and SM Divergence

To provide a basis for the construction of an MDS solution that represents stimulus coordinates in the postulated mental space, pairwise dissimilarities between individual stimuli are computed using KL divergence. This procedure translates a $n \times m$ confusion matrix of n stimuli S and m responses r (e.g., 16×4 for A_1B_1, \dots, A_4B_4 and a_1, \dots, a_4) into a $n \times n$ matrix of pairwise dissimilarities between stimuli (e.g., 16×16 for A_1B_1, \dots, A_4B_4 and A_1B_1, \dots, A_4B_4 , where diagonal entries denote dissimilarity to self). Formally, the KL divergence between the response distribution of stimulus S_i and the response distribution of stimulus S_j is

$$D_{KL}(r|S_i||r|S_j) = \sum_{k=1}^m p(r_k|S_i) \ln \left(\frac{p(r_k|S_i)}{p(r_k|S_j)} \right) \quad (4)$$

where S_i and S_j denote stimuli used in the experiment, and r_k denotes one of m possible responses. For example, if $S_i = A_1$ and $S_j = A_4$ in the stimulus set $A = \{A_1, A_2, A_3, A_4, A_5, A_6\}$ with response options $a = \{a_1, a_2, a_3\}$,

$$D_{KL}(a|A_1||a|A_4) = p(a_1|A_1) \ln \left(\frac{p(a_1|A_1)}{p(a_1|A_4)} \right) + p(a_2|A_1) \ln \left(\frac{p(a_2|A_1)}{p(a_2|A_4)} \right) + p(a_3|A_1) \ln \left(\frac{p(a_3|A_1)}{p(a_3|A_4)} \right). \quad (5)$$

¹ Note that Shannon entropy (and KL divergence) is typically defined using the binary logarithm. We use the natural logarithm because it is the base of the Sharma-Mittal family of entropies.

This process is repeated for every combination of S_i and S_j , and the result is stored in the corresponding cell of the $n \times n$ matrix of pairwise dissimilarities.

Similarly, the SM divergence between the response distributions of S_i and S_j is computed using

$$D_{SM}(r|S_i||r|S_j) = \frac{1}{\beta-1} \left[\left(\sum_{k=1}^m p(r_k|S_i)^\alpha p(r_k|S_j)^{1-\alpha} \right)^{\frac{1-\beta}{1-\alpha}} - 1 \right] \quad (6)$$

where $\alpha > 0$, $\alpha \neq 1$, and $\beta \neq 1$, and stored in a $n \times n$ dissimilarity matrix.

Executing MDS with KL and SM Divergence

Most of the statistical packages that execute MDS include Euclidean distance as the default distance metric, with the possibility to run the analysis using other Minkowski distance metrics as well. Typically, the MDS function is called with a similarity matrix and a set of parameters that define the kind of analysis to be run (e.g., metric vs. non-metric, number of dimensions, distance metric). The MDS function translates the similarity matrix into a matrix of pairwise distances or dissimilarities (disparities) and finds coordinates that provide the best fit between distances in the MDS solution and disparities in the dissimilarity matrix by minimizing a stress function. The user can then compare solutions acquired with a different number of dimensions or with different distance metrics to determine which one of them provides the best fit. To avoid overfitting, the number of dimensions is chosen based on the end point of a steep decline in the stress function.

In our case, the MDS function is called with a precomputed, KL- or SM-divergence-based dissimilarity matrix, due to which the first step of the MDS procedure is bypassed. These dissimilarities are then used by the MDS function for finding the best fitting MDS solution (coordinates in Cartesian space)², and the process is repeated for different values of α and β (i.e., the precomputed dissimilarity matrix is built again using different values of α and β). Because MDS requires a symmetric matrix, entries on opposite sides of the diagonal (e.g., (0,1) and (1,0)) in the dissimilarity matrix are averaged.³

² Note that the MDS package used in this paper (see Methodological Notes) uses Euclidean distance to quantify interpoint distances in the MDS solution regardless of the measure used for computing pairwise disparities. This produces a reasonable stress for MDS-KL but introduces bias for other values of α and β , especially $\beta > 1$. Using alternative distance metrics (Minkowski $r \neq 2$, cosine, etc.) in MDS-SM and/or developing methods for divergence-based MDS is a topic for future work.

³ Alternatively, KL divergence and SM divergence could be replaced with Jensen-Shannon and Jensen-Sharma-Mittal (Luza, 2021) divergence, respectively, or a divergence measure that satisfies metric properties, such as Jensen-Shannon distance (Endres & Schindelin, 2003; Österreicher & Vajda, 2003).

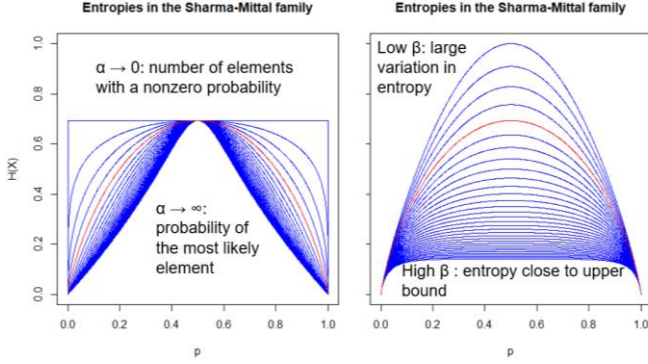


Figure 1: Illustration of the influence of α (when β is fixed to 0.99999, left) and β (when α is fixed to 0.99999, right) on the value of Sharma-Mittal entropy for a distribution of two probabilities. All entropies reach their maximum at uniformity ($p = 1 - p$) and minimum at certainty ($p = 0$ or $p = 1$). Shannon entropy is denoted with a red line.

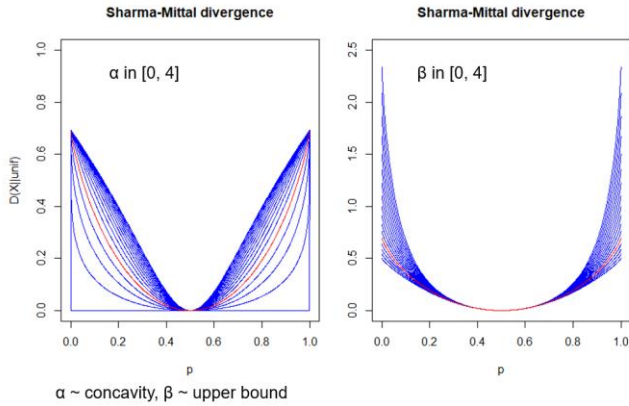


Figure 2: Illustration of the influence of α (when β is fixed to 0.99999, left) and β (when α is fixed to 0.99999, right) on the value of Sharma-Mittal divergence between $(p, 1 - p)$ and the uniform distribution $p = 1 - p = 1/2$. All divergences reach their minimum when the distributions are identical. KL divergence is denoted with a red line.

Example Application: Nosofsky (1986)

To provide an illustration of the kind of results that can be acquired with MDS-KL and MDS-SM, the proposed method was applied to data from Nosofsky (1986).

Procedure

Nosofsky collected identification and categorization data from two participants with stimuli that varied on two orthogonal dimensions with four levels. The stimuli were half-circles with a radial line pointing from the center, and they were varied in the size of the circle and the angle of the radial line. The identification part of the experiment consisted of three conditions: identifying both angle and size (AS), identifying angle only (A), and identifying size only (S). The categorization part of the experiment included classifying the stimuli into two groups based on four

different category structures: dimensional, criss-cross, interior-exterior, and diagonal. The category structures are illustrated in Figure 4.

To model the data of the AS condition, Nosofsky applied the MDS-choice framework (Shepard, 1957), which consists of three parts: (1) an MDS solution that represents stimulus coordinates relative to two or more feature dimensions, (2) a function that translates interstimulus distances in the MDS solution to interstimulus similarities, and (3) a choice rule that determines the probability of responding R_j when stimulus S_i is shown based on interstimulus similarities and response bias parameters. Nosofsky (1985) found that the model that best fit the data consisted of an MDS-solution with two dimensions, a Euclidean distance metric, and a Gaussian function for translating distance to similarity.

For the A, S, and categorization conditions, Nosofsky (1986) developed an extension of Medin and Schaffer's (1978) context theory of classification, the generalized context model (GCM). The GCM proceeds in the following way: (1) extract stimulus coordinates from the MDS-choice solution of the AS condition, (2) compute interstimulus distances using the augmented Minkowski metric (described below), (3) translate distances into similarities using the same similarity function that was used in the AS condition, and (4) use a choice rule for responding C_k (category k) when stimulus S_i is shown based on interstimulus similarities and response bias parameters. For more detail on the procedure, see Nosofsky (1986).

The Augmented Minkowski Metric

GCM presumes that, through attention allocation, the goal of the task can influence the shape of the perceptual space. This is modeled with the augmented Minkowski metric (Nosofsky, 1986, p. 41):

$$D_M(x, y) = c[\sum_{i=1}^n w_i |x_i - y_i|^r]^{1/r} \quad (7)$$

where $0 \leq w_i \leq 1$, $\sum w_i = 1$ denotes the attentional weight given to each dimension, and $0 \leq c < \infty$ is a scaling parameter. When a dimension is given a high attentional weight (e.g., angle in condition A), it is stretched and, hence, identification and categorization performance on that dimension is improved. On the other hand, when a dimension is given a low attentional weight (e.g., size in condition A), stimuli become more confusable on that dimension. In addition, learning or increased stimulus exposure duration can increase the distance between stimuli, which can be modeled with the scaling parameter c .

Note that the order parameter r of the Minkowski metric works analogously to the order parameter α of the Sharma-Mittal family of entropies: when α approaches 0, smaller probabilities are given more weight, which makes the entropy function more concave and divergence function more convex. Decreasing r , on the other hand, increases the weight given to smaller component distances, which makes the distance function more concave. Furthermore, the Sharma-Mittal degree parameter β increases the divergence

between two stimuli in proportion to the difference in their response distributions, whereas increasing the scaling parameter c increases the distance between all stimuli. Hence, even though α and r (and β and c) serve somewhat similar purposes, their exact implications on pairwise dissimilarities (and the consequent shape of the postulated perceptual space) are different.

Results

Condition AS Figure 3 illustrates the results of MDS-KL for subject 1 (upper row) and subject 2 (lower row) in the AS identification condition of the experiment. As can be seen from the figure, apart from differences in the rotation of the solution (which is arbitrary) and stretching in the corners of the solution, the results resemble the results of MDS-choice relatively closely. This is interesting given the very different ways in which these results are acquired: MDS-choice finds the stimulus coordinates and bias parameters that can best predict response probabilities given a distance metric, a translation from distance to similarity, and a choice rule, whereas the pairwise dissimilarities of MDS-KL are computed directly from the confusion matrix. Hence, in this sense, MDS-KL bypasses the choice rule and distance-to-similarity conversion of MDS-choice.

Conditions A, S, and Categorization When MDS-KL is applied to unidimensional identification or categorization, the results become one-dimensional (Figure 4). This is because the pairwise dissimilarity between two stimuli is determined by the difference in their response distributions: if the response distributions only carry information about one dimension (i.e., involve one response option per level of the dimension) the interstimulus distances are also determined solely by that dimension. Hence, unlike GCM, MDS-KL does not postulate a perceptual space with more information than is included in the response options. For the A and S conditions, MDS-KL can order and cluster stimuli based on the level of the response variable, and express “distances” between individual stimuli (Figure 4, left column). For the categorization conditions, MDS-KL can order stimuli from the most distinctive representative of category 1 to the most distinctive representative of category 2, cluster them based on their similarity to each other, and quantify the “distance” of outliers from other stimuli (Figure 4, middle and right column).⁴

Varying α and β To illustrate the influence of α and β on the MDS solution, MDS-SM is applied to the full identification (AS) condition of subject 1 (Figure 5).⁵

⁴ This is somewhat trivial for a two-category classification task as the same information could be deduced from response probabilities alone. However, for tasks with a larger number of response options the solution would be more insightful.

⁵ As mentioned in ², note that Euclidean MDS is not able to fully capture the influence of variation in α and β on pairwise dissimilarities. Hence, the solutions presented here are directional

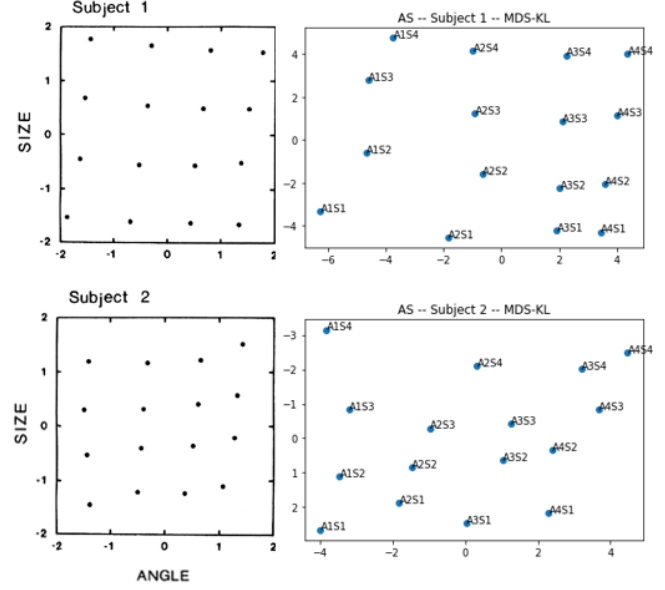


Figure 3: The results of MDS-KL for identification data in condition AS for subject 1 (upper right) and subject 2 (lower right), as compared to Gaussian-Euclidean MDS (upper left and lower left) in Nosofsky (1986).

When α approaches 0, pairwise dissimilarities between stimuli approach 0, which is reflected in the MDS solution as a clustering of the stimuli. When α increases, the divergence function becomes less convex, increasing dissimilarity overall and especially between stimuli that have moderately similar response distributions. This is reflected in the MDS solution as more space being created in the center of the solution.

When β approaches 0, the range of the divergence function becomes more restricted, resulting in less distance between stimuli that have less similar response distributions. When β increases, the divergence between less similar response distributions increases exponentially, producing an MDS solution with more distance overall and especially between the edges and corners of the solution.

When α and β are varied together, interstimulus distances approach 0 when α and β approach 0, and when α and β are increased together the overall scale of the solution increases with less prominent changes in the shape of the solution. When both α and β approach 1, the solution becomes equivalent to the solution of MDS-KL.

Comparison to GCM As compared to GCM, the results of MDS-KL for the A, S, and categorization conditions could be thought of as an alternative to augmented Minkowski distance, with the weight w_i given to each dimension being reflected in the order of the stimuli in the MDS-KL solution, and the scaling parameter c corresponding to the overall resolution of the MDS-KL solution. For example, in the S and dimensional categorization conditions, the order of the

rather than definite and the properties of MDS-SM should be evaluated together with more appropriate distance metrics.

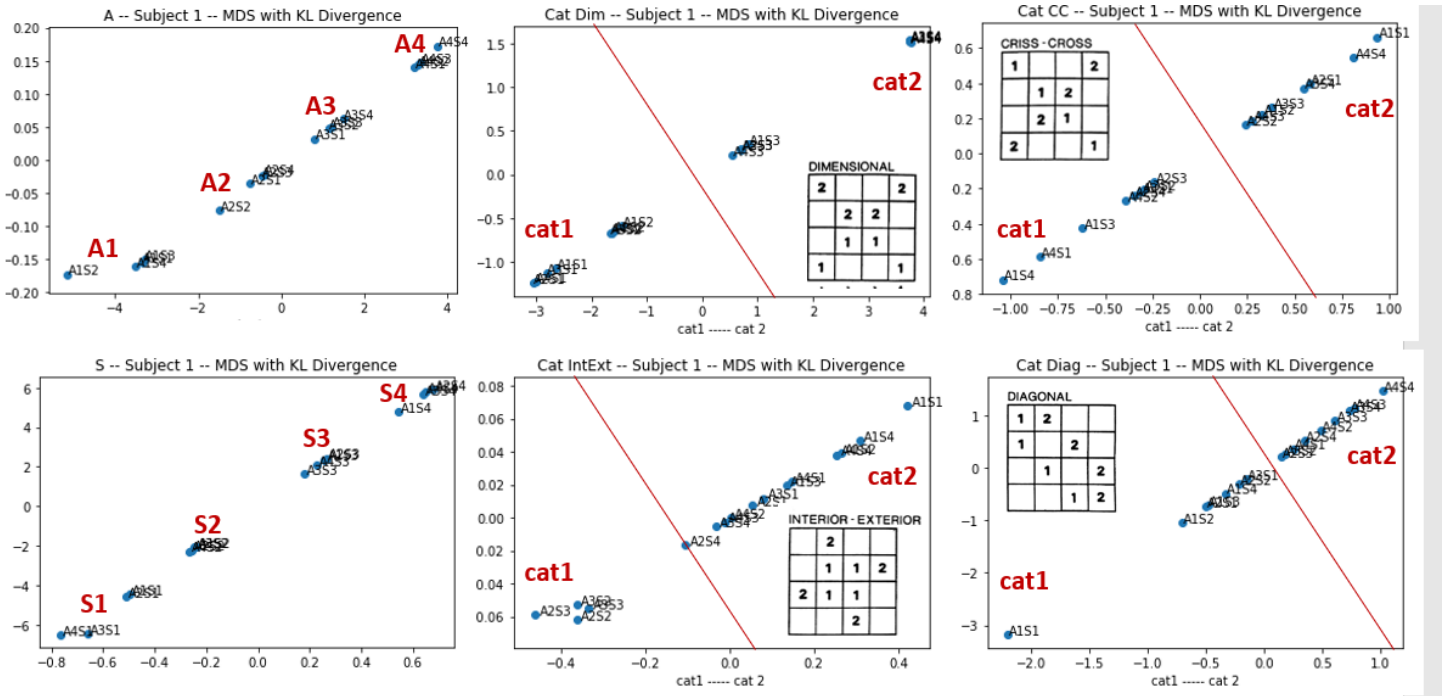


Figure 4: Left: The results of MDS-KL for subject 1 in identification condition A (upper left) and identification condition S (lower left). Middle and right: The results of MDS-KL for subject 1 in dimensional (upper middle), criss-cross (upper right), interior-exterior (lower middle), and diagonal (lower right) categorization. The red line depicts the boundary for a higher than 0.50 probability of assigning the stimulus to each category. Illustrations of the category structures are adapted from Nosofsky (1986).

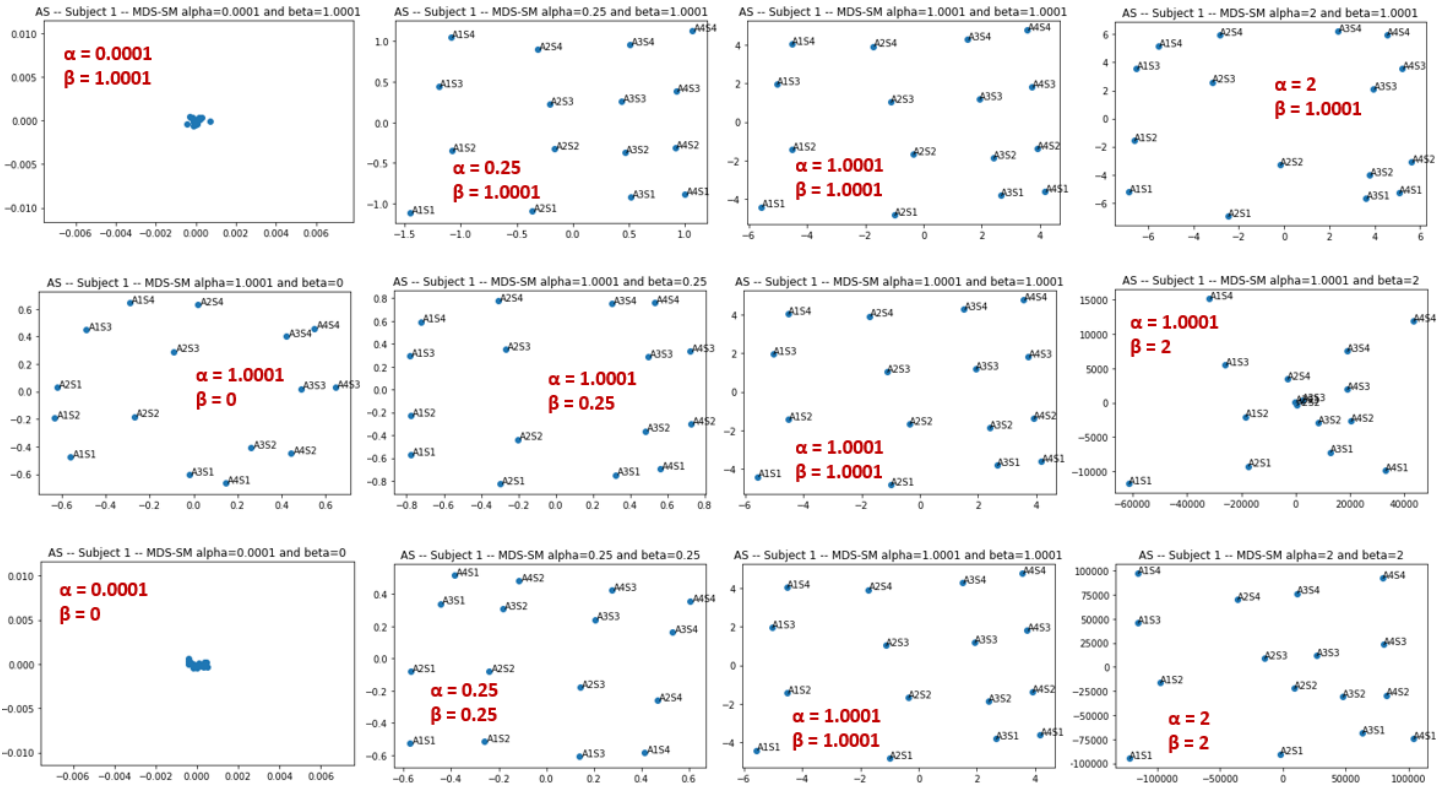


Figure 5: Illustration of the influence of α and β on the results of MDS-SM for subject 1 in the AS identification condition. Upper row: α is varied from 0.0001 to 2 while β is fixed to 1.0001. Middle row: β is varied from 0 to 2 while α is fixed to 1.0001. Lower row: α and β are varied together from ~ 0 to 2. When $\alpha \rightarrow 1$ and $\beta \rightarrow 1$ the results of MDS-SM are equivalent to the results of MDS-KL.

stimuli is determined solely by dimension S , whereas in other categorization conditions it is based jointly on A and S . Hence, the weight given to each dimension corresponds to the mutual information between the dimension and the ordering of the stimuli in the MDS-KL solution.

As for MDS-SM, while varying α and β together can produce results that resemble variation in the scaling parameter c , varying α or β separately does not correspond to any existing construct in GCM. Furthermore, because MDS-KL and MDS-SM are based on pointwise *differences* in response distributions, they can account for response bias without postulating it as an explicit part of the model. Due to this, MDS-KL and MDS-SM do not have direct equivalents to the choice rule or response bias parameters.

Conclusions

The purpose of this paper was to introduce a novel, KL-divergence-based dissimilarity measure for confusion matrix data and its extension with the Sharma-Mittal family of entropies. The protocol for generating a dissimilarity matrix using this method was described, along with the protocol for running MDS on the matrix, and the results of this process were illustrated using identification and categorization data from Nosofsky (1986).

MDS-KL provided remarkably similar results to the results of Gaussian-Euclidean MDS-choice in the full, bidimensional identification response condition, and was able to express order, clustering, and “distances” in unidimensional identification and categorization response conditions. These properties were compared to the weight and scaling parameters of the augmented Minkowski metric, noting that MDS-KL bypasses the choice rule and distance-to-similarity conversion of MDS-choice and GCM by computing dissimilarities directly from the confusion matrix. Therefore, when the structure of the perceptual space is of primary interest, MDS-KL (and MDS-SM) could be able to access it with less computational resources.

Complementing MDS-KL with the Sharma-Mittal family of entropies (MDS-SM) provided a novel way to adjust pairwise dissimilarities, with α corresponding to the weight given to small probabilities and β corresponding to the magnitude of atomic information when computing the divergence between two response distributions. Hence, α and β could be considered as measures of how diagnostic certain features of the response distribution (e.g., small probabilities) are for deducing the structure of the underlying perceptual representation. Depending on the context, this could reflect between-participant or between-condition variation in the shape of the response distribution induced e.g., by changes in task, response rule, or accuracy. On the other hand, as illustrated in this paper, variation in α and β can also be used to alter the shape of the perceptual representation. Unlike the augmented Minkowski metric, which adjusts scaling, dimensions, and the metric of fitted distances, MDS-SM adjusts pairwise dissimilarities (disparities) based on information from the entire response distribution. Consequently, MDS-SM produces results that

are qualitatively different from the results of the augmented Minkowski metric.

Suggestions for Future Work

An obvious next step to the work presented in this paper would be to translate distances generated by MDS-KL and MDS-SM into interstimulus similarities by using the similarity function and choice rules of MDS-choice and GCM and to test these against existing (e.g., Euclidean-Gaussian) MDS using diagnostic confusion matrix data (e.g., Nosofsky, 1986, 1987, 1989).⁶

The proposed approach could also be extended to other types of data. For example, optional processes (induced by e.g., changes in task instructions) can have an impact on the structure of the MDS solution (e.g., Melara, Marks, & Lesko, 1992) in similarity rating tasks. If these processes were reflected in the shape of the response distribution, α and β could be better able to account for them than approaches based on the Minkowski metric. Furthermore, because KL-divergence (and SM-divergence) is not symmetric and violates the triangle inequality, it could be better suited for describing data that is not symmetric and/or violates the triangle inequality (e.g., Tversky, 1977).

Finally, as suggested in ^{2,3,5}, the proposed work could be complemented with other types of entropies and divergence measures, and/or divergence-based methods for MDS. The approach could also be complemented with modern versions of mutual uncertainty analysis (Garner & Morton, 1969; e.g., Fitousi, 2013; Akrenius, 2021) to capture the influence of stimulus structure (in addition to task structure and individual differences) on perceptual distance.

Methodological Notes

The dissimilarity matrices of MDS-KL and MDS-SM were constructed from confusion matrix data using a Python program that implemented (4), (6), and the steps described in the same paragraph. Metric, 2-dimensional MDS was run on the matrices with `sklearn.manifold.MDS` in the `skicitlearn` Python package (Pedregosa et al., 2011) using the SMACOF (Scaling by MAjorizing a COMplicated Function) algorithm. Estimates of Sharma-Mittal divergence close to a limit ($\alpha \rightarrow 0$, $\alpha \rightarrow 1$, and $\beta \rightarrow 1$) were computed by using a number that differed from the limit by a fourth decimal (e.g., 0.0001).

Acknowledgments

I would like to thank an anonymous reviewer, Leslie Blaha, and Terry Stewart for their comments on an earlier version of this paper.

⁶ Alternatively, given the close connection between KL-divergence and mutual information, pairwise divergences could be translated into mutual information, and a mutual-information-based choice rule could be developed to yield response probabilities. This approach, however, would be more aligned with GRT (Ashby & Townsend, 1986; cf. overlap in perceptual distributions) than with MDS-choice and GCM.

References

- Akrenius, M. (2021). Applications of information theory to perceptual independence and separability. In Stewart, T. C. (Ed.). *Proceedings of the 19th International Conference on Cognitive Modelling* (pp. 1-7). University Park, PA: Applied Cognitive Science Lab, Penn State.
- Ashby, F. G., & Townsend, J. T. (1986). Varieties of perceptual independence. *Psychological Review*, 93, 154–179.
- Borg, I., & Groenen, P. J. F. (2005). *Modern multidimensional scaling: Theory and applications*. Springer Science + Business Media.
- Borg, I., & Lingoes, J. (1987). *Multidimensional similarity structure analysis*. Springer-Verlag.
- Crupi V., Nelson J.D., Meder B., Cevolani G., & Tentori K. (2018). Generalized information theory meets human cognition: Introducing a unified framework to model uncertainty and information search, *Cognitive Science*, 42, 1410–1456.
- Dunn, J. C. (1983). Spatial metrics of integral and separable dimensions. *Journal of Experimental Psychology: Human Perception and Performance*, 9(2), 242–257.
- Endres, D. M., & Schindelin, J. E. (2003). A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7), 1858–1860.
- Fitousi, D. (2013). Mutual information, perceptual independence, and holistic face perception. *Attention, Perception, & Psychophysics*, 75, 983–1000.
- Garner, W. R., & Morton, J. (1969). Perceptual independence: Definitions, models, and experimental paradigms. *Psychological Bulletin*, 72, 233–259.
- Garner, W. R. (1974). *The processing of information and structure*. Lawrence Erlbaum.
- Jeffreys, Harold (1948). *Theory of Probability (Second ed.)*. Oxford University Press.
- Kluza, P. A. (2021). Inequalities for Jensen-Sharma-Mittal and Jeffreys-Sharma-Mittal Type f -Divergences. *Entropy*, 23(12), 1688.
- Kruskal, J. B. (1964a). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1), 1–27.
- Kruskal, J. B. (1964b). Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29(2), 115–129.
- Kullback, S., & Leibler, R.A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22, 79–86.
- Lin, J., & Wong, S. K. M. (1990). A new directed divergence measure and its characterization. *International Journal of General Systems*, 17(1), 73–81.
- Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1), 145–151.
- Luce, R. D. (1963). Detection and recognition. In R. D. Luce, R. R. Bush, & E. Galanter (Eds.), *Handbook of Mathematical Psychology* (pp. 103–189). NY: Wiley.
- Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review*, 85(3), 207–238.
- Melara, R. D., Marks, L. E., & Lesko, K. E. (1992). Optional processes in similarity judgments. *Perception & Psychophysics*, 51(2), 123–133.
- Nielsen, F., & Nock, R. (2011). A closed-form expression for the Sharma–Mittal entropy of exponential families. *Journal of Physics A: Mathematical and Theoretical*, 45(3).
- Nielsen F. (2019). On the Jensen-Shannon symmetrization of distances relying on abstract means. *Entropy*, 21(5), 485.
- Nosofsky, R. M. (1985). Overall similarity and the identification of separable-dimension stimuli: A choice model analysis. *Perception & Psychophysics*, 38, 415–432.
- Nosofsky, R. M. (1986). Attention, similarity, and the identification–categorization relationship. *Journal of Experimental Psychology: General*, 115(1), 39–57.
- Nosofsky, R. M. (1987). Attention and learning processes in the identification and categorization of integral stimuli. *Journal of Experimental Psychology*, 13(1), 87–108.
- Nosofsky, R.M. (1989). Further tests of an exemplar-similarity approach to relating identification and categorization. *Perception & Psychophysics*, 45, 279–290.
- Österreicher, F., & Vajda, I. (2003). A new class of metric divergences on probability spaces and its applicability in statistics. *Annals of the Institute of Statistical Mathematics*, 55, 639–653.
- Pedregosa, F., Varoquaux, Gaël, Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct), 2825–2830.
- Sharma B.D., & Mittal D.P. (1975). New nonadditive measures of inaccuracy. *Journal of Mathematical Science.*, 10, 122–133.
- Shepard, R. N. (1957). Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space. *Psychometrika*, 22, 325–345.
- Shepard, R. N. (1962a). The analysis of proximities: Multidimensional scaling with an unknown distance function. Part 1. *Psychometrika*, 27(2), 125–140.
- Shepard, R. N. (1962b). The analysis of proximities: Multidimensional scaling with an unknown distance function. Part 2. *Psychometrika*, 27(3), 219–246.
- Shepard, R. N. (1964). Attention and the metric structure of the stimulus space. *Journal of Math. Psych.*, 1(1), 54–87.
- Torgerson, W. S. (1952). Multidimensional scaling: I. Theory and method. *Psychometrika*, 17, 401–419.
- Tversky, A. (1977). Features of similarity. *Psychological Review*, 84(4), 327–352.
- Wiener-Ehrlich, W. K. (1978). Dimensional and metric structures in multidimensional stimuli. *Perception & Psychophysics*, 24(5), 399–414.
- Wong, A. K. C., & You, M. (1985). Entropy and distance of random graphs with application to structural pattern recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(5), 599–609.

From Knowledge Graph to Cognitive Model: A Method for Identifying Task Skills

Ivana D.M. Akrum (ivana.akrum@tno.nl)

TNO, Kampweg 55, 3769 DE
Soesterberg, Netherlands

Niels A. Taatgen (n.a.taatgen@rug.nl)

University of Groningen, Bernoulli Institute
Groningen, Netherlands

Abstract

When we learn new tasks, rather than starting from scratch, we often reuse skills that we have learned previously. By integrating these previously learned skills in a new way, we can learn how to do new tasks with little effort. In this research, we test a method aimed at identifying the skills reused between tasks. More specifically, we use a knowledge graph as a tool for identifying reused skills. From this knowledge graph, we built a cognitive model that shows how the identified skills can be integrated to solve a new task. The final cognitive model can successfully solve a variety of related but distinct tasks. This shows it is possible to use knowledge graphs to identify the skills reused between tasks. This ability may benefit how we approach learning. Knowing, in advance, the skills needed to successfully complete a new task may allow us to learn said task in an easier, more focused manner.

Keywords: cognitive modelling; knowledge graph; skill transfer; cognitive tutors

Introduction

Cognitive architectures have provided many insights in how people process information, reason and learn by providing precise predictions through simulation. Despite the many successes, there are a number of limiting factors that become important when the complexity of tasks increases. A first limitation is that most models are constructed for a specific experiment or set of experiments. This means that the knowledge in the model is specifically tailored to that task. Instead, if humans have to perform a new task, they build task knowledge on the knowledge they already have, which many lead to a different set of knowledge for the task, and a different learning trajectory than a constructed model.

A second limiting factor is that models are constructed. It is generally assumed that a model is valid if it fits enough data, or better, if it is able to make testable predictions. However, it is quite possible that alternative models provide the same or even better fit. Moreover, as tasks become more complex it is unlikely that all subjects follow the same strategy, making it more likely that the eventual experimental results are produced by a mixture of slightly different models.

The goal of this paper is to use the *skill-based learning* approach (Hoekstra, Martens, & Taatgen, 2020) in combination with data-driven methods to arrive at a hybrid approach to modeling, in which we use data to inform us about the structure of the model. To fill in this structure we then use “traditional” modeling to fill the details and create a runnable model.

The skill-based approach

The assumption of the skill-based approach to modeling (Hoekstra et al., 2020) is that when humans have to perform a new task, they combine a number of existing knowledge components in a novel way, similar as in language where we compose words in novel sentences to create new meaning. We call these knowledge components *skills*. In terms of cognitive architectures, a skill is a set of production rules or operators (depending on the nomenclature of the architectures). We define a skill as the largest collection of operators that can be reused between tasks. For example, Hoekstra et al. have shown that the Attentional Blink task, which is a novel task for most subjects, can be successfully modeled as a combination of a visual search and a memory consolidation skill, and that the Attentional Blink effect is due to the choice of the wrong consolidation skill.

The large advantage of the skill-based approach is that it can explain how people can do many novel tasks with very little learning, assuming they already have the right skills, because they only have to instantiate a few variables in the necessary skills instead of acquiring a whole new set of operators.

We implement this idea in the PRIMs cognitive architecture (Taatgen, 2013). PRIMs has been derived from ACT-R (Anderson et al., 2004), and shares many of its representations and mechanisms. Specific in PRIMs is that it has been designed with reuse of knowledge in mind. Operators (production rules) in PRIMs are in declarative memory, instead of a separate procedural memory, and are selected based on activation. Operators are not linked to any specific task or goal, but are activated by context. To perform certain tasks, the skills relevant to that task are activated (i.e., placed in the goal buffer), and these in turn activate their relevant operators to perform that skill. Skills can be instrumental to multiple tasks, operators can be associated with multiple skills, and even the smaller components of operators can be reused among operators. So, whenever there is overlap in knowledge, even on different levels of abstraction, the architecture exploits it.

Data-drive modeling

How do you know what the right set of skills is to model tasks? Instead of leaving this completely up to the modeler, we can also try to derive skills from data. The key idea is that individual differences can tell us something about the number and nature of the skills. If one group of subjects fails

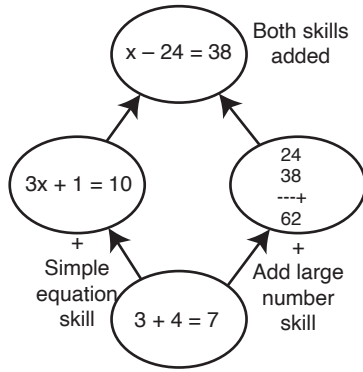


Figure 1: Example constructed knowledge graph

to solve a particular type of problem, while another group is successful, it means that the latter group has mastered one or more skills that the former has not. The goal is to derive a knowledge graph from the data, in which each node builds upon the nodes below it. Figure 1 shows a constructed example of such a graph. The basis for this graph is two (very broad) skills: the ability to solve simple equations, and the ability to add larger numbers. Students can have any combination of these skills, and if we have a large enough sample of students making problems from these four categories, the four groups will emerge that match the four nodes in the graph.

To do this more systematically, we have derived an algorithm that creates a graph out of data. We do not have the space here to discuss the details of the algorithm, but broadly speaking the outline is as follows:

The input for the algorithm is a matrix with subjects (students) as rows, and assignments as columns, and in each of the cells whether the student solved the assignment. We also predefine the number of skills that we want in the graph. The first step is to cluster the students in the rows, so that students that perform roughly equal on the assignments are grouped together. The second step is to derive the graph from the reduced matrix. We do this by starting with an empty graph, where the number of nodes depends on the number of skills we are looking for (two skills in the example). The algorithm then tries to assign problems to nodes with the goal to minimize a penalty. The penalty depends on the relation of that problem to other problems. If problem A is earlier in the graph than problem B (i.e., there is direct path from B to A), then you expect students that can solve B can also solve A. Any exception to this results in a penalty. Similarly, if problems A and B are assigned to the same node, any difference in the answer of students gives a penalty. All penalties between problems are added together, providing the value that we try to minimize, which we do by simulated annealing.

Objectives

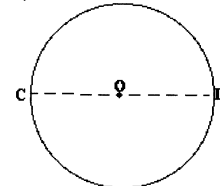
The goal of this study is to take a dataset of students solving geometry problems, and use the graph algorithm to identify the skills necessary to solve the problems. We then take the

outcome and use it as a basis for modeling these skills in PRIMs.

Dataset

The dataset used in this study is the ‘Geometry Area (1996-97)’ dataset, which is publicly available via DataShop (Koedinger et al., 2010). The dataset contains the answers of 59 students to 40 geometry problems. All problems focus on the areas of geometric shapes. To illustrate what these problems may look like, Figure 2 shows two example problems for circles.

CIRCLE_O: SECTION FOUR, #1

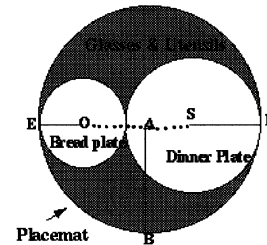


Problem Statement

1. In Circle O, diameter CD has a measure of 28 cm, find the radius, the area and the circumference of the circle.
2. In Circle O, radius OD has a measure of 28 cm, find the diameter, the area and the circumference of the circle.
3. If the area of circle O is $1764 \cdot \pi$ square cm, find the radius, the diameter and the circumference of the circle.
4. If the circumference of Circle C is $112 \cdot \pi$ cm, find the diameter, the radius and the area of the circle.

$\pi = 3.1416$

TWO-CIRCLES-IN-A-CIRCLE: SECTION FIVE, #5



Problem Statement

Aaron is preparing to host a big dinner party for the newly elected Mayor. He wants everything to be just right. He is expecting twelve guests. Setting his table, he puts the dinner plate to the right, and the bread plate to the left. Because his table is so small, he knows he needs circular placemats that do not extend beyond the combined diameters of the dinner and bread plate if he is going to fit everyone.

Given that the radius of the dinner plate is 6.6 inches, and the bread plate is 3.3 inches, find the size (area) of placemat that Aaron will need to purchase, and then the area of remaining space he will have for glasses and utensils.

Figure 2: Two circle problems in the dataset

As these examples indicate, problems in the dataset consist of an illustrated shape and one or multiple questions (with a maximum of four). A question may consist of multiple steps, as shown by e.g. question 1 of the CIRCLE_O problem. This question asks students to calculate the radius, area, and circumference of the depicted circle. Each of these calculations is its own step, and students were expected to provide a response for each step.

Rather than the raw response, the ‘Geometry Area (1996-97)’ dataset contains the evaluation of whether the student’s response was correct or not. Each row in the dataset thus specifies a student, a problem step, and its evaluated outcome. In an ideal situation, each student would have answered all steps of all problems. In practice, not all students solved all problem steps nor all problems. The full dataset contains only the outcomes of the problem steps students responded to.

To generate the knowledge graph, the outcomes to the problem steps are stored in a matrix that has the students as rows and the problem steps as columns. Each cell in this 59 by 139 matrix is a student outcome. The outcomes are encoded as 1s and 0s for correct and incorrect outcomes respectively. If a student did not respond to a problem step, an NA is recorded.

To achieve a reliable outcome, a subset of the matrix was selected as input for the graph algorithm. This subset consists of the 69 problem steps that have the most recorded responses, i.e. that were answered by the most students. Two students did not respond to any of these problem steps and were therefore excluded from the subset.

In addition to the outcome matrix, a second matrix is passed to the graph algorithm that contains the order of the problem steps. This order is different for each student, so the order matrix contains, for each student, the order in which they answered the problem steps.

Knowledge Graph

The knowledge graph algorithm takes as its input the outcome and order matrices. Both are needed to generate a knowledge graph that accurately represents the skills underlying the dataset. Specifically, the graph algorithm uses the matrices to calculate penalties, as explained in the Introduction. The algorithm compares each problem pair in the outcome matrix and assigns penalties according to the following rules:

1. If problem A and B are assigned to the same node, it is expected that all students have the same outcome to both problems. A penalty is assigned for each student that did not have the same outcome to both problems.
2. If problem A is placed in a later node in the graph than problem B, and the two nodes connect, then it is expected that students who answer problem A correctly, also answer problem B correctly. A penalty is assigned each time this is not the case.
3. If problem A is placed in an earlier node in the graph than problem B, and the two nodes connect, then it is expected that students who answer problem B correctly, also answer problem A correctly. A penalty is assigned each time this is not the case.
4. If problem A and B are assigned to two unconnected nodes, then their outcomes are expected to be different. A penalty is assigned each time a student has the same outcome for both problems.

The above rules result in four penalty matrices, one for each case. The total penalty consists of all four matrices added together. The penalty matrices are modulated by the order of the problems, where the penalty in case 2 and 3 may be negated if they can be attributed to a learning effect (i.e. a student has learned the correct outcome over time).

Using the penalty matrices, the graph algorithm assigns each problem to each possible node so as to minimise its penalty. It uses simulated annealing to find an optimal solution, where each problem is placed in the node that leads to its minimum penalty. To determine the optimal number of skills, we ran the algorithm for increasing numbers of skills, each time examining the penalty of the solution. After running it for seven skills the total penalty did not decrease anymore significantly.

The final knowledge graph generated through this method for this dataset is shown in Figure 3. The top node, node 0, contains problems that require none of the skills underlying the dataset (represented by the bit string: 0000000). The bottom node, node 127, contains all skills (1111111), meaning all skills are needed to solve the problems in that node.

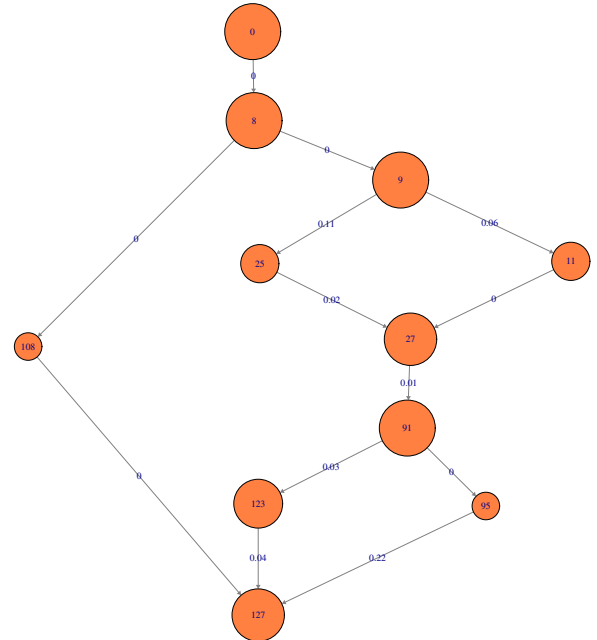


Figure 3: The knowledge graph generated by the graph algorithm

For time reasons, we will not identify all seven skills. Rather, we focus on node 0 and node 8 to identify one skill underlying this dataset. Figure 4 shows the content of these two nodes. Node 0 contains only one problem: PAINTING_THE_WALL. Node 8 contains several triangle and trapezoid problems. As visible by the bit strings at the bottom of the nodes, these two nodes differ (according to the graph algorithm) in one skill.

Cognitive Models

Through the cognitive models, we try and identify the skill that is needed to solve the problems in node 8, given the prior knowledge established by node 0. We will set up two models in the PRIMs architecture¹ One for node 0 (Model 0) and for node 8 (Model 8). These two models should differ in one skill only, meaning Model 8 should be able to reuse most of the skills it needs from Model 0.

Representing the Problems

As a starting point, we need to define a visual input that accurately represents the problems of the dataset. This includes a representation of both the geometric shape as well as the question(s) asked. As in ACT-R, visual input in PRIMs is represented by chunks in the visual buffer. To represent the geometric shape, multiple visual chunks are used.

Each visual chunk consists of (at most) seven slots (item) that together describe the represented visual object. The chunks are structured through a visual hierarchy, where a screen object represents the highest level of the hierarchy. Contained within the screen is a geometric shape that forms the “top” shape. The top shape, in turn, can consist of bases, heights, and other shapes. Given this hierarchy, it is possible for the cognitive model to shift its visual attention to another chunk in the same level of the hierarchy (through the third slot in each visual chunk) or to move one level down in the hierarchy (through the fourth slot in each chunk).

Beyond the hierarchical information, each visual chunk contains information about the visual object it is representing. Generally, the second slot identifies the visual item’s type, the fifth gives its width, the sixth its height, and the seventh its area. The widths, heights, and areas that are given in the questions are copied directly to the visual input, skipping over any necessary reading (since that can be quite complex and is not the model focus). If a value is missing (or not applicable, e.g. in the case of a base not having a height or area), it is recorded as `nil`. There is one visual chunk that does not follow this format, which is the top screen chunk. Next to the hierarchy, the screen chunks holds a top-level goal in the fifth slot and, if applicable, a target shape in the sixth slot. The questions of each problem are thus represented by parsing the given numbers and placing the top-level goal (e.g. finding a shaded area) in the screen chunk.

Model 0

As explained prior, node 0 contains only the PAINTING-THE-WALL problem, which consists of four questions. Each question asks the student to calculate the shaded area of the shape, that is: the area of the wall minus the door. The four questions are identical save from the given numbers for the various parts of the visual input.

In the visual input, the wall is represented by the top shape (as explained in the previous subsection), whose height is

given in each of the questions but whose base has to be calculated separately. This calculation involves adding the bases AE, EF (base of the door), and FB. Once the base of the wall is found, it has to be multiplied with the height of the wall to get the area of the wall. Then, the area of the door needs to be calculated (whose base and height are both given in the question). With both the area of the wall and the door, the shaded area can be found by subtracting the area of the door from the area of the wall.

To model this process of solving the PAINTING-THE-WALL problem, Model 0 starts off with three defined skills. The first skill is the `shaded-area` skill. This skill tries to find the shaded area by subtracting the area of lower-level shapes from the top-level shape. If the area of the top-level shape is not known (i.e. `nil` in the visual representation), then the model switches to an `area` skill that calculates the area of a rectangle, given its base and height. If in the `area` skill, the model finds the base is missing, it switches to a `base` skill to calculate the base of a shape by adding the segments that make up said base.

Each time the model finds an intermediate answer (like a base or area), it updates the visual input. This mimics the idea of writing down intermediate answers, which students were expected to do when they were solving the problems of the geometry dataset. Eventually, the visual input will be such that the top-level `shaded-area` skill can execute fully, which means the model then succeeds in calculating the shaded area and thus in solving the question.

Together, the three skills can be used to solve the entire PAINTING-THE-WALL problem. Their straightforward nature, however, makes it difficult to re-use the skills for other problems. After creating and running this model, it was found that the model could not solve the problems of node 8 by adding only one skill.

To make sure Model 0 could be adapted with one skill to solve the problems of node 8, its skills had to be made more general. Mathematical operations (for this model: addition, subtraction, and multiplication) are turned into their own, separate skills. Iterating over the various items in the visual hierarchy also becomes its own skill. In practice, this means that the `shaded-area`, `area`, and `base` (called `segments` in this generalised version) skills now focus on describing the process through which to get to the shaded-area, area, and base respectively. The skills specify which slots of the visual chunks to focus on, and which steps to take to get to a successful answer. They also specify what to do in case of a failure (i.e. switching to another skill). They rely on the `iterate-over` skill for iterating, specifying only what to iterate to, and on the math skills for doing the mathematical operations (after specifying which numbers to add/subtract/multiply).

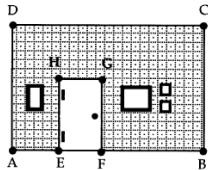
Model 8

The version of Model 0 with more generalised skills works as a basis for Model 8. A comparison of node 0 and node 8 makes it clear that this model contains many transferable skills. For example, it can use its `shaded-area` skill to find

¹See <https://github.com/IDMAkrum> for the full code of both models.

Node 0

PAINTING_THE_WALL: SECTION ONE, #5



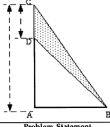
Problem Statement

- The height of a wall is 22.5' and a 7' x 17.5' rectangular door is positioned on the wall such as there is 10' of wall remaining on the left side and 3' of the wall remaining on the right side.
Find the area of the wall to be painted. Do not paint the door.
- The height of a wall is 25.0' and a 8' x 20.0' rectangular door is positioned on the wall such as there is 10' of wall remaining on the left side and 4' of the wall remaining on the right side.
Find the area of the wall to be painted. Do not paint the door.
- The height of a wall is 25.0' and a 8' x 20.0' rectangular door is positioned on the wall such as there is 10' of wall remaining on the left side and 2.5' of the wall remaining on the right side.
Find the area of the wall to be painted. Do not paint the door.

From:
To: 8
0000000

Node 8

TRIANGLE_TRIANGLE: SECTION TWO, #2

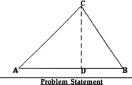


Problem Statement

Triangle ABC and triangle ABD each sharing a right angle located at vertex A, and a base AB. If AB = 49 cm, CD = 24 cm and AC = 79 cm, find the area of the shaded region. Note CDB is a triangle.

Node 8

TRIANGLE_ARC: SECTION TWO, #1

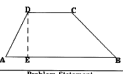


Problem Statement

Given:
In Triangle ABC, segment AB is the base, and segment CD is the altitude (or height).
1. If the measure of segment AB (the base) is 43 cm and the measure of segment CD (the height) is 35 cm, find the area of the Triangle?
2. If the area of Triangle ABC is 2146 square cm and the measure of segment AB (the base) is 56 cm, find the measure of segment CD (the height)?

Node 8

TRAPEZOID_ARC: SECTION THREE, #1

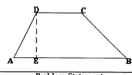


Problem Statement

In Trapezoid ABCD, segments AB and CD are the bases, and DE is the altitude (or height).
1. If the measure of segment DE is 6 cm, the measure of segment AB is 17 cm and the measure of segment CD is 15 cm find the area of the Trapezoid.
2. If the area of Trapezoid ABCD is 423.0 square cm, the measure of segment DE is 9 cm, and the measure of segment CD is 46 cm find the measure of segment AB (the other base).
3. If the area of Trapezoid ABCD is 307.5 square cm, the measure of segment AB is 34 cm and the measure of segment CD is 31 cm find the measure of segment DE (the height).

Node 8

TRAPEZOID_HEIGHT: SECTION THREE, REMEDIAL

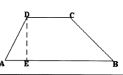


Problem Statement

In Trapezoid ABCD, segments AB and CD are the bases, and DE is the altitude (or height).
1. If the area of Trapezoid ABCD is 472.0 square cm, the measure of segment AB is 31 cm and the measure of segment CD is 28 cm find the measure of segment DE (the height).

Node 8

TRAPEZOID_BASE: SECTION THREE, REMEDIAL

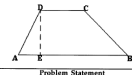


Problem Statement

In Trapezoid ABCD, segments AB and CD are the bases, and DE is the altitude (or height).
1. If the area of Trapezoid ABCD is 1452.0 square cm, the measure of segment DE is 25 cm, and the measure of segment CD is 56 cm find the measure of segment AB (the other base).

Node 8

TRAPEZOID_AREA: SECTION THREE, REMEDIAL



Problem Statement

In Trapezoid ABCD, segments AB and CD are the bases, and DE is the altitude (or height).
1. If the measure of segment DE is 7 cm, the measure of segment AB is 33 cm and the measure of segment CD is 31 cm find the area of the Trapezoid.

From: 0
To: 9
0001000

Figure 4: Node 0 and 8 of the knowledge graph

the shaded-area in the TRIANGLE_TRIANGLE problem. For the problems involving trapezoids, the segments skill can be used to add the two bases that trapezoids have. There are even similarities in calculating the areas of triangles, trapezoids, and rectangles. For all three shapes, a base and height must be multiplied with each other to get to the shape's area. In the case of triangles and trapezoids, however, the answer from the base times height multiplication still needs to be divided by two to get to the final area of the shapes. In fact, adding a division skill to Model 0 should make it capable of solving all node 8 problems.

To create Model 8, a division skill is thus added that works similarly to the addition, subtraction, and multiplication skills. Each of these skills models mathematical operations in the same way: as a simple memorization. Rather than performing a complex calculation, each math skill retrieves a memory chunk from the model's long-term memory. This memory chunk contains the result of the specified operation between two given numbers. The addition skill calls for an addition memory chunk, the subtraction for a subtraction chunk, et cetera. Of course, this is not a realistic representation of mathematical operations. However, in the dataset, no skill difference was found between problems with more complex numbers (e.g. double digits multiplication) compared to simpler numbers. It is assumed that students used a calculator for their mathematics, and therefore, this abstraction is acceptable.

Adding a division skills makes it possible for Model 8 to solve the problems in node 8. However, the skills from Model 0 do not specify how and when to use the division skill. Therefore, it is necessary to further expand the skills of Model

0, so that they can be applied for the new context of dealing with triangles and trapezoids.

Dealing with triangles is relatively easy. It differs from the original area skill only in that the answer from the base times height multiplication needs to be divided by two. Two operators are added to the area skill to specify this: one that says to divide the intermediate answer by two, and one that specifies that the area skill is finished only after this division has taken place. Both operators are specific to triangles and trigger only for triangles. As such, alternative versions of these operators are added to the area skill as well, which trigger only for trapezoids.

In the case of triangles, the two operators, when combined with existing operators, are enough to solve for the area of a triangle. For trapezoids, there is the added complexity of finding the base of the trapezoid. This calculation differs from the others in that it is not considered its own step in the dataset. It is therefore assumed students were not supposed to write down the intermediate answer of adding the trapezoid bases. Model 8 is updated with four operators to accommodate for this.

Firstly, the segment skill is given an alternative success scenario, where it does not write down the intermediate answer but rather adds it to working memory. A new operator in the area skill, in turn, ensures the alternative success scenario is triggered when dealing with trapezoid shapes. If the intermediate answer is stored in working memory, the missing-base-trapezoid operator prevents the model from adding the bases together again, while the base-of-trapezoid operator specifies what the model must

do with the intermediate answer (i.e. multiply it with the trapezoid height).

The addition of in total 8 new operators to the `area` and `segment` skills combined makes Model 8 a degree more complex than Model 0. To ensure Model 8 uses its new operators over some of the older ones, the new operators are given a higher activation. This is done because both models assume that the needed skills to solve the nodes are already mastered. Mistakes are not intentionally modelled and actively prevented where they do naturally occur.

The final models of node 0 and node 8 work according to expectations. Model 0 can solve the `PAINTING-THE-WALL` problem, while Model 8 can solve all problems of node 8, as well as the `PAINTING-THE-WALL` problem. Since it has more operators, Model 8 does require more time to solve the `PAINTING-THE-WALL` problem than Model 0. This difference can be mitigated by running Model 0 before running Model 8. In that case, the skills from Model 0 are more efficient through practice, which Model 8 inherits since it uses those same skills.

Discussion

The construction of the knowledge graph, and the model that solves subsequent problems shows the viability of a hybrid approach to modeling. Instead of constructing a model just on the basis of the intuitions of the modeler, we used a data-driven approach to help partition the model into seven skills, and subsequently started implementing these skills (only finishing the first two).

The knowledge graph

While there is no ground truth to evaluate the knowledge graph with, it is possible to make an intuitive estimate of its appropriateness. In a representative knowledge graph, it is expected that easier problems are placed in the earlier nodes and more difficult problems in the later nodes. There should additionally be some sense to the way the problems are split across nodes, where it should be possible to see the similarities that connect the problems within one node, and the differences between the problems across nodes.

Unfortunately, space limitations do not allow us to discuss all the nodes in the graph in detail, but the final knowledge graph matches intuition in some places and not in others. Problems that are deemed more difficult, like shapes within shapes (e.g. the `TWO-CIRCLES-IN-A-CIRCLE` problem from Figure 2) are placed primarily in node 127. By comparison, problems that are restricted to only one shape are placed in the earlier nodes. The graph additionally shows an intuitive hierarchy between shapes, with square and triangle problems being placed in earlier nodes than circle and pentagon problems.

The graph can, at times, match intuition less well when it comes to the differences and similarities between nodes. While all triangle and trapezoid problems are placed in node 8, circle problems that seem similar are split across nodes 27 and 29, and a problem that asks students to calculate different

triangles within a rectangle has its steps split across four different nodes. Generally, the quality of the knowledge graph differs per node but overall, the resulting knowledge graph makes some intuitive sense.

An elaborate discussion of the knowledge graph can be found in Akrum (2022).

The skills model

The reader may wonder at this point what the model's predictions are, and how they relate to the data. However, that was not the purpose of this model. It shows that it can solve multiple different problems at different skill levels. We found that the ideal that we started with, namely that the skills underlying the graph correspond one-to-one with skills in the model, does not hold in practice. To model the transition from node 0 to node 8, we needed to add the division skill, but we also needed to augment the existing skills to include triangles and trapeziums. While theoretically possible, it is not very likely that the students taking this tutorial were lacking the division skill. Rather, it is just that this skill was not needed for the node 0 problem. More likely, however, it is the augmentation to the skill to calculate the area for triangles and trapeziums (which does include division) that most distinguishes node 0 from 8.

Another possibility is that the transition between nodes involves adding multiple skills. As long as the data cannot distinguish between two skills, the graph algorithm will not identify them. This can be either due to a lack of assignments that separates them, or because not enough students have mastered one but not the other.

Applications in Tutoring

Cognitive tutors are typically based on models that are constructed. Their quality therefore solely depends on the expertise and skill of the modeler. By augmenting this process by a data-driven analysis that guides the model, a more realistic and reliable model can be built that matches skill differences in student data. Such models can provide insight to the teacher and student of the student's current skills, and provide guidance in what the best materials to study and practice with are.

Acknowledgements

This project was funded by the ECP platform voor de Informatiesamenleving.

References

- Akrum, I. (2022). *From Knowledge Graph to Cognitive Model: A Method for Identifying Task Skills*. [Master's Thesis, University of Groningen]. Student Theses Faculty of Science and Engineering. <https://fse.studenttheses.ub.rug.nl/id/eprint/28034>.
- Anderson, J. R., Bothell, D., Byrne, M. D., Douglas, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of mind. *Psychological review*, 111(4), 1036-1060. doi: 10.1037/0033-295X.111.4.1036

- Hoekstra, C., Martens, S., & Taatgen, N. A. (2020). A skill-based approach to modeling the attentional blink. *Topics in Cognitive Science*, 12(3), 1030-1045. doi: 10.1111/tops.12514
- Koedinger, K. R., de Baker, R. S. J., Cunningham, K., Skogsholm, A., Leber, B., & Stamper, J. (2010). A Data Repository for the EDM community: The PSLC DataShop. In C. Romero, S. Ventura, M. Pechenizkiy, & R. Baker (Eds.), *Handbook of Educational Data Mining*. CRC Press.
- Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review*, 120(3), 439 - 471. doi: 10.1037/a0033138

A Pipeline for Analyzing Decision-Making Processes in a Binary Choice Task

Amirreza Bagherzadehkhorsani (amir.bagherzadeh@psu.edu)

Department of Industrial and Manufacturing Engineering, Penn State, University Park, PA 16802 USA

Farnaz Tehranchi (farnaz.tehranchi@psu.edu)

School of Engineering Design and Innovation, Penn State, University Park, PA 16802 USA

Abstract

In this study, we propose a roadmap for the analysis of various factors on cognitive models' parameters and utilizing different cognitive models to better understand the human decision-making process in a binary choice task. Our experiment of a binary choice task is a Biased Coin Flip Game, where users predict the outcome of 150 trials of biased coin flips without knowing the coin's bias. In a previous study, we conducted a factorial ANOVA on the Biased Coin Flip Game to identify factors that significantly influence users' decision-making strategies, such as gender, the presence or absence of the WinRate, and the coin's bias value. In this paper, we employed a Genetic algorithm to identify cognitive models' parameters that fit the users' behaviors the best in scenarios specific to each combination of effective factors. Subsequently, we fitted linear models on cognitive models' parameters to examine the relationship between the identified parameters and the influential factors on decision-making. By analyzing and interpreting the coefficients of these linear models, we aim to gain insights into how these factors affect users' decision-making processes and understand human decision-making better. Our proposed roadmap serves as a valuable resource for researchers who aim to interpret cognitive model parameters in different experimental settings. By providing a systematic approach to investigating the relationships between influential factors and cognitive model parameters, this work provides a deeper understanding of human decision-making processes and baselines for future modeling approaches in this domain.

Keywords: cognitive modeling; ACT-R; decision-making; binary choice task; human performance modeling

Introduction

The human decision-making process is a complicated blend of cognitive, emotional, and environmental factors that has been a topic of interest across multiple disciplines (Gigerenzer & Gaissmaier, 2011; Kahneman & Tversky, 2013). Understanding the underlying principles that shape our decisions can have critical applications in improving individual and social outcomes. One aspect of decision-making that has received considerable attention is the human utility function (Bourgin, Peterson, Reichman, Russell, & Griffiths, 2019; Peterson, Bourgin, Agrawal, Reichman, & Griffiths, 2021; Wang & Ruhe, 2007). It refers to the value that the decision-maker's mind assigns to each potential outcome of a choice in an environment where multiple choices are presented to the decision-maker. The outcome of each option is unknown to the decision-maker. This paper aims to analyze (a) the elements that impact the human utility function, (b) the significance of their impact, and (c) the

decision-making strategies users follow in different settings of a choice task experiment referred to as the "Biased Coin Flip Game." The game consists of virtual coin tosses in which the probabilities of the outcomes are not equal, i.e., the coin is biased. In each trial, the coin flip can result in either "Heads" or "Tails," and users must predict the outcome. The user's choices, Heads and Tails, correspond to the two potential results of the next coin flip which the user aims to predict correctly. This study aims to explore the factors that influence users' decision-making strategies in this binary choice task.

Probability learning is a key component of decision-making. It has been shown that probability learning plays a significant role in an individual's ability to adjust their decisions based on the probabilities associated with each choice (Gallistel, 1990; Rescorla, 1972). The Biased Coin Flip Game leverages this concept by presenting the users with a coin flip task where one side of the coin appears with a higher probability than the other one, which is unknown to the users. Through repeated trials, individuals are expected to learn the bias and adjust their decision-making strategies accordingly (Vulkan, 2000). However, the number of trials needed to learn the bias, and the strategy that users choose to follow are highly susceptible to the visual cues represented to users (Shanks, Tunney, & McCarthy, 2002). To the best of our knowledge, none of these studies have analyzed how different types of feedback affect the utility value that users devote to each option. Hence the development of a cognitive model is necessary to understand how different feedbacks effect users' decision making and how to optimize the design for a desirable behavior by them.

To investigate the influence of visual cues on decision-making strategy and decision time, (Bagherzadeh & Tehranchi, 2023) studied the effect of multiple cues such as (a) the hidden/unhidden win rate indicator (from now on in this paper, we refer to it as WinRate), and (b) hidden/unhidden the five most recent coin results alongside demographic characteristics of users such as (c) gender and (d) Education and experimental settings such as (e) bias value and (f) the initial coin outcomes in the first 20 trials. Utilizing Fractional Factorial Design (i.e., a statistical experiment design method to analyze the significance of each cue), they analyzed the effect of these cues on decision-making strategy and decision-making time. In this work, we investigate how these factors affect the decision-making process of the users. Knowing how different factors affect users' decision-making processes will help us to better understand human cognition,

design better interfaces, and know what to expect from users in different experimental settings.

In recent years, cognitive architectures have emerged as a tool to model and simulate human cognition and behavior in complex tasks, such as probability learning, multi-cue decision-making, and utility function optimization (Anderson, 2009). These architectures offer a comprehensive framework for understanding the cognitive processes underlying decision-making, incorporating the interaction between various cognitive components such as memory, perception, attention, and reasoning. The integration of findings from the fields of psychology, neuroscience, and artificial intelligence makes cognitive architectures a great tool to simulate human behavior and predict decision-making outcomes in a wide range of contexts (Newell, 1990). Modeling users' decision-making process with different cognitive architectures, will help us better understand users' behavior in the presence of different factors and will help designers to know what types of behaviors to expect from users in different settings and designs and consequently, make better design decisions (Peschl & Stary, 1998).

Cognitive architectures provide a great infrastructure for studying probability learning and multi-cue decision-making and can provide valuable insights into the processes that affect the human utility function and decision-making strategy. For example, through the use of cognitive architectures, researchers can simulate the effect of different visual cues on decision-making processes and investigate how users tend to think differently about the expected feedback from choices by modifying their internal utility functions based on the feedback they have received previously. Additionally, cognitive architectures can shed light on the role of various cognitive factors, such as working memory capacity and attentional control, which shape decision-making strategies and optimizes utility functions (Gray, Sims, Fu, & Schoelles, 2006; Marewski & Schooler, 2011). Previously, we used ACT-R and PyIBL cognitive architectures, deep reinforcement learning with an epsilon Greedy decision-making algorithm, and Thompson sampling to model the decision-making process in the Biased Coin Flip Game (Bagherzadeh & Tehranchi, 2022). We compared the versatility of these models in simulating different decision-making behaviors that were observed from users in the literature. We showed that PyIBL and ACT-R are capable of simulating Matching and Maximizing which are the most common behaviors observed from users. However, we did not collect any data to analyze users' behavior in the Biased Coin Flip Game. These findings will not only help us understand how different factors affect decision-making processes but also, will help us understand the differences between ACT-R and PyIBL decision-making processes, and understand which one is more suitable to simulate users' decision-making in a binary choice task.

Incorporating cognitive architectures into our study of the Biased Coin Flip Game can further advance our understanding of the relation between visual cues that are presented to the users and the decision-making strategy.

Furthermore, the use of cognitive architectures can help researchers to identify potential cognitive bottlenecks or biases that may lead to suboptimal decision-making and suggest approaches to minimize the factors that lead to suboptimal decision-making through the design of more effective information presentation strategies—Suboptimal strategy is referred to any strategy that users' choices do not lead to the highest possible outcome.

This study aims to analyze how different cues affect the decision-making strategy of users with different demographic characteristics and suggest ways to optimize decision-making environments suitable for different demographics. By doing so, we hope to provide insights into designing environments that promote optimal decision-making, avoid convergence to suboptimal strategies of decision-making. Identifying what the most effective visual cues are for guiding optimal decision-making can have practical implications in a wide range of fields, including interface design.

Decision-Making in ACT-R

Adaptive Control of Thought-Rational (ACT-R) cognitive architecture is a widely used computational model of human cognition (Anderson, 1990; Anderson & Lebiere, 1998). ACT-R offers a theoretical foundation for understanding the processes involved in human decision-making by simulating cognitive processes such as memory, learning, perception, and problem-solving.

Production rules are the centerpiece of ACT-R architecture. They govern the manipulation of symbolic representations in the users' working memory (Anderson, 2009). The architecture also incorporates a sub-symbolic level, which models the activation and retrieval of declarative knowledge from long-term memory (Anderson & Lebiere, 1998). This dual representation makes ACT-R ideal for the simulation of both rule-based and instance-based decision-making processes (Gonzalez, Lerch, & Lebiere, 2003).

ACT-R has been applied to various decision-making studies, ranging from simple binary choices to complex problem-solving tasks (Ritter, Tehranchi, Dancy, & Kase, 2020). These studies have shown that the ACT-R framework can effectively model human decision-making behavior and provide insights into the underlying cognitive processes (Marewski & Mehlhorn, 2011). For instance, ACT-R has been used to examine the role of cognitive biases in decision-making (Johnson, 2006) and investigate the impact of expertise on decision-making performance (Taatgen, 2013). ACT-R employs a utility-based decision-making algorithm to select the most appropriate production rule in each given situation based on the value of the utility function. The utility function in ACT-R plays a critical role in estimating the expected value of firing a specific production rule, guiding the system toward a course of action (Anderson, 2009).

The utility of a production rule (U_i) in ACT-R is calculated using the following equation:

$$U_i(n) = U_i(n-1) + \alpha [R_i(n) - U_i(n-1)]$$

Where:

- α is the learning parameter

- R_i (n) is the effective reward value given to production i for its n^{th} usage
- U_i (0) is the initial utility value for production i .

The decision-making algorithm in ACT-R operates in a stochastic manner, using a Conflict-Resolution Equation (Bothell, 2017), also known as the Boltzmann equation, to select a production rule based on its utility, probabilistically. The probability P_i of selecting a production rule i is given by the equation:

$$Probability(i) = \frac{e^{U_i/\sqrt{2}s}}{\sum_{j \in m} e^{U_j/\sqrt{2}s}}$$

Where the summation j is over all the productions which currently meet the conditions required. The parameter s is called noise value. However, here, as for the similarity to other models, we refer to it as Temperature. ACT-R multiplies the Temperature value, s by the square root of two (similar to T in the Boltzmann equation).

In order to fit the models' results to users' data, we can adjust the reward values, the learning parameter α , and the temperature value, s . We tune these parameters to minimize the Mean Square Error (MSE). We use the Genetic algorithm (Bozorg-Haddad et al., 2017) to find the combination of parameters that correspond to the smallest MSE.

Decision-Making in PyIBL

PyIBL (Python Instance-Based Learning) is a cognitive modeling framework that provides a computational implementation of Instance-Based Learning Theory (IBLT) (Gonzalez et al., 2003). IBLT is a psychological theory that aims to describe and predict human decision-making behavior in dynamic, complex, and uncertain environments such as the Biased Coin Flip Game. The core idea behind IBLT is that users rely on their past experiences or instances to make choices, rather than following predefined rules or optimization processes. PyIBL is specifically designed to simulate human cognitive processes and help researchers develop and test cognitive models that are grounded in IBLT. The PyIBL framework allows for the creation of cognitive models by incorporating key concepts from IBLT, such as instance storage, retrieval, and adaptation. In this framework, instances are stored in memory as chunks, and decision-making is based on the retrieval of the most relevant chunk(s) from memory, considering similarity and activation values (Lebiere, Wallach, & Taatgen, 1998). By simulating human decision-making and gaining insights into cognitive processes, PyIBL contributes significantly to the fields of cognitive science, artificial intelligence, and human-computer interaction. PyIBL has been developed based on PyACTUp, which is a Python implementation of ACT-R's declarative memory and decision-making process. However, the decision-making process in ACT-R and PyIBL, even though similar, have differences that affect decision-making behavior. We previously explained ACT-R decision-making and now we explain the PyIBL decision-making process and

illustrate the similarities and differences between ACT-R and PyIBL in the decision-making process.

PyIBL uses the concept of blending to calculate the utility value of each choice in its decision-making process (Lebiere, 1999). The blending mechanism consists of base-level activation, weights, utilities, noise, and Temperature.

Activation

The retrieval of an instance from the memory of the PyIBL model relies on its activation value. The activation value is determined by two factors: (a) the frequency and recency of the instance's experience by the model, and (b) how well it matches the attributes of the retrieval target. The activation is computed using the following formula:

$$A_i = B_i + \epsilon_i$$

Where:

- A_i : Activation of chunk i . It is also called "match score" M_i
- B_i : This is the base-level activation and reflects the recency and frequency of use of the chunk. We elaborate on this and how to calculate this more.
- ϵ_i : The noise value.

Base Level

The frequency and recency of the chunk i are described by its base-level activation, B_i , which is influenced by the Memory's decay parameter, d . The base-level activation is calculated based on a function of the time passed since the previous occurrences of i , represented as t_{ij} in the following equation.

$$B_i = \ln \left(\sum_j t_{ij}^{-d} \right)$$

Activation Noise

The activation noise, ϵ_i , implements the stochasticity of retrievals from Memory. It is sampled from a logistic distribution centered on zero. It is normally resampled each time the activation is computed.

Blending

A weight is calculated for chunks using their corresponding activation values to present the contribution of chunks in the blending value.

$$w_i = e^{\frac{A_i}{\tau}}$$

Where:

- w_i : The weight assigned to chunk i .
- A_i : Activation of chunk i . It is also called "match score" M_i
- τ : The temperature value.

With the activation values calculated for all the chunks corresponding to an action, the blending value is calculated as follows:

$$BV = \sum_{i \in m} \frac{w_i}{\sum_{j \in m} w_j} u_i = \sum_{i \in m} \frac{e^{\frac{A_i}{\tau}}}{\sum_{j \in m} e^{\frac{A_j}{\tau}}}$$

Lastly, the action with the largest blending value will be taken. If the outcome is already represented by a chunk, the base-level activation will be updated. If not, a chunk will be

Table 1: PyIBL Models MSE and RMSE in Percentage are compared. The lowest RMSE values (the gray cell) are associated with Memory window size of 4.

Measurement	Memory window size (Considering the number of consecutive Heads)				Memory window size (Considering wins after playing Heads)			
	1	2	3	4	1	2	3	4
MSE	0.0029	0.0025	0.0023	0.0027	0.0038	0.0036	0.0024	0.0020
RMSE*100	5.4006	5.0518	4.7977	5.2251	6.1618	6.0326	4.9575	4.5069

created to represent the outcome in the next blending equation.

Even though blending has been implemented in ACT-R's retrieval time calculation, the blending equation cannot be used in the ACT-R decision-making process and simulating a decision-making process with blending is not possible in ACT-R. We tune the parameters of both cognitive models and analyze if either is simulating users' behavior more accurately and use that model for further analysis.

Methodology

Experiment Description

In this paper, we built upon our previous findings (Bagherzadeh & Tehranchi, 2023). We used 2^{6-2} Fractional Factorial Experimental Design to identify the factors that influence users' decision-making strategy in a Binary choice task. The study examined the impact of two cues (feedbacks), namely the last five results and the WinRate, across two types of settings: (a) the bias value of the coin and (b) Random seed to study the primacy bias in the first 20 trials. Because it can be difficult for users to recover from poor probability learning from initial 20 trial outcomes, in one level of this factor, the Heads and Tails appeared on the virtual coin equally and in another level, Heads appeared on 70 percent (14 out of 20) of trials. The study also accounted for demographic characteristics such as gender and education. We reported that potentially only gender, bias, and WinRate significantly influenced decision-making strategies. As a result, perhaps, cognitive models should only take these factors into account when making decisions. Using these findings, we analyze how these cues affect the decision-making process through cognitive modeling.

Modeling

Unique models to simulate the users' behavior for each combination of the effective factors (gender, bias value, and WinRate) are required. Hence, at least eight models are needed to simulate all the possible combinations of factors. We used two different cognitive architectures: (a) ACT-R and (b) PyIBL. To evaluate our models' performance, we used users' data for every combination of factors which were collected in the previous study (Bagherzadeh & Tehranchi, 2023). For the initial modeling details, please refer to (Bagherzadeh & Tehranchi, 2022). Initially, we adjusted the models' behaviors to match the users in each combination. We attempted to tune two parameters in the decision-making algorithms of both ACT-R and PyIBL: (a) Temperature and (b) noise value alongside the reward values for winning and losing. To find the best parameter values, we employed a Genetic algorithm (algorithm's parameters are: maximum of 2000 iterations, 100 different sets of parameter values per iteration) that minimized the Root Mean Square Error (RMSE). For each set of parameters, we ran the model four times and took the average proportion of Heads in blocks of 10 trials to reduce the impact of randomness and obtain a more reliable estimate of the model decision-making behavior with that parameter set. The Genetic algorithm will be terminated if no improvements observed in RMSE in 200 iterations. If in two hundred iterations of the Genetic algorithm, no improvement is observed in the value of the RMSE, the algorithm is terminated.

None of the models' results aligned with the users' data, resulting in a relatively high RMSE (6.1%). Hence, we turned to the feedback provided by the users after the experiment. They all stated that their decision-making strategy was based

Table 2: ACT-R Models MSE and RMSE in Percentage are compared. The lowest RMSE values (the gray cell) are associated with the Memory size of 3.

Measurement	Memory window size (Considering the number of consecutive Heads)				Memory window size (Considering wins after playing Heads)			
	1	2	3	4	1	2	3	4
MSE	0.0039	0.0036	0.0032	0.0034	0.0039	0.0036	0.0031	0.0034
RMSE*100	6.2449	6.0011	5.6826	5.8686	6.2761	6.0116	5.6134	5.8686

on the number of consecutive wins they had while playing “Heads.” For instance, if they won four times in a row with Heads, they would switch to Tails. However, their decisions after playing Tails was only based on the last result. As Tails is the less likely outcome of the coin flip, the users will only switch to Tails if the outcome has been Heads for multiple trials in a row. Hence, if they win with Tails, they will switch to Heads. And if they lose with Tails, it means that Tails hasn’t been the outcome for even a longer period of time compared to the last coin flip. As a result, they will play Tails again with higher confidence.

Hence in our second modeling attempt, we considered a memory window of four and set different rewards for 1, 2, 3, and 4 consecutive wins with Heads and two reward values for winning and losing with Tails. This is also in line with (Cowan, 2001) that users can hold up to four instances in their minds. Similar to the first attempt at modeling, the reward values were also chosen based on the goodness of RMSE using a Genetic algorithm. In PyIBL, the results were improved significantly across all cases (RMSE = 4.5%). To make sure that users are not utilizing less than four trials as some papers suggest a smaller number for working memory (Brockbank & Vul, 2020), we also developed models with smaller time windows. It resulted in a 10 percent increase in RMSE when the memory window decreased to 3. Lastly, we explored whether users made decisions based on the coin flip results (“Heads” or “Tails”) rather than the outcome (“win” or “lose”). As shown in Table 1, the best performance was achieved by a memory window of four consecutive wins.

In all instances, PyIBL outperformed ACT-R. The ACT-R model that took into account up to 4 consecutive wins exhibited an RMSE of 5.86%. However, by considering up to 3 consecutive wins, the RMSE decreased to 5.61% (refer to Table 2).

The result from this experiment showed that the PyIBL cognitive model of the decision-making process is more representative of users’ behavior than the decision-making process of ACT-R. An example of the PyIBL model with the memory of four compared to users in the case of (Male, no WinRate shown, and bias value of 0.6) is presented in Figure 1.

Sensitivity Analysis

A linear model is fitted where the dependent variables are the PyIBL optimal parameters that we found by Genetic algorithm. Our independent variables are the factors that were identified by (Bagherzadeh & Tehrani, 2023), i.e., (a) gender, (b) WinRate, and (c) bias value. The coefficients of the independent parameters provide valuable insights into how these different factors influence the reward that users expect to receive from choices and decision-making parameters, which consequently, modify the utility value and the decision-making process of different users. The general linear model for predicting the parameters is as follows:

$$Par = C_1x_1 + C_2x_2 + C_3x_3 + C_0$$

The three factors used in the general linear model to predict the decision-making parameters are gender (x_1), the presence

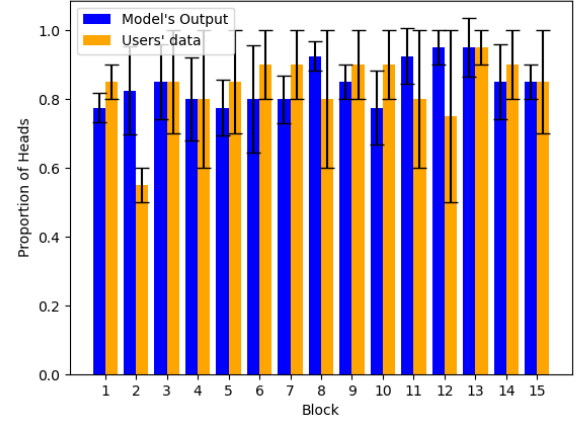


Figure 1: The comparison of PyIBL model with memory window of four wins ($n=4$) with two male participants ($x_1 = 0$), with no WinRate ($x_2 = 0$) and Bias value of 0.6 ($x_3 = 0$) with 95% confidence interval.

or absence of the WinRate (x_2), and the value of bias (x_3). gender is represented using a binary variable (with male=1 and female=0). The presence of the WinRate is also represented using a binary variable, with a value of 1 indicating its presence and a value of 0 indicating its absence. The value of bias is represented using a binary variable, with a value of 1 representing a bias of 0.67 and a value of 0 representing a bias of 0.6. The coefficients C_1 , C_2 , and C_3 were obtained by fitting a linear model to the data and representing the effect of each factor on the decision-making parameter being predicted.

The coefficients were found by minimizing the RMSE of the predicted parameters by the linear model and the parameters found by the Genetic algorithm for each combination of factors. The value of the coefficients of the linear model can be found in Table 3.

Analysis

The coefficients of the factors in the formula for Temperature are found to be significantly smaller than the Temperature intersection C_0 . This suggests that sensitivity of users to the utility function is not affected by the factors considered in this study. The high value of the Temperature parameter indicates that users have a strong willingness to explore different choices and are less focused on valuing feedback based on their frequency of occurrence.

Regarding the rewards, it seems that after a loss with the Heads (zero consecutive wins) and after two wins with Heads, users tend to play Heads again as the reward values are the highest among all cases. The effect of the features on rewards is most evident in three cases: (a) After a win, female users tend to play Heads more often than male users, (b) after one win with Heads, male users are more prone to play Heads, and (c) WinRate shows its effect more evidently after the users have won twice with Heads. If WinRate is shown, the users will be less likely to play Heads. Finally, contradictory to our expectation, after four wins, if the bias is

higher, meaning if the probability of the Heads being the result of the coin flip increases, users are less likely to choose Heads.

Table 3: The coefficient of the parameters for PyIBL models considering up to four consecutive wins. H_i is the reward assigned after winning i , $i \in \{1,2,3,4\}$ consecutive times. H_0 is the reward after losing with “Heads”. T_0 , T_1 are the rewards assigned to winning and losing with Tails.

Parameter	C_1	C_2	C_3	C_0
Temperature	-0.035	-0.071	-0.811	7.825
Decay rate	-0.271	0.174	0.246	1.261
Noise	0.571	-0.640	-0.091	3.158
Default Utility	0.676	0.062	-0.222	3.077
H_0 reward	0.313	0.092	-0.447	2.032
H_1 reward	1.971	0.523	-0.675	1.413
H_2 reward	-1.094	-0.495	0.678	3.802
H_3 reward	2.412	-1.721	-0.668	-1.196
H_4 reward	0.409	0.592	-2.551	-1.071
T_0 reward	-1.516	0.454	-2.423	0.858
T_1 reward	0.775	-0.225	0.549	-0.127

In playing Tails, a loss is more encouraging to users to play Tails again in contrast to winning. This is a logical action. Because the probability of Tails is relatively low and two Tails in a row is an unlikely scenario in users’ minds. Hence, they are more likely to play Heads after winning with Tails. Finally, the noise seems to be affected the most by WinRate. Based on the result, it seems that when WinRate is shown to the users, the users are more likely to explore than exploit. They are likely to try various options even when they anticipate a higher reward from a particular choice. While if they were to exploit, they would exclusively select the option with the highest expected reward. This shows that WinRate induces a sense of insecurity that makes the users think their strategy is suboptimal and they need to exploit other strategies to reach a better Win rate. Also, users tend to try to achieve a higher Win rate value. In other words, users try a wider set of strategies to see if any of their strategies can result in a Win rate closer to 1.0.

Discussion and Future Work

Previously we examined the impact of two distinct cues, namely the last five results and the WinRate, across two types of experiment factors: one with bias values and the other with a random seed to reduce bias in the first 20 trials (Bagherzadeh & Tehranchi, 2023). We also accounted for demographic characteristics such as gender and Age group. We found that only gender, WinRate, and bias value affect the decision-making strategy of the users. Using these findings, here we developed 16 distinct cognitive models to simulate users’ decision-making process with ACT-R and

PyIBL. The developed models produced different behaviors due to differences in the algorithms used to model decision-making in ACT-R and PyIBL. A Genetic algorithm was used to find the models’ parameters that resulted in the smallest RMSE. PyIBL models performed significantly better than ACT-R models as evidenced by lower RMSE values of PyIBL models in comparison to ACT-R models.

This result shows that the blending formula used in the calculation of the PyIBL utility function for each choice is more in line with what has been observed from the users. As a result, we chose PyIBL for a set of linear models where the dependent parameters were the rewards assigned to different outcomes alongside the parameters from the decision-making algorithm, such as Temperature, Decay, and Noise. And the independent parameters were (a) gender, (b) Win rate indicator which in this paper, we referred to it as WinRate, and (c) the bias value of the coin. We also conducted further analysis of the coefficients of these parameters to gain insight into the real-world interpretation of the coefficients and how they should be perceived. The result might not be a total representation of the reality of the effects. However, our work suggests a pipeline of how to analyze the decision-making process of different users in different scenarios. With a large number of users and more parameters to involve in our experiment, there is so much more to learn from the human decision-making process and how differences result in different types of behaviors. More studies and analyses are required to reach the real-world interpretation of these coefficients and how they should be perceived.

References

- Anderson, J. (1990). *The adaptive character of thought*. Hillsdale, NJ: Lawrence ErlbaumAssociates. Inc.
- Anderson, J. (1990). *The adaptive character of thought* 1990.
- Anderson, J., & Lebiere, C. (1998). Hybrid modeling of cognition: Review of the atomic components of thought. In: Erlbaum.[AGB].
- Anderson, J. R. (2009). *How can the human mind occur in the physical universe?* : Oxford University Press.
- Bagherzadeh, A., & Tehranchi, F. (2022). *Comparing Cognitive, Cognitive Instance-Based, and Reinforcement Learning Models in an Interactive Task*. Paper presented at the Proceedings of ICCM-2022-20th International Conference on Cognitive Modeling.
- Bagherzadeh, A., & Tehranchi, F. (2023). *The Analysis of the Effect of Visual Cues in a Binary Decision-Making Environment* Paper presented at the Conference proceedings AHFE.
- Bothell, D. (2017). ACT-R 7 Reference Manual. Retrieved from act-r.psy.cmu.edu/wordpress/wp-content/themes/ACT-R/actr7/reference-manual.pdf
- Bourgin, D. D., Peterson, J. C., Reichman, D., Russell, S. J., & Griffiths, T. L. (2019). *Cognitive model priors for predicting human decisions*. Paper presented at the International conference on machine learning.
- Brockbank, E., & Vul, E. (2020). Recursive adversarial reasoning in the rock, paper, scissors game.

- Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24(1), 87-114.
- Gallistel, C. R. (1990). *The organization of learning*: The MIT Press.
- Gigerenzer, G., & Gaissmaier, W. (2011). Heuristic decision making. *Annual Review of Psychology*, 62, 451-482.
- Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science*, 27(4), 591-635.
- Gray, W. D., Sims, C. R., Fu, W.-T., & Schoelles, M. J. (2006). The soft constraints hypothesis: a rational analysis approach to resource allocation for interactive behavior. *Psychological Review*, 113(3), 461.
- Johnson, J. G. (2006). Cognitive modeling of decision making in sports. *Psychology of sport and exercise*, 7(6), 631-652.
- Kahneman, D., & Tversky, A. (2013). Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I* (pp. 99-127): World Scientific.
- Lebiere, C. (1999). *Blending: An ACT-R mechanism for aggregate retrievals*. Paper presented at the Proceedings of the sixth annual act-r workshop, george mason university, fairfax, va, usa.
- Lebiere, C., Wallach, D., & Taatgen, N. (1998). *Implicit and explicit learning in ACT-R*. Paper presented at the Proceedings of the second European conference on cognitive modelling.
- Marewski, J. N., & Mehlhorn, K. (2011). Using the ACT-R architecture to specify 39 quantitative process models of decision making. *Judgment and Decision Making*, 6(6), 439-519.
- Marewski, J. N., & Schooler, L. J. (2011). Cognitive niches: An ecological model of strategy selection. *Psychological Review*, 118(3), 393.
- Newell, A. (1990). *Unified theories of cognition*, harvarduniv. Press, Cambridge, Mass.
- Peschl, M. F., & Stary, C. (1998). The role of cognitive modeling for user interface design representations: An epistemological analysis of knowledge engineering in the context of human-computer interaction. *Minds and Machines*, 8, 203-236.
- Peterson, J. C., Bourgin, D. D., Agrawal, M., Reichman, D., & Griffiths, T. L. (2021). Using large-scale experiments and machine learning to discover theories of human decision-making. *Science*, 372(6547), 1209-1214.
- Rescorla, R. A. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement. *Classical conditioning, Current research and theory*, 2, 64-69.
- Ritter, F. E., Tehranchi, F., Dancy, C. L., & Kase, S. E. (2020). Some futures for cognitive modeling and architectures: design patterns for including better interaction with the world, moderators, and improved model to data fits (and so can you). *Computational and Mathematical Organization Theory*, 1-29.
- Shanks, D. R., Tunney, R. J., & McCarthy, J. D. (2002). A re-examination of probability matching and rational choice. *Journal of Behavioral Decision Making*, 15(3), 233-250.
- Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review*, 120(3), 439.
- Vulkan, N. (2000). An economist's perspective on probability matching. *Journal of economic surveys*, 14(1), 101-118.
- Wang, Y., & Ruhe, G. (2007). The cognitive process of decision making. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 1(2), 73-85.

Cognitively and Linguistically Motivated Part of Speech Tagging: Quantitative Assessment of a Near Human-Scale Computational Cognitive Model

Jerry T. Ball & Stuart M. Rodgers^a

^aInstitute for Defense Analyses

Abstract

We provide a quantitative assessment of the part of speech tagging accuracy rate of Double R Grammar. Double R is a cognitively and linguistically motivated near human-scale computational cognitive model for the analysis of written English which is focused on the encoding of two key dimensions of meaning: referential and relational meaning. It contains a lexicon which encodes explicit declarative knowledge of words and grammatical constructions, and a procedural memory which encodes implicit knowledge about how to analyze input expressions. With ~100,000 words and multi-word units, the lexicon's size aligns with estimates of the size of the human mental lexicon. On 2 previously unseen sample corpora, the system achieved a **98.50%** part of speech tagging accuracy rate over 2604 tokens. While there are limitations to direct comparisons to competing approaches, the current state of the art for part of speech tagging accuracy over the annotated Penn Treebank corpus is ~98%.

Introduction

We describe a cognitively and linguistically motivated part of speech tagging capability within the written word recognition subcomponent of the computational implementation of Double R Grammar (Ball, in preparation). Double R is a near human-scale computational cognitive model for the grammatical analysis of written English. It is implemented in the ACT-R cognitive architecture (Anderson, 2007; Anderson et al., 2004; Salvucci, 2020), and adheres to well-established cognitive constraints on human language processing. It also adheres to basic principles of cognitive (Langacker, 1987, 1991; Ball, 2007a) and construction (Goldberg, 1995; Sag, 2010, Ball, 2007b) grammar, and is strongly usage based.

The computational implementation of Double R Grammar is approaching the grammatical breadth and accuracy of leading computational linguistic systems. The mental lexicon was developed using a combination of automated and manual techniques and contains ~60,000 words and ~40,000 multi-word units with associated parts of speech and morpho-grammatical features. The ~100,000 words and multi-word units align with numerous estimates of the size of the human mental lexicon (Aitchison, 2003), although we do not claim that Double R's mental lexicon encodes all the knowledge that humans have of words and multi-word units—especially low-level perceptual knowledge and knowledge of fine-grained meaning. The words and multi-word units in the mental lexicon were

mainly borrowed from the COCA corpus (Davies, 2008-), the Penn Treebank corpus (Marcus et al., 1993), and the multi-word corpus of Hartmann, Szarvas & Gurevych (2012), and are assigned a part of speech specific base-level activation based on their frequency of use. The retrieval of lexical items corresponding to input tokens depends on the spread of activation from the lexical, morphological, and grammatical context, and the base-level activation.

Grammatical productions determine how to integrate retrieved lexical items and projected grammatical constructions into grammatical representations. There are ~2500 manually created productions that cover the common grammatical patterns of English. The basic processing mechanism is pseudo-deterministic in that it pursues the single best analysis, but is capable of non-monotonically adjusting to the evolving context. The processing mechanism adheres to two well established cognitive constraints on human language processing: incremental and interactive processing.

We provide a quantitative assessment of the part of speech tagging accuracy rate of the of the computational implementation of Double R Grammar. On 2 previously unseen sample corpora, the system achieved a **98.50%** part of speech tagging accuracy rate over 2604 tokens. Although this accuracy rate is not directly comparable to competing machine learning approaches trained over an annotated corpus, or deep learning approaches trained over big data, the current state of the art for part of speech tagging accuracy is around **98%** for systems trained on the annotated Penn Treebank corpus.

Two key features of the computational implementation are the ability to incrementally improve its capabilities, and the explicit symbolic representation of linguistic knowledge. These features make it possible to build functional systems for new domains by adding domain specific words and grammatical constructions, along with supporting grammatical integration productions to the domain general capability. We demonstrated this capability in the Synthetic Teammate Project (Myers et. al, 2019). We describe ongoing and future research to support further improvement of the part of speech tagging capability.

Implementation Details

In Double R Grammar, lexical knowledge is represented by word plus part of speech chunks that are a combination of word specific information, the part of speech of the word,

and information about the semantic, grammatical and morphological features of the word. Word plus part of speech chunks are the lexical entries in the mental lexicon. The term chunk is used technically to mean a small piece of explicit declarative knowledge. Word plus part of speech chunks are not atomic—they contain internal structure to support the encoding of more than just the word form and its part of speech. Word plus part of speech chunks are also organized into a multiple inheritance hierarchy. At the top two levels, there are 18 parts of speech that correspond closely to the traditional part of speech categories—e.g. noun, verb, adjective, adverb, preposition. More broadly, there are 56 parts of speech for which lexical items exist in the mental lexicon—e.g. personal pronoun, determiner, nominal. These 56 parts of speech provide the basis for the quantitative assessment.

Since the ultimate goal of Double R Grammar is language understanding, we attempt to minimize ambiguity in the assignment of part of speech labels. Rampant ambiguity is pernicious for language understanding. Where there is no difference in meaning across the different uses of a lexical item, there should not be a difference in the part of speech. The syntactic position of a lexical item in a given use is only one factor which is used to determine the part of speech. The likely parts of speech of a lexical item—as encoded in the mental lexicon—are determined by its common uses, since the lexical item will not have occurred frequently enough in an uncommon use to be encoded in the mental lexicon with the part of speech typically associated with the uncommon use. In our view, parts of speech provide coarse-grained information about the meaning and common uses of lexical items. For full meaning determination, coarse-grained parts of speech and grammatical structure need to be supplemented with fine-grained meaning that may or may not be morphologically or grammatically encoded. Whether or not fine-grained meaning is grammatically encoded, it still has the potential to influence grammatical structure.

In Double R Grammar, part of speech tagging occurs within the written word recognition subcomponent of a cognitively and linguistically motivated grammatical analysis mechanism. The cognitive and linguistic principles which underlie the grammatical analysis mechanism and written word recognition are discussed in detail in Ball (in preparation). We mention some of them in this paper.

Cognitively, Double R Grammar is a computational cognitive grammar of written English that details an explicit system of lexical and grammatical representation and processing. Because of the focus on the grammatical encoding of referential and relational meaning, we call the underlying theory Double R Grammar. The computational implementation identifies the referring expressions in the input—e.g. object referring expression or nominal, situation referring expression or clause—and the relationships between these referring expressions—e.g. a transitive verb relating a subject and an object. In Double R Grammar, explicit linguistic knowledge is encoded by lexical items

and grammatical constructions in the mental lexicon. During grammatical analysis, lexical items corresponding to the input tokens are retrieved, grammatical constructions are projected, and retrieved lexical items and projected grammatical constructions are integrated together into explicit grammatical representations that are accessible to conscious awareness and manipulation within ACT-R working memory buffers. The retrieval, projection and integration of lexical items and grammatical constructions is implemented via productions that encode implicit knowledge of grammar within procedural memory. Although productions are implicit in the sense that they are not declarative memory elements that are open to conscious awareness in working memory buffers, they are explicitly represented, and can be examined and manually adjusted.

Double R Grammar adheres to two well-established cognitive constraints on human language processing—incremental and interactive processing (Altmann & Mirkovic, 2009). Double R incrementally analyzes the written linguistic input one word or multi-word unit at a time, using all available lexical, morphological, and grammatical (and eventually fine-grained semantic) information interactively (in parallel) to make the best choice at each choice point. Once a choice is made, it is assumed to be correct, and processing proceeds incrementally forward. However, the subsequent input may require modification of the evolving representation via a non-monotonic mechanism of context accommodation. Overall, the processing mechanism is pseudo-deterministic in that it pursues the single best analysis given the current input and context, but accommodates the subsequent input and context when necessary (Ball, 2011).

Linguistically, Double R Grammar aligns with cognitive (Langacker, 1987, 1991; Ball, 2007a) and construction (Goldberg, 1995; Sag, 2010, Ball, 2007b) grammar, and is strongly usage based. Double R Grammar also adopts and adapts many ideas from traditional grammar as codified in several reference grammars (Huddleston & Pullum, 2002; Quirk et al. 1972, 1985; Givón, 1993). Some aspects of generative grammar—most notably X-Bar Theory (Chomsky, 1970)—are also adapted (Ball, 2005).

Grammatical analysis begins with the submission of an unedited written input that may range in size from a single word to an entire corpus of text, subject to computer system performance limitations. The input is incrementally analyzed one word or multi-word unit at a time, and a grammatical representation is generated. Grammatical representations consist of multiple integrated declarative memory chunks.

During incremental grammatical analysis, the written word recognition subcomponent analyzes the next four space delimited input tokens and attempts to retrieve a lexical item in the mental lexicon that matches the first and zero or more of the remaining three input tokens. Retrieval of a matching lexical item is biased by the grammatical, morphological, and lexical context. If retrieval succeeds, the retrieved lexical item takes part in grammatical analysis.

If retrieval fails, the first input token is subjected to further analysis. If this token consists of only alphabetic characters with zero or more hyphens or underscores, the input token is analyzed as an unknown word, since it would otherwise be in the mental lexicon. Unknown words are first analyzed morphologically to identify any prefixes or suffixes. If a prefix or suffix is identified, the remainder of the token is analyzed to see if there is a matching lexical item. Some spelling adjustment may be needed to identify the remainder as a lexical item—e.g. *intensifier* → *intensify* + *ier*. If there is a matching lexical item corresponding to the remainder or base, it is retrieved and a copy of the lexical item is made. The copy is updated to incorporate information about the prefix or suffix, and a new word plus part of speech chunk is created. The newly created word plus part of speech chunk participates in lexical and grammatical analysis. If the remainder does not correspond to a lexical item, the grammatical, morphological, and orthographic context is used to bias retrieval of an unknown word plus part of speech template. The retrieved word plus part of speech template lacks word specific information and has default values for grammatical features. If the retrieved template is a proper noun, the input token is treated as an unknown proper noun and a new word plus part of speech chunk is created and used in grammatical analysis. If the unknown word plus part of speech template is not a proper noun, the input token is subjected to spelling correction using the edit distance algorithm. A cohort of spelling correction candidates is identified, morphological biases relevant to any identified prefix or suffix are set, orthographic biases are set, and the spelling correction candidate with the highest activation is retrieved and used in subsequent grammatical analysis. A copy of the retrieved part of speech template is also updated based on the original input token, and a new word plus part of speech chunk is created and stored in declarative memory where it can be accessed, if the input token occurs again. This provides the computational implementation with a double shot learning capability following spelling correction. If no spelling correction candidate is identified, the retrieved part of speech template is updated with information from the input token and a new word plus part of speech chunk is created. In this case, single shot learning occurs since this new chunk is used immediately in grammatical analysis.

The Quantitative Assessment

To determine part of speech tagging accuracy, it is first necessary to determine the number of tokens for which parts of speech must be assigned. Unlike the annotated Penn Treebank corpus which is pre-tokenized, determining the number of tokens for an unannotated corpus is more difficult. Although the most straightforward measure of token size would be the number of space delimited tokens, this measure is inadequate in several respects. In the unannotated sample corpora used in this study, there are numerous space delimited tokens that contain punctuation that is concatenated with what would otherwise be space

delimited word tokens (e.g. *world—replete*, *'Teddy,'*). For measurement purposes, we split off punctuation and treat punctuation characters as separate tokens, unless the punctuation characters are part of a word (e.g. *P.I.*). We also split off clitics like *'s* (e.g. *John's*) and *'ll* (e.g. *we'll go*). These adjustments align with the way tokenization works in the Penn Treebank corpus. Familiar compound words that are in the mental lexicon may be hyphenated (e.g. *state-of-the-art*), space delimited (e.g. *state of the art*), or concatenated (e.g. *nonetheless*). Should the tokens that match subparts of compound words be treated as separate tokens? The Penn Treebank treats space delimited compounds as multiple tokens, but treats hyphenated and concatenated compounds as a single token. This can lead to odd tokenizing behavior. For example, in the expression *a new generation of bin Ladens*, the compound proper noun *bin Ladens* would be tokenized as *bin* and *Ladens*. This complicates recognition of the familiar compound proper noun *bin Laden* to which the plural suffix *s* is atypically attached. Instead of relying on ambiguous spaces to delimit tokens, we use the linguistically motivated concept of a free morpheme. A free morpheme is capable of standing on its own as a space delimited token, but may also be part of a compound word. Each free morpheme is counted as a separate token. Within compounds, we do not count the hyphen or space as a separate token. We also do not count bound morphemes—including prefixes and suffixes—as separate tokens. In the case of *bin Ladens*, which is not in the mental lexicon, we treat it as a two token compound proper noun with the plural *s* suffix attached to the rightmost token. For *bin Ladens* to be recognized, the plural *s* suffix must first be removed so that the compound proper noun *bin Laden* can be retrieved from the mental lexicon. Once retrieved, the plural *s* can be reattached and a new compound proper noun token for *bin Ladens* can be created.

For each retrieved lexical item, it is necessary to determine if the assigned part of speech is preferred in the grammatical context. Since the Penn Treebank corpus is annotated, it is possible for systems trained on this corpus to determine this automatically. For unannotated corpora, some other mechanism is needed. We do this assessment manually using sampling techniques. For multi-unit words, we count each free morpheme separately for part of speech tagging accuracy purposes. For *bin Ladens* which is categorized as a plural proper noun, we count this result as two correctly tagged free morphemes *bin* and *Ladens*, even though *bin Ladens* corresponds to one lexical entry.

For some words in some contexts, there is ambiguity in the part of speech assignment that cannot be grammatically resolved. For example, if a word can be both an adjective or a noun, either one may be preferred—depending on fine-grained meaning—when the word functions as a modifier in a nominal—e.g. *a sterling* (adj) *example* vs. *a sterling* (noun) *spoon*. Words describing colors can atypically be used with or without a nominal head, without an obvious difference in meaning—e.g. *I like the red* vs. *I like the red one*. For color words, we categorize them as composite

adjective nouns—i.e. a hybrid part of speech category that inherits from both the higher level adjective and noun categories. The existence of composite categories is a key feature of Double R Grammar which is important for minimizing ambiguity. The existence of composite categories follows from the multiple inheritance hierarchy of parts of speech. If a part of speech category has multiple parents, it is a composite category. Words like *fast* are categorized as composite adjective adverbs since they can function as both nominal and verbal modifiers without a difference in meaning—e.g. *the fast car* vs. *he ran fast*. The existence of composite categories minimizes ambiguity and facilitates part of speech tagging accuracy. Composite categories make most sense for closely related categories like adjective and adverb. A composite category like noun verb would be helpful for minimizing ambiguity in the use of words like *zoning* in *the zoning* (noun?) *of the section* vs. *they are zoning* (verb) *the section*, but these two categories are semantically very different and it is unclear what the meaning of a composite noun verb category would be. There is no notion of composite categories or multiple inheritance in the Penn Treebank tagset which provides a flat listing of atomic parts of speech, although there is some implicit hierarchy—e.g. NN (noun, singular) → NNS (noun, plural) | NNP (proper noun, singular) and NNP → NNPS (proper noun, plural).

For the quantitative assessment, we collected a sample of previously unseen, unedited and unannotated texts and grammatically analyzed them. The sample includes a few paragraphs from a Clive Cussler novel, available on line, and paragraph length abstracts of 25 books on the topic of spy novels from www.smashwords.com. We chose to use book abstracts, since each abstract is about a different book with a different set of characters and subject matter. However, since the abstracts were not annotated with parts of speech, they needed to be small enough in size to make manual assessment feasible. On this previously unseen corpus, the computational implementation correctly tagged 1810 out of 1838 tokens for an accuracy rate of **98.48%**. We submitted a second corpus of 8 abstracts of books on the topic of self-help and two political biographies, and the computational implementation correctly tagged 755 out of 766 tokens for an accuracy rate of **98.56%**. This accuracy rate was achieved even though only **97.65%** of the input tokens matched a lexical entry in the mental lexicon. Combining the two samples, the computational implementation correctly tagged 2565 out of 2604 tokens for an accuracy rate of **98.50%**. We computed the 95% confidence interval around 98.50% as ranging from 97.95% to 98.90%. We plan to make the quantitative assessment details available—including a word by word analysis—on the ACT-R web site at <http://act-r.psy.cmu.edu/publication/> and in Ball & Rodgers (forthcoming).

Comparison to Other Approaches

We believe that the accuracy rate reported in this paper is state of the art for a cognitively and linguistically motivated

approach that does not use machine learning over an annotated corpus, or deep learning over big data. Basically, it is state of the art for an approach that combines explicit symbolic representations operating over a probabilistic substrate, but does not use machine learning to learn the probabilities or production rules. We make this claim because we are unaware of any large-scale symbolic system, with or without probabilities, that does better. Perhaps the closest in performance is the rule-based Brill part of speech tagger (Brill, 1992) which achieved a part of speech tagging accuracy rate of around 95% on the Brown corpus (a precursor to the Penn Treebank corpus), and was used in the part of speech annotation of the Penn Treebank corpus. The Brill POS tagger first creates a baseline by identifying the highest frequency part of speech for each known word in a corpus. For unknown words, if the word is capitalized it is treated as a proper noun. Otherwise, it is treated as a noun. The creation of this simple baseline achieves a part of speech tagging accuracy of ~90%. Following creation of the baseline, rules for adjusting the part of speech of each word are iteratively applied until the tagging accuracy stabilizes. These rules can either be manually created or learned via machine learning. As an example, there is a rule that changes the part of speech of a word initially categorized as a past tense verb (VBD) to past participle (VBN), if it follows an auxiliary verb—e.g. *the ball was kicked*.

The tagging accuracy rate of the Brill POS tagger reaches asymptote at ~95% on the Brown corpus. Interestingly, the rules that adjust the part of speech are not associated with probabilities, even though machine learning may have been used to create them. Instead the iterative application of all the rules is repeated until further iterations reach asymptote. Since the rules are ordered, inappropriate serial order effects are likely as in the categorization of *airspeed* as a past tense verb (VBD) because of the *-ed* ending.

Like the Brill tagger, the computational implementation of Double R Grammar has productions that can non-monotonically change the part of speech of a word in context. For example, there is a production that changes the part of speech of a past tense verb to past participle in the context of a regular (i.e. non-modal) auxiliary verb, similar to the rule used by the Brill tagger. This production was added to handle the case where only the past tense verb is in the mental lexicon and necessarily gets retrieved even in the context of a regular auxiliary verb. But the Double R Grammar production has access to more of the grammatical context than the Brill tagger rule. For example, in the expression *the soccer player that could kicked the ball*, the past tense verb *kicked* follows the modal auxiliary verb *could*, but should not be recategorized as a past participle as the Brill tagger rule suggests. Surface position is not always indicative of grammatical constituency or function, and modal auxiliaries like *could*, unlike regular auxiliaries like *be* and *have*, are not typically followed by either past tense or past participle verbs. As another example, in the expression *John's sad story*, there is a production that

overrides the initial treatment of the adjective *sad* as a predicate (i.e. *John's sad*) to be a modifier within a predicate nominal when the noun *story* is processed.

We can view the computational implementation of Double R Grammar as a rule-based system—like the Brill part of speech tagger—but with probabilistic utilities assigned to the production rules. If we can demonstrate that our rule-based approach achieves a state of the art capability with respect to machine learning approaches, then we can cite Rule 1 of the Google Machine Learning Handbook in support of our approach: “Don’t be afraid to launch a product without machine learning...If machine learning is not absolutely required for your product, don’t use it until you have data” (Google Machine Learning Handbook, downloaded 2023). For part of speech tagging of unannotated text, it is unclear where that data would come from.

With respect to machine learning approaches trained over the annotated Penn Treebank corpus, the current state of the art is ~98% tagging accuracy (POS Tagging state of the art, downloaded 2023). Unfortunately, the state of the art percentages displayed on this website are not directly comparable to our results. For one, Double R Grammar uses a different set of part of speech tags organized into a multiple inheritance hierarchy. Whereas the Penn Treebank tagset consists of 36 parts of speech, Double R Grammar includes 56 parts of speech for which at least one lexical item exists in the mental lexicon. Whereas word plus part of speech chunks in Double R Grammar are non-atomic and have internal structure, Penn Treebank part of speech tags are atomic, without internal structure. Whereas the underlying theory of parts of speech in Double R Grammar avoids having multiple parts of speech for a word, where the meaning of the word does not differ in different contexts, words are assigned different parts of speech in the Penn Treebank based primarily on syntactic position, often ignoring similarity in meaning. For example, in the expressions *the bull is running*, *the running bull* and *the running of the bull*, the word *running* is categorized as a present participle verb in Double R Grammar. Since there is no difference in meaning across these uses, and since the meaning and most common use of *running* is as a verb, its categorization as an adjective or gerund on the basis of syntactic position is unwarranted. Instead of changing the part of speech of the word *running* based on a particular use, we allow the verb *running* to exhibit different grammatical functions in different contexts. The syntactic position of a word is only one factor used to determine its part of speech. Although it is possible to map Double R Grammar parts of speech to Penn Treebank tags for comparison purposes, it is more difficult to reconcile theoretical differences in part of speech assignment. For example, the Penn Treebank tags *running* in *people are running* (RB) *scared* (VGN) as an adverb (RB)—perhaps because it functions as a modifier of the head verb *scared*. In Double R Grammar, *running scared* is a familiar multi-word verb. Within this multi-word verb, the functional

status of *running* and *scared* is unclear. More generally, Double R Grammar makes extensive use of space delimited multi-word units. Multi-word units facilitate processing and reduce ambiguity. The achievement of >98% tagging accuracy is dependent on the inclusion of familiar space delimited multi-word units in the mental lexicon. Additional differences include the pre-tokenization of the input in the Penn Treebank corpus and the pre-identification of sentence boundaries. The computational implementation of Double R Grammar accepts unedited textual input without pre-tokenization or pre-identification of sentence boundaries. Since the computational implementation is not trained on a particular corpus, its performance generalizes to previously unseen corpora. Any comparison of the performance of the computational implementation of Double R Grammar to a machine learning approach should consider the ability to generalize to new unannotated corpora, in addition to considering performance on the held out testing portion of the annotated corpus. When evaluating performance on an unannotated corpus, use of sampling techniques for measurement purposes becomes important.

In addition to part of speech tagging accuracy, it is possible to compare the computational implementation with machine learning approaches in computational terms like part of speech tagging speed, memory usage, and CPU usage. An important goal of the computational implementation is to be able to process language in near real-time on available hardware in order to support the development of functional systems capable of interacting with humans in written English. In the synthetic teammate project, we achieved that goal (Myers, et al. 2019). However, machine learning based part of speech taggers, once trained, are much faster at part of speech tagging than the computational implementation.

The computational implementation is designed to be incrementally improved by the addition of new lexical items, new parts of speech, new grammatical constructions, and new grammatical productions. Incremental improvement of machine learning systems trained on an annotated corpus is more difficult. At a minimum, it is necessary to annotate a new (perhaps smaller) corpus to support machine learning over a new domain. Deep learning approaches, which do not rely on an annotated corpus, may be able to overcome this challenge.

Linguistic knowledge in Double R Grammar is explicit. The knowledge encoded in machine learning systems, and especially deep learning systems is largely implicit. Because of the implicit nature of knowledge in such systems, they are difficult to modify, and their internal behavior is difficult to explain. The explicit nature of knowledge in Double R Grammar makes it possible to make explicit changes and test and explain the results. However, since Double R Grammar is highly interactive, it is difficult to keep in mind all the interactions when making a change. This makes regression testing following a substantive change very important. The ability to make an explicit change and test to see the result is an important

advantage of our computationally grounded approach over non-computational approaches. If the explicit change results in previously unseen errors, the change can be modified to correct the errors. While the output of machine learning approaches can also be analyzed, when errors occur it is often not possible to correct the error, since linguistic knowledge is only implicitly represented, especially in deep learning systems.

Future Directions

The quantitative assessment of the part of speech tagging accuracy rate of the computational implementation has revealed the importance of lexical entries in the mental lexicon to part of speech tagging accuracy. The primary source of part of speech tagging errors is the absence of an appropriate lexical entry. In the most common case, the lexical item is either missing (i.e. the word is unknown), or the part of speech assigned to a lexical item is incorrect for a particular use. As a result of the use of automated techniques to create the mental lexicon (Freiman, Rodgers & Ball, 2008), a sizeable number of lexical items were either assigned incorrect parts of speech, or parts of speech that are incompatible with Double R Grammar. Since the computational implementation can be incrementally improved, missing lexical items can simply be manually added or extended to new parts of speech, and incorrect parts of speech can be corrected. We are currently updating the mental lexicon manually to minimize such errors.

A secondary source of errors results from part of speech ambiguity. Many words are associated with multiple parts of speech and meanings. Resolving the ambiguity of genuinely ambiguous words is a significant challenge. We currently use the lexical, morphological and grammatical context to jointly resolve ambiguity via base-level and spreading activation during lexical retrieval. We are not currently able to use fine-grained meaning or low-level perceptual knowledge—including phonological knowledge—to resolve lexical ambiguity. However, the encoding of frequently occurring multi-word units in the mental lexicon facilitates ambiguity resolution, since multi-word units are less ambiguous than individual words. Although the ~100,000 words and multi-word units in the mental lexicon align with numerous estimates of the size of the human mental lexicon (Aitchison, 1993), it is clear from testing that many familiar words and multi-word units are still missing, and some existing words are unfamiliar. We plan to continue to add familiar words and multi-word units to the mental lexicon so that our claims to have a near human-scale mental lexicon will be better supported, although we do not claim that the lexical entries contain all the lexical knowledge encoded by humans.

A third source of errors is due to incorrect tokenization, whether that results from the failure to split a complex space delimited token appropriately, or the failure to recognize a familiar multi-word unit. The computational implementation uses regular expressions—bypassing low-level perceptual analysis—to further tokenize space

delimited input tokens which do not match any lexical item. This tokenization capability can be further improved.

A fourth source of errors is the input itself. These include misspellings of words—e.g. *teh* for *the*, word substitution errors—e.g. *it is **and** apple*, inappropriate concatenations—e.g. *the **airspeed** **isstable***, and nonce expressions like *the paperboy **porched** the newspaper* which use familiar words in novel ways (Clark & Clark, 1979). The occurrence of unknown words is also a potential source of errors. The mechanisms for spelling correction and the handling of concatenation errors, nonce expressions and unknown words can all be improved. However, we do not currently have a mechanism for correcting word substitution errors. We also have not yet developed a suitable theoretical basis for the incorporation of fine-grained meaning.

We plan to evaluate the performance of the computational implementation on samples from the test section of the Penn Treebank corpus. Since we will not be using the Penn Treebank tagset, manual evaluation of the results will be necessary, but use of the Penn Treebank corpus should provide a better basis for comparison to competing machine and deep learning approaches which utilize this corpus. For this purpose, we first plan to add any missing vocabulary from the training sections of the corpus, before running samples from the test sections through the computational implementation. We are looking for suitable metrics to support direct comparison to machine learning and deep learning approaches. Commonly used metrics like precision and recall—which rely on a fully annotated corpus—are not available when evaluating performance on an unannotated corpus, or using a different tagset than the annotated corpus. Following creation of a concordance for the Penn Treebank corpus, we plan to compare the tagging of a sample of words like *running* as part of the evaluation.

The representation and integration of the grammatical context is an important factor in the part of speech tagging accuracy. Although the grammatical analysis capabilities have been under development for many years, further improvement is still needed. With respect to part of speech tagging, we plan to continue to add productions that adjust the part of speech and/or grammatical features of lexical items in appropriate grammatical contexts. We have already mentioned the capability to type shift a past tense verb to past participle in the context of a regular auxiliary verb. More generally, when the right context is definitive, since it is not incrementally available, context accommodation in the form of type-shifting or overriding is needed.

One direction we do not intend to pursue is the use of big data to identify very infrequently occurring words. Although it is likely that this would improve performance, this approach lacks cognitive motivation. Of course, when the computational implementation is used in a new domain, addition of domain specific vocabulary and supporting grammatical productions is needed. The ultimate goal is the attainment of a human-scale domain general mental lexicon, supplemented with domain specific vocabulary for specific applications.

References

- Aitchison, J. (2003). *Words in the Mind: An Introduction to the Mental Lexicon*, 3rd Ed. NY: Basil Blackwell.
- Altmann, G. & Mirkovic, J. (2009). Incrementality and Prediction in Human Sentence Processing. *Cognitive Science*, 222, 583-609.
- Anderson, J. (2007). *How Can the Human Mind Occur in the Physical Universe?* NY: Oxford University Press.
- Anderson, J., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111, 1036-1060.
- Ball, J. (2023, preprint). Chapter 4: Written Word Recognition. https://www.researchgate.net/publication/353560016_Double_R_Grammar_Book_Chapter_4_Written_Word_Recognition
- Ball, J. (2023, in preparation). Double R Grammar, the Grammatical Encoding of Referential and Relational Meaning. Preprint available at <https://www.researchgate.net/profile/Jerry-Ball/research>
- Ball, J. (2011). A Pseudo-Deterministic Model of Human Language Processing. In L. Carlson, C. Hölscher, & T. Shipley (Eds.), *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, 495-500. Austin, TX: Cognitive Science Society.
- Ball, J. (2007a). A Bi-Polar Theory of Nominal and Clause Structure and Function. *Annual Review of Cognitive Linguistics*, 27-54. Amsterdam: John Benjamins.
- Ball, J. (2007b). Construction-Driven Language Processing. In S. Vosniadou, D. Kayser & A. Protopapas (Eds.) *Proceedings of the 2nd European Cognitive Science Conference*, 722-727. NY: LEA.
- Ball, J. (2005, unpublished). Towards a Semantics of X-Bar Theory. https://www.researchgate.net/publication/2928693_Towards_a_Semantics_of_X-Bar_Theory
- Ball, J. & Rodgers, S. (forthcoming). Cognitively and Linguistically Motivated Part of Speech Tagging: Detailed Quantitative Assessment of Part of Speech Tagging in Double R Grammar.
- Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*. Association for Computational Linguistics, Stroudsburg, PA, 152-155.
- Chomsky, N. (1970). Remarks on nominalization. In R. Jacobs & P. Rosenbaum (eds.), *Readings in English Transformational Grammar*, 184-221. Waltham, MA: Ginn.
- Clark, E. & Clark, H. (1979). When Nouns Surface as Verbs. *Language*, 55(4), 767-811.
- Davies, M. (2008-) *The Corpus of Contemporary American English (COCA): 560 million words, 1990-present*. Available online at <https://corpus.byu.edu/coca/>.
- Freiman, M., Rodgers, S. & Ball, J. (2008, unpublished). Building a Functional Mental Lexicon.
- Givon, T. (1993). *English Grammar A Function-Based Introduction, Volumes 1 and 2*. Amsterdam/Philadelphia: John Benjamins Publishing Company.
- Goldberg, A. (1995). *A Construction Grammar Approach to Argument Structure*. Chicago: The University of Chicago Press.
- Google Machine Learning Handbook (downloaded, 2023). <https://developers.google.com/machine-learning/guides/rules-of-ml>.
- Hartmann, S., Szarvas, G., & Gurevych, I. (2012). Mining Multiword Terms from Wikipedia. In Pazienza, M. & Stellato, A. (eds.) *Semi-Automatic Ontology Development: Processes and Resources*. Information Science Reference.
- Huddleston, R. & Pullum, G. (2002). *The Cambridge Grammar of the English Language*. Cambridge, UK: Cambridge University Press.
- Langacker, R. (1987, 1991). *Foundations of Cognitive Grammar*, Vols 1 and 2. Stanford.
- Marcus, M., Santorini, B. & Marcinkiewicz, M. (1993). Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2): 313-330.
- Myers, C., Ball, J., Cooke, N., Freiman, M., Caisse, M., Rodgers, S., Demir, M. & McNeese, N. (2019). Autonomous Intelligent Agents for Team Training: Making The Case for Synthetic Teammates. *IEEE Intelligent Systems*, 34, 3-14.
- POS Tagging State of the Art (downloaded, 2023). [https://aclweb.org/aclwiki/POS_Tagging_\(State_of_the_art\)](https://aclweb.org/aclwiki/POS_Tagging_(State_of_the_art))
- Quirk, R., Greenbaum, S., Leech, G. & Svartvik, J. (1985). *A Comprehensive Grammar of the English Language*. Essex, England: Longman Group.
- Quirk, R., Greenbaum, S., Leech, G. & Svartvik, J. (1972). *A Grammar of Contemporary English*. Longman Group.
- Sag, I. (2010). Sign-Based Construction Grammar: An informal synopsis. In H. Boas & I. Sag (Eds.), *Sign-Based Construction Grammar*. Stanford: CSLI.
- Salvucci, D. (2018). *Java ACT-R, a Java Simulation & Development Environment for the ACT-R Cognitive Architecture*. <http://cog.cs.drexel.edu/act-r/>.

Role Stability and Team Performance in a 4-Player Cooperative Cooking Game

Sounak Banerjee (baners8@rpi.edu)

Cognitive Science Department, Rensselaer Polytechnic Institute

Wayne D. Gray (grayw@rpi.edu)

Cognitive Science Department, Rensselaer Polytechnic Institute

Abstract

The Cooperative Action Task (CAT) is a platform for studying the development of team coordination in complex dynamic task environments. Teams of four cooperate to play a cooking video game across eight 1-hr sessions. Team members communicate using gaze cursors that display the gaze location of each player. Team coordination in the game is achieved through a combination of planned and adaptive actions. Planned actions involve players acting according to pre-assigned roles to reduce behavioral variability, while adaptive actions are characterized by dynamic adaptations to changing task demands. The results of the study reveal that strategic reduction of behavioral variability was beneficial to game performance for all teams. Additionally, team performance was lower when teams switched between strategies across games in the same kitchen.

Keywords: Action Coordination; Games; Teams; Complex Task; Coordination Strategy; Team Roles; Planning; Adaptation

Introduction

There has been a rising interest in team research among cognitive scientists due to its significance in virtually every form of human coordination. However, analysis of team behavior in complex tasks can be challenging due to the dynamic nature of human interactions. Simulated virtual environments are an excellent tool for such analyses because they offer the complexity of naturalistic tasks while ensuring sufficient control over the task environment (Elliott et al., 2017; Cooke, Rivera, Shope, & Caukwell, 1999).

Computer games are excellent simulations for studying complex human behavior, especially expert behavior and task learning (Gray, 2017). For example, Tetris based studies have shed light on the various advanced strategies that experts use in the game and their implications on human learning (Gray & Banerjee, 2021; Sibert, Gray, & Lindstedt, 2020; Lindstedt & Gray, 2013). Others focused on differences in cognitive abilities among novices and experts (Large et al., 2019; Green & Bavelier, 2003).

For the current study, we developed a cooperative cooking game called “The CAT”; that is, the Cooperative Action Task. Here, the CAT was used to explore the development of team coordination (in 4-player teams) across eight 1-hour gameplay sessions. The experimental setup was further designed to enforce restrictions on communications within the team: players were prevented from verbally communicating

with other team members during gameplay. However, players were allowed to communicate through a gaze-based communication system.

Current literature on cooperative behavior in humans reveals that coordinating humans rely on strategic reductions in action variability to improve action predictability for partners when communication is limited. One such study explored behavior in coordinating dyads in an action synchronization task, where access to information about partner’s actions was limited (Vesper, Schmitz, Sebanz, & Knoblich, 2013). The authors discovered that subjects reduced action variability and improved coordination by speeding up their movements.

In the current study, teams reduced behavioral variability of players by assigning roles to its members. The teams used this strategy to compensate for the lack of a rich communication channel. To test for player persistence in sticking to assigned roles and its effect on team performance: we define ‘Role Stability’ (RS)—a measure of a player’s tendency to stick to a certain role for the duration of a game. Results show that reduction of behavioral variability through adherence to player roles did improve team performance.

Methodology

Experimental Setup

The setup for the experiment is illustrated in Figure 1. It includes 5 computers, 4 eye-trackers (each attached to a monitor), 4 Xbox controllers, and 4 acoustic pods. Each pod had one controller, one eye-tracker, and one monitor inside. All 5 computers were set up outside the pods.

One of the 5 computers was used to run the game (the central node), and the video output for this computer was mirrored across all 4 monitors using an HDMI splitter. This meant all four players simultaneously received the same video stream for the game inside each pod. Since the game ran on the central node, the telemetry data (player actions and game state information) was also locally stored there at 60 frames per second (60Hz). The remaining 4 computers (edge nodes) were each connected to one of the 4 eye-trackers and placed outside the pods. Each edge node collected gaze data from the connected eye-tracker, stored the data locally, and sent it to the central node over the Local Area Network.

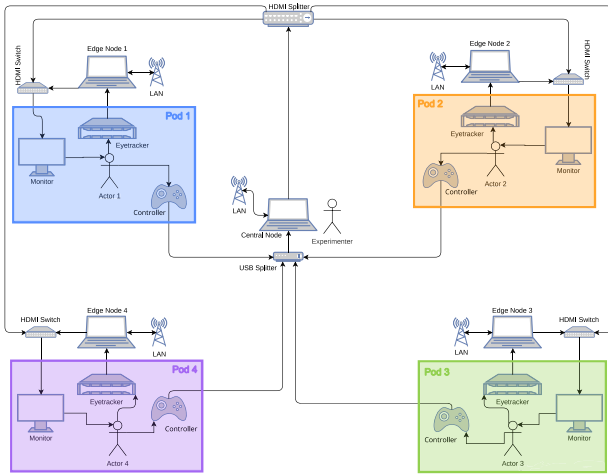


Figure 1: Layout of the experimental setup for the CAT. The 4 translucent regions correspond to the 4 pods in the setup. All entities enclosed within each one of the translucent regions represent the content of the pods.

The Cooperative Action Task

We present the Cooperative Action Task (The CAT), a game-based experimental paradigm (developed using the Unity game engine) to study human coordination within small 4-person teams in a controlled virtual environment. The goal of each team is to work together inside a virtual kitchen to prepare and deliver orders on time.

Orders appear at the top of the game interface, along with a timer indicating the time remaining to prepare the order. The example in Figure 2 presents two outstanding orders, a mushroom soup (expires in 35 seconds) and an onion soup (expires in 65 seconds). Players execute a series of actions to prepare each order as they come in. For example, to prepare the mushroom soup (from Figure 2) players from the team would have to chop three mushrooms and one onion (at the chopping counters), cook them in a pot (on a stove), plate the soup and carry it to the delivery zone. A dirty plate appears on the plate holder 10 seconds after each delivery. Players must then wash the dirty plate at the sink to prepare for the next order. Progress bars are used to indicate the progress of the cooking, chopping, and washing processes. Finally, if an item burns from being left on the stove too long, players need to dispose of it in the trash.

In the current version of the system, only gaze-based communication within teams was allowed during gameplay. To eliminate the possibility of any verbal communication, each player was placed in individual acoustic pods. Point-of-gaze was indicated using translucent disc-shaped gaze-cursors on the game interface, one corresponding to each player (see ‘Gaze cursor’ labels in Figure 2). Every player could see all four gaze cursors on their screen, thus giving each team member access to others’ gaze locations. Further, players could also draw attention to their own gaze cursors by making their

cursors pulse rapidly for half a second; this could be achieved by pressing a button on their controller.

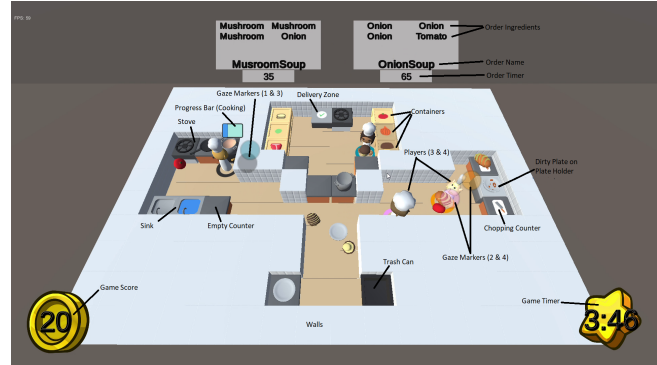


Figure 2: A (labelled) screenshot of a game in the ‘Clover’ kitchen layout. In this layout, the player at the top is locked out of the rest of the kitchen and is the only player with access to raw ingredients and the delivery zone.

Each game is a combination of a kitchen layout and an order list. Kitchen layouts are task environments that present unique challenges to team coordination, while order lists are used to tune the task’s difficulty by varying the amount of time available to prepare orders. Teams are awarded a score for each order they correctly deliver. The score for a specific order is 10 times the number of ingredients in the order. The theoretical maximum score possible for each order list is a function of the number and types of orders in the list.

During each session, teams played eight 5-minute games in 4 kitchen layouts (2 games per kitchen). Every pair of consecutive sessions shared the same set of 4 kitchen layouts. For example, games in sessions 1 and 2 were played in kitchen layouts 1 through 4, sessions 3 and 4 used layouts 5 through 8, and so on. So, each team played 4 games per kitchen. 16 kitchen layouts were used for the study; each layout presented a combination of various task constraints. Constraints included lack of space (counter space/floor space), narrow corridors, isolated players, and partitioned kitchens.

All 16 kitchen layouts were combined with 12 unique order lists to generate 64 unique games. The number of orders in any order list was kept high enough to ensure none of the teams would be able to complete all orders in the list.

Participants

The participants were 24 university students (9 female and 14 male and one participant chose not to answer). Participant age ranged from 19-27 years (mean=20.6, SD=1.84). All participants were between the ages of 19 and 22, except one 24- and one 27-year-old. Only one of the 6 teams was a homogeneous all-male team, the rest had both male and female members.

A campus-wide announcement was made for the study. 24 participants were selected from a pool of 40 students who expressed interest in the study. The selection criterion was based on the feasibility of all 4 participants being able to

come into the lab (together) at least 3 times a week. Groups of 4 people with similar schedules were selected, it was done to minimize the number of canceled sessions due to the unavailability of one or more individuals. All experimental procedures were reviewed and approved by University IRB.

Procedure

Participants were first brought in for an introductory session, during which: (1) The study requirements and the participants' responsibilities were explained. (2) Subject IDs and team numbers were assigned, which remained constant for the entire duration of the study. (3) Three 1-hour timeslots were allotted to each group based on the availability of all 4 members. Two of the three timeslots were selected for the group's usual weekly schedule, that is, when they would come to the lab each week for the study. The third timeslot was used as a fallback option for rescheduling sessions, if necessary.

The study required participants to come to the lab for 11 one-hour sessions. The 11 sessions were executed in the following order: (1) In the first session, participants completed a Cognitive Task Battery (CTB) of 7 tasks; (2) the next 4 sessions (sessions 2-5), participants played the game; (3) during the sixth session, participants completed the Advanced Raven's Matrices test; (4) this was followed by 4 more game sessions (sessions 7-10); (5) in the final session, the CTB from session 1 was repeated.

Each game session involved participants playing eight 5-minute games (40 minutes total). Each player played the game inside their assigned (by the experimenter) acoustic pods. After playing the first 4 games, players were asked to step out of their pods and take a short break before returning to their pods to play the last 4 games of the session. Players were encouraged to discuss game strategy during the session breaks and at the end of each session. The experimenter on duty manually logged these discussions.

Data

The data analyzed in this study was obtained from six university student teams, each playing 64 games across the 8 game sessions. Data from one game was lost due to technical problems (Game 2 for Team 2). So, we performed this analysis using data from 383 games. Action, game state, and gaze information were recorded at 60Hz by the system. Game state information included the position of every object and player at every frame, score, active orders, time remaining per active order, and time remaining for the game. Action data included all button-press information for players and the resulting action within the game environment.

Analysis and Results

We began our analysis by plotting the average performance across all games played in each kitchen layout to test for the effects of various kitchen constraints on task performance. Figure 3 represents the average performance across all games in each kitchen layout. Six teams participated in the study,

each playing 4 games per kitchen. This meant, we had data from 24 games for each kitchen layout, with the exception of the 'BaseLevel' kitchen, which had 23 data points because data from one game was lost due to technical issues.

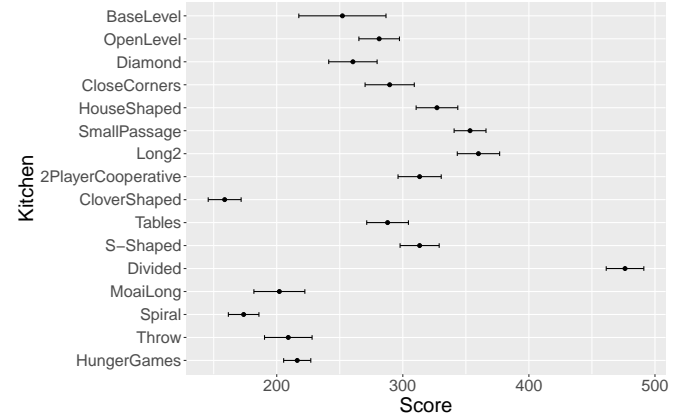


Figure 3: The graph presents the mean and standard error of scores across all games played in each kitchen. The kitchen layouts are arranged in chronological order (the sequence in which teams played games in the kitchens) from top to bottom.

Figure 3 presents several interesting trends for changes in performance across kitchens, which include increasing performance for games played in the first 7 kitchens and the relatively low performance in the final 4. However, in the current study we focus on the 'Divided' kitchen layout because of the consistent and considerably high scores associated with the games played in this kitchen. This was confirmed using a Tukey's HSD test which showed that the game scores for the Divided kitchen differed significantly ($p < .05$) from all other kitchens.

The high scores in the Divided kitchen are particularly intriguing because it is the only kitchen design where each of the four players was placed in separate sections of the kitchen and forced to work in isolation (Figure 4). Additionally, Teams played several games in 11 other kitchens before playing in the Divided kitchen. All 11 kitchen layouts were designed to allow (and, in some cases, force) players to collaborate with each other. Yet, none of the teams were able to adopt a cooperative strategy which was more efficient than working in isolation.

Interestingly, the Divided kitchen was not entirely devoid of coordination among team members. For example, to ensure multiple players did not end up preparing the same order, each individual had to keep track of the orders others were working on. Given the fast-paced nature and the complex structure of the game, in addition to the frequent overlap of ingredients in many orders, it was challenging to keep track of everything. However, based on experimenter observation, apart from a small number of instances, players were able to prepare orders without redundancy. Indeed, to reduce uncer-

tainty, some teams used gaze cursors to indicate the orders on which they were working.



Figure 4: The ‘Divided’ kitchen layout. In this layout, every player is isolated to their own small kitchen with all necessary resources.

Teams also used designated player roles to reduce uncertainty during coordination on multiple occasions (teams discussed these strategies during session breaks and at the end of sessions). Pre-assigning player roles reduced each players’ action variability, improving the team’s predictability for player behavior, which ultimately aided coordination. Responsibilities for different player roles included chopping, cooking, and fetching (moving items around the kitchen for various purposes) items. Washing roles were almost never assigned because it is a relatively rare event and was always handled on the fly. So, washing actions were excluded from the analysis.

A correlation analysis of the different actions indicated that players who performed more cooking actions were also more likely to fetch items (0.31), while, chopping actions were negatively correlated with both cooking (-0.12) and fetching (-0.13). All correlations were statistically significant ($p < 0.05$). The correlations between different actions indicate players’ tendency to organize their behavior around certain actions (roles) in the game.

To study the effects of player roles on team performance, we use ‘Role Stability’ (RS) to measure a player’s tendency to adhere to specific roles in a game. We must first define ‘Action Vectors’ (AV) before we define role stability. An action vector is simply a 3-dimensional vector assigned to each player representing their contributions to different actions in a specific game. The 3 components of the vector represent the percentage of cooking, chopping, and fetching actions performed by each player. For example, the value of the cooking component of a particular player for a specific game is obtained using the following formula:

$$\frac{N_{cooking}^P}{N_{cooking}^{Total}} * 100$$

Where $N_{cooking}^P$ is the number of cooking actions executed by the player P in a game, and $N_{cooking}^{Total}$ is the total number of

cooking actions executed by all players in that game.

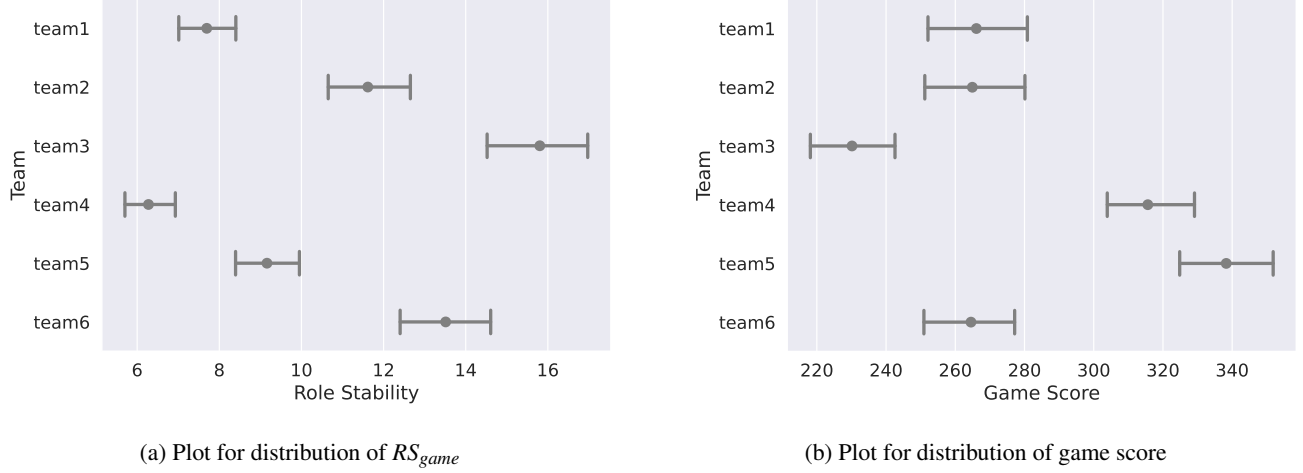
RS of a player in a specific game is simply the standard deviation of an AV. The value of RS is low when the values of the components for the corresponding AV are relatively similar (players engaging in all actions uniformly), while a high RS indicates one or two components have relatively higher values (players engaging more in specific activities). So, higher RS values indicate greater adherence to player roles, and lower RS values suggest more adaptation in players (switching roles as necessary).

Washing actions were excluded from AVs and, consequently, the measure of RS because the number of washing events in a game was negligible compared to other actions, and did not contribute sufficiently to the goals of the current analysis.

To study the relationship between RS and team performance, we first obtained a single measure of RS for each game that reflects the overall strategy used in the game by the team that played it; we refer to this as RS_{game} . RS_{game} is calculated by averaging the RS values of all 4 players in each game. A high value indicates a greater affinity among players to stick to existing roles, while a low value indicates a general adaptive behavior in the team. Figure 5 presents the mean and standard errors (grouped by teams) for RS_{game} values (5a) and game score (5b) for all 383 games.

Comparison of the two plots in Figure 5 reveals that the two highest scoring teams (Teams 4 and 5 in Figure 5b) had some of the lowest RS_{game} values (Figure 5a), while, the range of RS_{game} values for the worst performing team (Team 3) was the highest. Teams 2 and 6 demonstrated mediocre performance, and their RS_{game} values also hovered somewhere in the mid-ranges. Finally, the performance of Team 1 matched those of Teams 2 and 6 but their RS_{game} values were very low. The overall trend indicates an inverse relationship between RS_{game} and team performance (with a correlation of -0.28).

To account for the hierarchical nature of the data, mixed effects regression models were used to further test for the effect of role stability on team performance. However, in addition to RS_{game} , we define a second composite variable for role stability to use in the model. RS_{game} conveys information about the strategy used by a team in a specific game but fails to capture team behavior across games in a task environment (kitchen layout). The new variable was used in the model to add information about the spread of RS values across games played by a team in a kitchen. The values for the new variable were obtained by calculating the standard deviation of RS_{game} values for all 4 games played by each team in each kitchen layout. A high value of the standard deviation indicates greater variations in team strategy for games played in a specific kitchen, while a low value suggests use of similar strategies across games in a specific kitchen layout. Since the new variable provided a measure of a team’s tendency to stick to similar strategies in a specific kitchen layout, we call this variable strategy consistency (SC).


 Figure 5: Mean and standard error plots for RS_{game} (left) and game score (right) across all games played by each team.

Random Effect			
Variable		Variance	Std. Dev.
Team		1976	44.5
Kitchen Layout		7288	84.9
Fixed Effects			
Variable	Coeff.	Std. Err.	t value
Intercept	123.1	54.72	2.25
RS_{game}	4.24	1.19	3.55
SC	-4.21	3.74	-1.26
Baseline	0.34	0.12	2.79

 Table 1: Results of model fit for a mixed effects model predicting team performance based on the value of RS_{game} , SC, and baseline performance for each team. The model also has random intercepts for each team and each kitchen.

The game score of each game was used as the dependent variable to fit mixed effects models. Random intercepts were used for Team IDs and kitchen layouts. RS_{game} and SC values were used as the fixed effect predictor variables. In addition, a baseline score for each team was also added as a fixed effect. The baseline score for a team was set to the score of the first game of the second session. Games from session 1 were not used for this purpose because teams spent their first sessions familiarizing themselves with the game mechanics. So, by using a game from the first session we open ourselves up to the possibility of selecting a game that might not be an accurate measure of the team’s baseline performance.

Three models were fit to the data to determine the usefulness of RS_{game} and SC in predicting game score. The first model was a simple random effects model with random intercepts for each team along with the baseline performance of teams as a fixed effect (null model). For the second model, RS_{game} was added as a fixed effect to the null model. For

the third and final model, SC was added as fixed effect to the second model. Model fits were assessed using the Akaike Information Criterion (AIC).

The model with no fixed effects was the worst of the three models (AIC: 4510). The model with only RS_{game} as a fixed effect was an improvement over the first model (AIC: 4498). Finally, the model with both fixed effects was the best model (AIC: 4493). Changes in AIC values between the 3 models were statistically significant, which indicate that both measures added predictive power to the model. The results of the final model fit are shown in Table 1. The positive coefficient for RS_{game} suggests that teams scored higher in games where they were more persistent about sticking to their roles compared to games in which they showed more adaptive behavior, while the negative coefficient for SC indicates that teams that were more likely to stick to a strategy across games for a particular kitchen design, performed better in general.

Discussion

We used a cooperative cooking game (The CAT) to study human coordination in a complex dynamic task. Six 4-player teams of university students each played the game across eight 1-hour sessions. Team communication was limited to a shared-gaze paradigm implemented in the system. However, teams were allowed to discuss gameplay outcomes and strategies during session breaks and at the end of each session.

Our data suggests that all 6 teams reached peak performance when players were isolated in their own sub-kitchens and forced to work alone (in the ‘Divided’ kitchen layout). In this layout, each player had to perform all actions necessary to prepare an order, including chopping, cooking, and fetching items. In other kitchen designs, where the kitchen is shared between players, teams would share responsibilities among players. However, effectively dividing responsibilities during gameplay was challenging in the absence of verbal communication channels. So, players often stuck to pre-assigned roles

(decided by the team) to reduce prediction uncertainty and improve coordination within the team.

We use ‘Role Stability’ to measure players’ tendency to stick to specific roles in a game. Due to the hierarchical structure of the data, we fit mixed effects models (with random intercepts for teams and kitchens) to determine the relationship between role stability and game performance. The results of the analysis suggest a positive relationship between role stability and performance, that is, reduction in behavioral variability through adherence to assigned player roles was beneficial to performance in general.

The results of the hierarchical model seemingly contradict the patterns observed in figure 5, which indicates an inverse relationship between role stability and game performance. However, this is not the case, as the apparent differences between the two results may be attributed to the random effects of team behavior and kitchen designs. Indeed, the plots in figure 5 indicate that teams which used more adaptive strategies on average (lower values of mean role stability) had higher game scores overall. Higher performance among adaptive teams in dynamic tasks have been suggested in the past. In one such study, teams that showed higher adaptability to role structure changes performed better when they were faced with unforeseen changes in the task (LePine, 2003). Dynamic allocation of team roles have also been shown to benefit team performance among artificial agents playing video games (Kim, 2006). Future publications may consider a deeper analysis of the inter- and intra-team differences in coordination.

Finally, The results also show that teams that stuck to similar strategies across games in each kitchen design were more likely to score higher. Teams that frequently switched strategies may have been experimenting with different strategies to find one optimal for the team, which could have led to poor performance.

Conclusion

We introduced – The CAT – a game-based experimental paradigm for studying team coordination. Each team played a cooperative cooking game across eight 1-hour sessions. Data was collected from six 4-player teams.

Team coordination strategies in the CAT belong to a spectrum between fully adaptive and fully planned behavior. Participating teams used a combination of strategies which involve assignment of player roles (planned behavior) and dynamic adaptation to changing task demands (adaptive behavior), for coordination. Assignment of player roles reduced behavioral variability thus increasing action predictability among team members, which in turn helped with team coordination. Our results show that strategic reduction of behavioral variability through adherence to player roles was beneficial to team performance. Finally, the results also show that teams performed worse when there was higher variation in their gameplay strategies across games.

References

- Cooke, N. J., Rivera, K., Shope, S. M., & Caukwell, S. (1999, sep). A Synthetic Task Environment for Team Cognition Research. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 43(3), 303–308. doi: 10.1177/154193129904300337
- Elliott, L. R., Coovert, M. D., Schifflett, S. G., Salas, E., Cooke, N. J., & Shope, S. M. (2017). Designing a Synthetic Task Environment. In *Scaled worlds: Development, validation and applications* (1st ed., p. 273–288). Routledge.
- Gray, W. D. (2017, mar). Game-XP: Action Games as Experimental Paradigms for Cognitive Science. *Topics in Cognitive Science*, 9(2), 289–307. doi: 10.1111/tops.12260
- Gray, W. D., & Banerjee, S. (2021, oct). Constructing Expertise: Surmounting Performance Plateaus by Tasks, by Tools, and by Techniques. *Topics in Cognitive Science*, 13(4), 610–665. doi: 10.1111/tops.12575
- Green, C. S., & Bavelier, D. (2003, may). Action video game modifies visual selective attention. *Nature*, 423(6939), 534–537. doi: 10.1038/nature01647
- Kim, I.-C. (2006). Dynamic role assignment for multi-agent cooperation. In A. Levi, E. Savaş, H. Yenigün, S. Balcisoy, & Y. Saygın (Eds.), *Computer and Information Sciences – ISCIS 2006* (pp. 221–229). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Large, A. M., Bediou, B., Cekic, S., Hart, Y., Bavelier, D., & Green, C. S. (2019, December). Cognitive and Behavioral Correlates of Achievement in a Complex Multi-Player Video Game. *Media and Communication*, 7(4), 198–212. doi: 10.17645/mac.v7i4.2314
- LePine, J. A. (2003). Team adaptation and postchange performance: Effects of team composition in terms of members’ cognitive ability and personality. *Journal of Applied Psychology*, 88, 27–39. doi: 10.1037/0021-9010.88.1.27
- Lindstedt, J. K., & Gray, W. D. (2013). Extreme Expertise: Exploring Expert Behavior in Tetris. In M. Knauff, M. Pauen, N. Sebanz, & I. Wachsmuth (Eds.), *Proceedings of the 35th annual meeting of the cognitive science society, cogsci 2013, berlin, germany, july 31 - august 3, 2013*. cognitivesciencesociety.org.
- Sibert, C., Gray, W., & Lindstedt, J. (2020). Interrogating Feature Learning Models to Discover Insights Into the Development of Human Expertise in a Real-Time, Dynamic Decision-Making Task. *Topics in Cognitive Science*, 9, 374–394. doi: 10.1111/tops.12225
- Vesper, C., Schmitz, L., Sebanz, N., & Knoblich, G. (2013). Joint Action Coordination through Strategic Reduction of Variability. *Annual Meeting of the Cognitive Science Society (cogsci)*(October), 1522–1527.

Improving Reinforcement Learning with Biologically Motivated Continuous State Representations

Madeleine Bartlett^{1†} (madeleine.bartlett@uwaterloo.ca),
Kathryn Simone^{1†} (kpsimone@uwaterloo.ca),
Nicole Sandra-Yaffa Dumont^{2†} (ns2dumont@uwaterloo.ca),
P. Michael Furlong² (michael.furlong@uwaterloo.ca),
Chris Eliasmith² (celiasmith@uwaterloo.ca),
Jeff Orchard¹ (jorchard@uwaterloo.ca),
and Terrence C. Stewart³ (terrence.stewart@nrc-cnrc.gc.ca)

¹ Cheriton School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada

² Centre for Theoretical Neuroscience, University of Waterloo, Waterloo, ON N2L 3G1, Canada

³ National Research Council of Canada, University of Waterloo Collaboration Centre, Waterloo, ON N2L 3G1, Canada

Abstract

Learning from experience, often formalized as Reinforcement Learning (RL), is a vital means for agents to develop successful behaviours in natural environments. However, while biological organisms are embedded in continuous spaces and continuous time, many artificial agents use RL algorithms that implicitly assume some form of discretization of the state space, which can lead to inefficient resource use and improper learning. In this paper we show that biologically motivated representations of continuous spaces form a valuable state representation for RL. We use models of grid and place cells in the Medial Entorhinal Cortex and hippocampus, respectively, to represent continuous states in a navigation task and in the CartPole control task. Specifically, we model the hexagonal grid structures found in the brain using Hexagonal Spatial Semantic Pointers, and combine this state representation with single-hidden-layer neural networks to learn action policies in an Actor-Critic framework. We demonstrate our approach provides significantly increased robustness to changes in environment parameters (travel velocity), and learns to stabilize the dynamics of the CartPole system with comparable mean performance to a deep neural network, while decreasing the terminal reward variance by more than 150x across trials. These findings at once point to the utility of leveraging biologically motivated representations for RL problems, and suggest a more general role for hexagonally-structured representations in cognition.

Keywords: Reinforcement Learning; Grid Cells; Spatial Semantic Pointers; Representations

Introduction

Humans and animals are able to learn how to interact with their environment through a process of trial-and-error, repeating behaviours that lead to high rewards, and avoiding behaviours that lead to punishments. This process is known as conditioning and has inspired the development of Reinforcement Learning (RL) algorithms for training computational systems. RL algorithms, in turn, have provided further insights into the nature of learning in biological agents.

Classical RL algorithms discretize state and action spaces, and assume that time can be divided into discrete time steps. However, biological agents exist and evolve in continuous time and space, and therefore their learning mechanisms must

also operate in continuous domains. While discretized representations are convenient when working with standard computers and can often produce good results in RL, they have limitations. For instance, a coarse discretization can result in non-smooth control output and poor performance, while a fine discretization can lead to an explosion in the number of states, memory resources, and time required to learn. Additionally, selecting a discretization that does not match the “correct discretization” of the environment may result in either poor representation or inefficient resource use, or both. Obtaining a good discretization scheme often requires prior knowledge or trial and error. Furthermore, the environment itself may not remain stable during its operational lifespan. This could cause the selected and optimal discretizations to diverge over time, at the expense of either performance or wasted representational resources.

In RL, feature representation plays a crucial role in performance. Various feature encoding methods have been proposed, including discretization of the state space through techniques like tile coding (Sutton, 1996) and using deep auto-encoders to obtain latent state representations (Lange & Riedmiller, 2010). In deep RL networks, a linear output layer is typically used, and so the majority of the neural network can be viewed as a state encoding network followed by linear value function approximation. Additionally, biologically inspired state representations have been explored. For instance, RL agents using grid-cell-like representations have outperformed both deep AC models and agents with place-cell-like representations in 2D navigation tasks (Banino et al., 2018). This supports the idea that grid cells provide a useful basis for RL tasks. However, to our knowledge these benefits have been established only for navigation tasks. The extent to which these benefit generalize to other types of tasks remains an open question.

The use of grid or place cell-like encodings of spatial information in RL networks has been demonstrated to facilitate faster learning on spatial navigation tasks (Gustafson & Daw, 2011; Banino et al., 2018; Dumont & Eliasmith, 2020; Bartlett et al., 2022a,b). The method of modelling grid cells

[†] These authors contributed equally.

used in the present work, first presented by Dumont & Eliasmith (2020), builds on Spatial Semantic Pointers (SSPs; Dumont et al., 2023; Komer & Eliasmith, 2020; Komer et al., 2019), a high-dimensional representation of continuous values used in cognitive models that employ vector symbolic architectures (VSAs). This method allows us to represent continuous state information as a specific type of SSP (hexagonal SSPs, or HexSSPs). The resulting representations can be mapped to individual neurons giving rise to grid cells. HexSSPs have been demonstrated to be flexible, computationally efficient and, being a VSA, highly interpretable (Dumont et al., 2023; Komer & Eliasmith, 2020; Bartlett et al., 2022a,b). In general, SSPs can be generated for any spatial environment regardless of size or shape, and scaled to accommodate changes in the environment (Komer & Eliasmith, 2020). Komer & Eliasmith (2020) further demonstrated that HexSSPs encoding spatial location can support supervised learning of policies to navigate through complex, continuous-space environments containing obstacles. HexSSPs have also proven useful as representations for semantic mapping, Bayesian optimization, and neural representations of probability (Dumont et al., 2023; Furlong et al., 2022; Furlong & Eliasmith, 2022). Due to their ability to represent structured data as vectors (e.g., Voelker et al., 2021), these representations may permit extending simple algorithms to more complex spaces. In particular, SSPs can be used to represent sequences or trajectories through continuous spaces, along with hierarchical representations that mix discrete and continuous data. Consequently, algorithms designed to use SSP input can be applied to tasks with complex feature data.

Recent research has illustrated the usefulness of HexSSPs when learning navigation policies in an online fashion using RL (Bartlett et al., 2022a,b). However, this past work was limited to a task defined in discrete space (Gymnasium’s MiniGrid: Chevalier-Boisvert et al., 2018). In this paper, we present the results of a series of simulations demonstrating the benefit of using HexSSPs to represent continuous state to solve tasks with an Advantage Actor-Critic (A2C) network. The A2C network is first tested on a novel spatial navigation task ‘RatBox’, designed as a continuous-space variant of MiniGrid. Additionally, as the representational capacity of HexSSPs generalizes to representing continuous *feature spaces* (Dumont & Eliasmith, 2020), we also run simulations on a continuous state RL benchmark task, CartPole (Brockman et al., 2016). This problem is analogous to balancing an inverted pendulum, relevant for numerous animal behaviors, such as walking or perching on a branch. CartPole has previously been solved efficiently with continuous representations of the state in an actor-critic model with neural networks (Anderson, 1989) and spiking neural networks (Frémaux et al., 2013). Furthermore, grid cell-like representations of the state have been shown to improve the performance of a Deep Q network on the CartPole problem (Yu et al., 2020).

Methods

Hexagonal SSPs

Spatial Semantic Pointers (SSPs) are a high-dimensional vector representation of lower-dimensional continuous spaces (Plate, 1995; Komer et al., 2019; Komer & Eliasmith, 2020), developed within the framework of the Semantic Pointer Architecture (Eliasmith, 2013). SSPs represent state variables by selecting frequency components in the Fourier domain and using those components to project continuous state variables into the high-dimensional frequency space, followed by an inverse Fourier transform. Specifically, to represent m -dimensional data, $\mathbf{x} \in \mathbb{R}^m$, we generate an encoding matrix, $\Theta \in \mathbb{R}^{d \times m}$, and define the SSP representation of \mathbf{x} as:

$$\phi(\mathbf{x}) = \mathcal{F}^{-1}\{\mathbf{e}^{i\Theta\mathbf{x}}\}, \quad (1)$$

where elements of Θ are sampled uniformly from the interval $[-\pi, \pi]$ and d is the dimensionality of the SSP representation. We further constrain Θ so that $\mathbf{e}^{i\Theta\mathbf{x}}$ has conjugate symmetry, to ensure the inverse Fourier transform does not generate imaginary components. This method for representing continuous values is also known as *fractional binding* (Komer et al., 2019), *fractional convolution powers* (Plate, 1994), or *fractional power encoding* (Frady et al., 2022).

In this work we use Hexagonal SSPs (HexSSPs), a variant of SSPs in which the encoding matrix is specifically structured to model grid cell activity. The encoding matrix is constructed so that the dot product with an encoded point and other points in the domain mimics the activity of grid cell neurons in the medial entorhinal cortex (MEC) of the hippocampus (Dumont & Eliasmith, 2020).

Algorithm 1 Hexagonal SSP Generator. Given data \mathbf{x} with dimensionality m , this returns its SSP encoding $\phi(\mathbf{x})$. The input scales, \mathcal{S} , are scalar values, and rotations, \mathcal{R} , are a set of m -dimensional rotation matrices.

- 1: **procedure** HEX-SSP($\mathbf{x}, m, \mathcal{S}, \mathcal{R}$)
 - 2: $\mathbf{v}_1, \dots, \mathbf{v}_{m+1} \leftarrow$ Coordinates of regular m -dim simplex
 - 3: $\mathbf{V} \leftarrow \begin{pmatrix} | & & | \\ \mathbf{v}_1 & \dots & \mathbf{v}_{m+1} \\ | & & | \end{pmatrix}^T$
 - 4: $\Theta \leftarrow \text{stack}(\{\mathbf{s}\mathbf{R}\mathbf{V} \mid \mathbf{s} \in \mathcal{S}, \mathbf{R} \in \mathcal{R}\})$
 - 5: $\phi(\mathbf{x}) = \mathcal{F}^{-1}\{\mathbf{e}^{i\Theta\mathbf{x}}\}$
 - 6: **return** $\phi(\mathbf{x})$
 - 7: **end procedure**
-

The algorithm for constructing HexSSPs is given in Algorithm 1. HexSSP encoding matrices are constructed from $m+1$ vectors that form a regular m -simplex in m -dimensional space. The simplex is determined by minimizing the expression $\sum_i^m \sum_{j=1, i \neq j}^m \mathbf{v}_i \cdot \mathbf{v}_j$, where $\mathbf{v}_i, \mathbf{v}_j$ are unit vectors that make up the simplex. Stacking these vectors produces an initial $(m+1) \times m$ encoding matrix, \mathbf{V} . We can specify the kernel function, $k(\mathbf{x}, \mathbf{x}')$ that is approximated by the dot product between two SSPs, $\phi(\mathbf{x})$ and $\phi(\mathbf{x}')$. With this initial encoding

matrix, \mathbf{V} , the resulting kernel function will have a hexagonally tiled pattern.

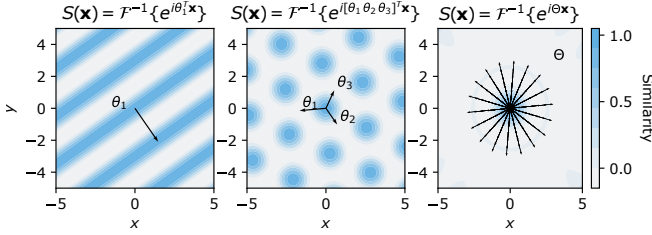


Figure 1: **Construction of HexSSPs from Fourier basis functions.** Left: An encoding matrix consisting of a single Fourier basis function, $e^{i\theta_1^T x}$, results in a kernel function, $k(\mathbf{x}, \mathbf{x}')$, with oscillations in 2D space characteristic of spatial frequency representation. $k(\mathbf{x}, \mathbf{x}')$, in turn, is approximated by the dot product between HexSSPs, each representing 2-dimensional variables. Middle: When $m+1$ such Fourier basis functions are included (3 for the 2-dimensional space depicted here, and spaced 120° apart), the interference pattern results in the hexagonally-patterned kernel function. Right: As more rotations and scales of these vectors are used to generate the encoding matrix, the kernel function becomes smoother, with one centralized peak and more shallow local optima.

The firing patterns of grid neurons in the MEC of the hippocampus are characterized by different orientations and sizes. To mimic this features, the complete encoding matrix, Θ , used to construct HexSSPs is composed of multiple rotations and scalings of \mathbf{V} . This choice also has useful practical implications: as more rotations and scales are added to the representation, the kernel function becomes smoother, with one centralized peak and more shallow local optima, as shown in Figure 1. SSPs created with such encoding matrices are also more robust to noise than randomly generated representations, and so can be more accurately encoded in spiking neural networks via grid cells (Dumont & Eliasmith, 2020).

Advantage Actor-Critic Network

The Advantage Actor-Critic (A2C) network implemented for these simulations was the same as that presented in (Bartlett et al., 2022a). The network architecture is shown in Figure 2. It was implemented in Python and Nengo (Bekolay et al., 2014) using the principles of the Neural Engineering Framework (NEF; Eliasmith & Anderson, 2003). The code is available at <https://github.com/maddybartlett/ImprovedRLContinuousStateReps>.

States were represented either as a one-hot vector or by a population of rectified linear neurons. When solving the Rat-Box task, encoders were sampled from the regions of SSP space associated with the observation space using the Sobel sampling method, thus generating a population of grid cell neurons. The number of neurons in this case was calculated such that the number of neurons was at least 10 times the dimensionality of the HexSSP and a power of 2 (a requirement

of the Sobel sampling method). For the CartPole task, the encoders were randomly sampled from the whole SSP space, as the observation space is unbounded for some state variables. Other than the state representation layer, no other part of the network used neurons. Learning was performed on the connection weights from the state representation layer to the output. Connection weights were initialized to zero.

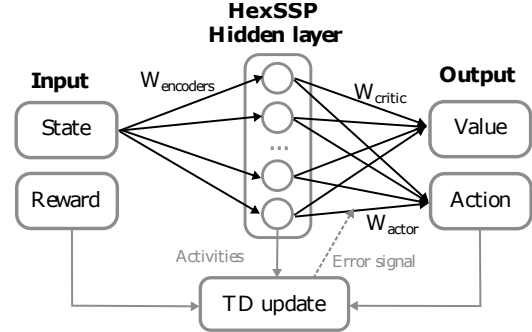


Figure 2: A schematic of the Advantage Actor-Critic (A2C) network. W_{encoders} project a HexSSP representation of the state to neurons in the Hidden Layer.

Hyperparameter optimization

The performance of RL networks and algorithms is sensitive to the selection of hyperparameters (Sutton & Barto, 2018). To identify the hyperparameter configuration that maximizes performance, we first defined the performance metric as the terminal return, in turn computed as the mean reward received in the last 100 episodes of a single trial. We then searched for the configuration of hyperparameters that maximize performance, a process referred to as hyperparameter optimization, using the simulated annealing algorithm implemented in the Neural Network Intelligence (NNI; Microsoft, 2021) Python package. This algorithm begins by randomly sampling from the hyperparameter space, and progresses by sampling from regions that achieved higher performance. Each NNI experiment therefore determines the performance achievable given the stochastic selection of hyperparameters. For both the HexSSP network and all state discretizations, 100 such NNI experiments were conducted. The random seed was fixed across all NNI experiments. Hyperparameter optimization was performed over the parameter ranges specified in Table 1, which were selected based on performance results obtained through systematic exploration of the hyperparameter space (Bartlett et al., 2022a) on a navigation task similar to that used in this present work. In cases where identical, optimal performance could be achieved with multiple sets of hyperparameters, the set of hyperparameters was selected based on the most temporally stable terminal behavior. For discrete representations of the state, the number of bins per state was set for each NNI experiment and the remaining hyperparameters were left free.

Symbol	Variable	Range - RatBox	Range - CartPole
ε	probability of an off-policy action	[0.3, 0.6]	[0.2, 0.6]
α	learning rate	[0.001, 0.5]	[0.001, 0.5]
β	action value discount	[0.8, 1.0]	0.9
γ	state value discount	[0.8, 1.0]	0.99
η	proportion of active cells	\circ [0.01, 0.5]	\circ [0.01, 0.5]
N	number of neurons		\circ {1024, 2048, 4096}
\mathcal{R}	rotations of V	\circ [4, 5, 6, 7, 8, 9, 10]	\circ [4, 5, 6, 7]
S	scalings of V	\circ [4, 5, 6, 7, 8, 9, 10]	\circ 8
l	length scale of representation	\circ [1, 100]	\circ [0.01, 1.0]

Table 1: Hyperparameter values tested during optimization of networks solving each task. Hyperparameters marked by \circ apply only to models using HexSSPs for state representation.

Evaluation on continuous space navigation with obstacles (RatBox)

Discrete state representations are incapable of perfectly capturing the boundaries of irregularly shaped objects. A novel 2D environment, ‘RatBox’, containing 4 obstacles that an agent must navigate around to reach a goal location was developed for these experiments (see Figure 3), to assess the ability of HexSSPs to learn an efficient policy in this scenario. The state of the agent in this environment is its 2D position and heading, $s = (x, y, w)$. The state space within the environment is continuous in that the agent can be in any location within the 600×600 space. Additionally the agent is able to face any direction, $w \in [0, 2\pi]$.

The discrete agent’s action space consists of a set of vectors, $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{n_a}\}$. In this task there are 4 action primitives, each corresponding with a ‘compass’ direction (North, South, East, West). The discrete A2C network learns a policy over this discrete action space. To allow for a continuous action space, the action taken is a weighted sum of the action primitives, *i.e.*, $\mathbf{a}(t) = \sum_i^{n_a} c_i \mathbf{a}_i$. In this case, the output from the actor portion of the network is a 4-vector, c , consisting of the learned value for each action primitive. This weighted sum is the direction vector for the agent moving at a fixed speed. The agent’s maximum speed was set to 10,000 pps (pixels per second), which equates to 100 pixels per timestep.

To use this method of representing a continuous action space, we formulate our policy as an isotropic Gaussian distribution over the action vector, $\pi_W(s) = \mathcal{N}(\mu_W(s), \sigma^2 I)$, with small isotropic noise, $\sigma^2 \ll 1$, and where $\mu_W(s)$ is the weighted sum of the discrete action vectors,

$$\mu_W(s) = \text{softmax}(W\phi(s))^T [\mathbf{a}_1, \dots, \mathbf{a}_{n_a}]^T \quad (2)$$

This parameterization of the mean action vector is the softmax of a linear decoding from the state population, $W\phi(s)$.



Figure 3: The RatBox environment

We assume the isotropic Gaussian noise added to this action is small to obtain an approximately-deterministic stochastic policy. Then we can derive the approximate policy gradient from the expected rate of reward as a function of the policy parameters, $J(W)$:

$$\nabla_W J(W) = \mathbb{E}[\nabla_W \log \pi_W(s) A(s, a)] \quad (3)$$

$$\nabla_W \log \pi_W(s) = (I - \pi_W(s)) \pi_W(s) \phi(s)^T \quad (4)$$

$$W_{\text{new}} = W_{\text{old}} + \alpha \nabla_W J(W), \quad (5)$$

where $A(s, a)$ is the advantage function, α is the learning rate, and I is the $n_a \times n_a$ identity matrix. With this update, we can improve the policy – parameterized by the decoding weight matrix, $W \in \mathbb{R}^{n_a \times d}$ – with the TD(0) actor-critic learning rule.

The network was tested under two different conditions. In the baseline condition, the agent’s state was represented using a tabular representation. We generated several resolutions by applying 6, 8, 10 or 12 partitions to each of the three state dimensions. In the second condition, the state was represented using HexSSPs and a population of grid cell neurons. In the 10- and 12-bin discrete conditions, none of the hyperparameter combinations tested by NNI were able to solve the task. We therefore instead used the hyperparameters found for the 6-bin condition. The optimal network for each condition was then trained 10 times using 10 different random seeds. Interestingly, in the 10-bin condition, the final network was able to learn the task on some of the random seeds.

The average reward was calculated over a 100-trial window, and then averaged across the 10 random seeds. The results, shown in Figure 4, illustrate that the HexSSP and 6-bin solutions were able to learn the task. Performance declines as the resolution becomes finer in the discrete condition.

Learning in non-stationary environments can be challenging for RL algorithms that use a tabular representation of the state, as changes in the environment can potentially cause incompatibilities between the optimal and actual discretizations. In general, continuous state representations avoid this limitation by allowing for generalization between states. This holds for the HexSSP approach we use here, where the extent of generalization over the state space is specified by the length scale parameter. Assuming an appropriate length scale, we would therefore expect our algorithm to exhibit robustness to changes in the environment, as compared to the state discretization approaches. To test this prediction, we performed hyperparameter optimization on the network with the agent’s speed set to 10,000 pixels per second (pix/sec), and then assessed performance with the agent’s speed set to either 10,000 pix/sec or reduced to 5,000 pix/sec. No other changes were made and the networks were tested again with 10 random seeds. The results are shown in Figure 4. While the performance of the network using HexSSPs to represent the state drops slightly following the change in speed, closer inspection revealed that this was due to two of the ten random seeds resulting in no learning, while the rest of the seeds led to performance as good as the original (data not shown). In contrast, none of the baseline networks were able to solve the

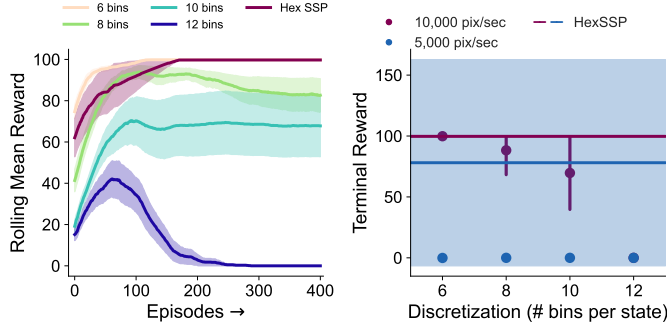


Figure 4: **Performance on RatBox.** Left: Learning curves using HexSSPs or tabular approaches for state representations, averaged across 10 random seeds. The shaded area depicts the standard error of the mean. Right: Average terminal reward (with 95% confidence interval) across the different representations when the agent’s maximum speed was 10,000 pix/sec vs. 5,000 pix/sec.

task following the change in agent speed, suggesting that the networks would need a different discretization, or to be re-optimized, in order to adapt. This result shows that continuous HexSSPs are a more general solution that is less sensitive to changes in the task or environment compared to standard discrete representations.

Evaluation on the inverted pendulum (CartPole) problem

Here we characterize learning on the standard CartPole task from OpenAI’s Gym Library (Brockman et al., 2016). The CartPole’s dynamics are unstable, making performance particularly sensitive to state representation accuracy. Errors introduced due to state discretization should therefore cause a decrease in performance. Indeed, prior work investigating learning with continuous and discrete control schemes report fewer trials to learn with a continuous algorithm compared to a discrete algorithm on the similar CartPole Swingup task (Doya, 2000). We therefore first sought to validate that the proposed continuous representation confers a learning advantage over the state discretization approach. We characterized learning across 5 discretizations corresponding to a distinct number of partitions applied to each of the 4 state variables. Hyperparameter optimization was performed for each discretization, and performance characterized across 10 runs of the model with different seeds.

As can be seen in Figure 5, terminal reward grows slowly as the resolution of the discretization increases, but is well below the terminal reward achieved by the HexSSP model. Crucially, and in contrast to performance with most of the tabular approaches, terminal performance using HexSSPs to represent state exhibits relatively small variation across seeds (95% confidence interval: [496.88,499.97]). This is especially surprising as the only source of randomness using tabular approaches is the initial conditions given by the environment;

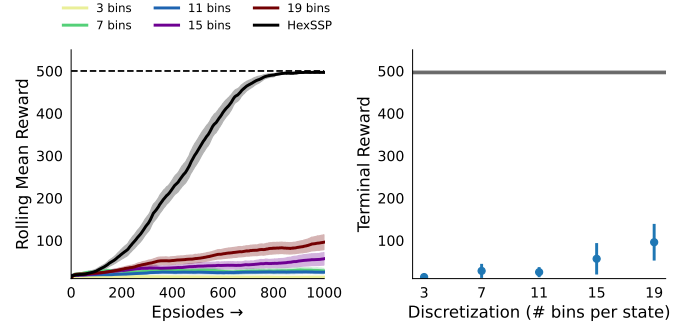


Figure 5: **Performance on CartPole.** Left: Learning curves using HexSSPs or tabular approaches for state representation. Shown is the mean across 10 runs of the model. The shaded area depicts the standard error of the mean. The dotted line indicates the maximum possible episodic reward achievable. Right: Terminal rewards observed in 10 runs of each model (and 95% confidence intervals) for each discretization condition. The gray bar denotes the 95% confidence interval for performance using HexSSPs.

HexSSPs are subject to this source of randomness in addition to that incurred by the sampling of encoders for each model run. Of course, hand-tuned discretizations may facilitate superior or more consistent performance, but this requires trial and error or *a priori* knowledge of the problem. HexSSPs reliably produce high terminal performance regardless of initial environment or network conditions in the tested scenario.

Performance comparison against a deep network baseline

So far we have described two advantages of representing state information as a continuous variable with HexSSPs, as measured against state discretization approaches. On the navigation task with RatBox, HexSSP representations conferred greater robustness to changes in model parameters. On the CartPole control task, we observed better terminal performance and lower sensitivity to initial conditions. However, neural networks can also represent continuous state information, and multi-layer networks, in particular, can learn effective representations for decoding value and policy functions. This can include representations similar to that which we have used here in their hidden layers. What specific advantage, if any, is offered by using HexSSP representations on RL problems over the state-of-the-art?

To address this question, we compare the performance of our algorithm to an A2C method that uses a multi-layer perceptron policy (Raffin et al., 2021) on the CartPole task. Figure 6 shows learning curves for both the proposed HexSSP single-layer network model and the deep network baseline model. On the CartPole task, the two methods exhibit comparable performance, as shown by the overlap in the medians and interdecile range produced by a range of initial seeds. Interestingly, the HexSSP method produces more re-

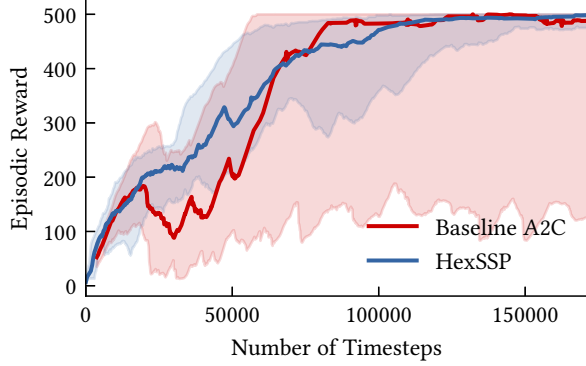


Figure 6: Performance of a Single layer A2C network with HexSSPs and a deep A2C network on CartPole. Comparing the moving average over episodic rewards between our proposed method using HexSSP representations and a baseline implementation of A2C with a multilayer perceptron policy from (Raffin et al., 2021). Solid lines are the medians and shaded areas are the interdecile range, taken over 20 randomly chosen seeds.

liable terminal performance, indicating less sensitivity to the initial seed as shown by lower variance in the terminal reward (Baseline: $\sigma^2 = 20966.52$, HexSSP: $\sigma^2 = 127.84$, $L = 6.19$, $P = 0.017$, Levene’s test of equal variances using group medians). The high reliability of performance with HexSSP representations as compared to that observed with a state-of-the-art approach suggests a fundamental robustness across different types of continuous feature spaces. We speculate that since the baseline model must learn the state representation, it is therefore able to modify this representation late in training when it may no longer be advantageous to do so. The representation for the HexSSP model is fixed, so additional training is not able to degrade its performance in this way.

Discussion

In this paper, we presented HexSSPs as a method for representing continuous states when solving tasks using RL. We evaluated the HexSSPs on a spatial navigation task and compared performance to networks using tabular representations, and to a multi-layer perceptron where the state representation is learned via backpropagation. While a discrete representation could solve the RatBox task as well as the HexSSP method, we found that the discrete method was not robust to changes in the task; the HexSSPs proved to be very robust. We also found that the HexSSP representation was able to achieve a final performance greater than any of the tabular representations on the standard benchmark CartPole task. The HexSSP solution’s performance was also comparable to that of the Deep A2C network, but it is noteworthy that the HexSSP network produced a more reliable terminal performance suggesting that it is more robust to changes in network initialisation. Notably, the performance on CartPole was achieved while adopting a relatively inefficient method of sampling encoders from the subspace spanned by the problem

domain. An important direction for future work is therefore to explore the impact of more efficient sampling methods on this task.

Experiments comparing the HexSSP method and tabular approaches were designed to assess differences in average performance. Assuming the performance metric follows a Gaussian distribution, the sample size of 10 opted for in this work confers to us a 99.8 % probability that the mean performance falls within the span of the sampled data points (computed as $P = (1 - \frac{1}{2^{N-1}}) \times 100\%$, where N is the number of sample points). However, the observation that 2 (20%) of the random seeds yield particularly poor performance on the RatBox task suggest that more sampling would be needed if one were to move beyond characterizing average performance and towards understanding the full distribution of model behavior.

A key characteristic of the HexSSP method is that it allows one to build neuron populations whose firing patterns mimic those of grid and place cells found in MEC and hippocampus across many animal species. The performance improvement on the RatBox task gained by leveraging these hexagonally-patterned representations is expected considering their role in spatial navigation in biological agents. These firing patterns are readily comparable to neural recordings from animal models used in neuroscience, such as rodents. However, to our knowledge, these representations have not been implicated in motor control analogous to that required to solve the CartPole task. In this case, comparing the performance of our network against the behavior of a biological agent may yield insight into the specific role of hexagonally-patterned representations in biological cognition.

On the RL problems explored in this work, we observed the benefits of HexSSPs being more robust to noise and better able to generalize to changes in the environment. These features are essential for autonomous agents needing to learn online, potentially in dynamic or unstable environments. Thus an interesting avenue for future work would be to explore the extent to which the HexSSP method proves useful in solving non-stationary problems. Additionally, the robustness to noise is significant for cognitive models, and this representation supports encoding in spiking neural networks. Moreover, these spiking implementations can be implemented straightforwardly on neuromorphic hardware.

Online Resources

All code necessary to reproduce these results are hosted at: <https://github.com/maddybartlett/ImprovedRLContinuousStateReps>

Acknowledgments

This project was supported in part by collaborative research funding from the National Research Council of Canada’s Artificial Intelligence for Logistics program (AI4L-116), as well as by CFI (52479-10006) and OIT (35768) infrastructure funding, the Canada Research Chairs program, and NSERC Discovery grant 261453.

References

- Anderson, C. W. (1989). Learning to control an inverted pendulum using neural networks. *IEEE Control Systems Magazine*, 9(3), 31–37.
- Banino, A., Barry, C., Uribe, B., Blundell, C., Lillicrap, T., Mirowski, P., ... others (2018). Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705), 429–433.
- Bartlett, M. E., Stewart, T. C., & Orchard, J. (2022a). Biologically-based neural representations enable fast on-line shallow reinforcement learning. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 44).
- Bartlett, M. E., Stewart, T. C., & Orchard, J. (2022b). Fast online reinforcement learning with biologically-based state representations. In *Proceedings of the 20th international conference on cognitive modeling*.
- Bekolay, T., Bergstra, J., Hunsberger, E., DeWolf, T., Stewart, T. C., Rasmussen, D., ... Eliasmith, C. (2014). Nengo: a Python tool for building large-scale functional brain models. *Frontiers in neuroinformatics*, 7, 48.
- Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). OpenAI gym. *arXiv preprint arXiv:1606.01540*.
- Chevalier-Boisvert, M., Willems, L., & Pal, S. (2018). *Minimalistic gridworld environment for gymnasium*. Retrieved from <https://github.com/Farama-Foundation/Minigrid>
- Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural computation*, 12(1), 219–245.
- Dumont, N. S.-Y., & Eliasmith, C. (2020). Accurate representation for spatial cognition using grid cells. In *Cogsci*.
- Dumont, N. S.-Y., Stöckel, A., Furlong, P. M., Bartlett, M. E., Eliasmith, C., & Stewart, T. C. (2023). Biologically-based computation: How neural details and dynamics are suited for implementing a variety of algorithms. *Brain Sciences*, 13(2), 245.
- Eliasmith, C. (2013). *How to build a brain: A neural architecture for biological cognition*. Oxford University Press.
- Eliasmith, C., & Anderson, C. H. (2003). *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press.
- Fraday, E. P., Kleyko, D., Kymn, C. J., Olshausen, B. A., & Sommer, F. T. (2022). Computing on functions using randomized vector representations (in brief). In *Neuro-inspired computational elements conference* (p. 115-122).
- Frémaux, N., Sprekeler, H., & Gerstner, W. (2013). Reinforcement learning using a continuous time actor-critic framework with spiking neurons. *PLoS computational biology*, 9(4), e1003024.
- Furlong, P. M., & Eliasmith, C. (2022). Fractional binding in vector symbolic architectures as quasi-probability statements. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 44).
- Furlong, P. M., Stewart, T. C., & Eliasmith, C. (2022). Fractional binding in vector symbolic representations for efficient mutual information exploration. In *Proc. icra workshop, towards curious robots, mod. approaches intrinsically-motivated intell. behav.* (pp. 1–5).
- Gustafson, N. J., & Daw, N. D. (2011). Grid cells, place cells, and geodesic generalization for spatial reinforcement learning. *PLoS Computational Biology*, 7(10), e1002235.
- Komer, B., & Eliasmith, C. (2020). Efficient navigation using a scalable, biologically inspired spatial representation. In *Cogsci*.
- Komer, B., Stewart, T. C., Voelker, A. R., & Eliasmith, C. (2019). A neural representation of continuous space using fractional binding. In *41st annual meeting of the cognitive science society*. Montreal, QC: Cognitive Science Society.
- Lange, S., & Riedmiller, M. (2010). Deep auto-encoder neural networks in reinforcement learning. In *The 2010 international joint conference on neural networks (ijcnn)* (pp. 1–8).
- Microsoft. (2021, 1). *Neural Network Intelligence*. Retrieved from <https://github.com/microsoft/nni>
- Plate, T. A. (1994). *Distributed representations and nested compositional structure*. Citeseer.
- Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6, 623–641.
- Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268), 1–8.
- Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. In D. Touretzky, M. Mozer, & M. Hasselmo (Eds.), *Advances in neural information processing systems* (Vol. 8). MIT Press.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- Voelker, A. R., Blouw, P., Choo, X., Dumont, N. S.-Y., Stewart, T. C., & Eliasmith, C. (2021). Simulating and predicting dynamical systems with spatial semantic pointers. *Neural Computation*, 33(8), 2033–2067.
- Yu, C., Behrens, T. E., & Burgess, N. (2020). Prediction and generalisation over directed actions by grid cells. *arXiv preprint arXiv:2006.03355*.

Quantifying Performance in Magnitude Comparison Tasks Using a Drift-Diffusion Model

Mark Bensilum (abens02@student.bbk.ac.uk)

Department of Psychological Sciences, Birkbeck, University of London
Malet Street, London WC1E 7HX

Richard P. Cooper (R.Cooper@bbk.ac.uk)

Department of Psychological Sciences, Birkbeck, University of London
Malet Street, London WC1E 7HX

Abstract

We investigate the viability of the drift-diffusion framework to account for behaviour on magnitude comparison tasks. Data from both published studies on magnitude comparison and a simulation are analysed to estimate the key drift-diffusion model parameters, using the EZ-diffusion method and HDDM package. All methods resulted in linear mappings between drift rate and difficulty (indexed using 1 - smaller/larger), with an intercept that was consistently close to zero for non-symbolic tasks. The EZ method was rapid and simple to apply, but subject to bias when using aggregate data or when individual accuracy was very high. In contrast, the HDDM tool produced results that were less biased, but individual differences were under-estimated. We conclude that application of parameter estimation methods, particularly in research on individual differences, requires careful consideration of their limitations.

Keywords: magnitude comparison; numerical cognition; distance effect; drift-diffusion model; parameter estimation; individual differences

Introduction

In tasks where two quantities are compared, a distance effect is usually observed, in which performance improves as the difference between the magnitudes increases (e.g. Buckley & Gillman, 1974). In contrast, when asked to judge whether three symbolic numbers are in order or not, a reverse-distance effect is often found, such that performance is better when the distance is small (as in the case of consecutive numbers).

Most studies of distance effects have used either accuracy or response time (RT) to investigate the effects, analysing these performance metrics individually, or sometimes combined into a single measure. An alternative approach is to use a drift-diffusion model (DDM) to quantify the decision process. A DDM assumes that evidence for a decision is extracted from the stimulus and accumulates until a given threshold is reached, at which point a decision is deemed to have been made.

Key advantages of the DDM approach include its ability to disentangle the time required for a decision from that involved in other processes (such as encoding the stimulus or performing the motor response), its prediction of the distribution of response times (rather than just an average) and the fact that it can account for both the speed and accuracy of responses (including a possible trade-off between the two). However, while some studies have used a DDM to investigate magnitude comparison tasks (e.g., Park & Starns, 2015; Ratcliff & McKoon, 2018; Krajcsi, Lengyel, & Kojouharova,

2018), few of these have focused on individual differences in the distance effect across a range of ratios using different tasks and estimation methods.

The analyses presented here do just this. They are part of a wider study exploring the reliability of distance and reverse-distance effects under different conditions. In this article, analyses are presented of data from a range of secondary sources, as well as some simple simulations, with a view to developing a baseline model that can account for the key behavioural findings from comparison tasks and, importantly, allow individual differences to be modelled.

Estimating DDM parameters

The three most fundamental DDM parameters are the mean drift rate, boundary separation and non-decision time. The drift rate represents the rate at which evidence is accumulated, and is related to the quality of information contained in the stimulus and the participant's sensitivity to it. The boundary separation represents the evidence threshold at which one or other decision is made, such that a wider separation is associated with a more cautious, slower decision. The non-decision time corresponds to processes that are not directly involved in the decision itself (primarily stimulus encoding and motor response). In the simplest version of the DDM, the initial level of evidence is assumed to be central between the two decision boundaries, so that there is no pre-existing bias for either response at the start of a trial, and the predicted mean RT for correct responses will be the same as that for incorrect ones. More complex models can include variation in the initial evidence level, as well as trial-to-trial variation in the three fundamental parameters. These will not be considered here.

Several methods for estimating the key DDM parameters from a set of experimental data have been described, including fitting RT distributions using a chi-square or Kolmogorov-Smirnov test, the EZ-diffusion functions (Wagenmakers, Van Der Maas, & Grasman, 2007), and the software packages HDDM (Wiecki, Sofer, & Frank, 2013) and fast-dm (Voss & Voss, 2007). These vary greatly in their complexity and run-times, and each method has its own pros and cons (see Alexandrowicz & Gula, 2020 and Ratcliff & Childers, 2015 for some examples of direct comparisons). The present analyses focus on two methods, which were selected for the following reasons. Both methods are freely available, provide

“off the shelf” functionality and have been used in numerous studies across a wide range of tasks. The EZ-diffusion method was chosen as it offers a very simple and quick closed-form method of estimating the three key parameters outlined above. Furthermore, it is possible (in principle) to apply the EZ functions even when raw trial-level data are not available. The EZ functions require three inputs: the proportion of trials with a correct response, and the mean and variance of the RT. The HDDM tool (an open-source Python package) was selected because it allows a hierarchical analysis to be performed, which may increase the statistical power of the results when the number of trials in each condition is small. HDDM uses trial-level data on the accuracy and RT of responses and a Bayesian approach to estimate the joint posterior distribution of the parameters in question, using a Markov chain Monte Carlo method. In the hierarchical version of this model, the parameter estimates for each individual are constrained by the distribution of estimates for the group as a whole.

Study 1: Secondary data analysis

Sources of secondary data

In order to explore how magnitude effects might be explained within the DDM, data from three previous studies on magnitude comparison were selected.

1. Summary-level data from Agrillo, Piffer, and Adriano (2013) were taken from the published article. Three variants of non-symbolic comparison were used: dot arrays, line lengths and audible tone duration, and results were presented in a way that allowed the EZ functions to be applied for each condition. In each task, quantities were presented sequentially and participants were asked to judge which had a greater magnitude. The ratio of the two quantities (larger/smaller) ranged from 1.05 to 4.00 in each variant.
2. Trial-level data from Krajcsi et al. (2018), who carried out non-symbolic (dot array) and symbolic (single-digit number) comparison tasks, were used. In both tasks, the authors presented two quantities simultaneously. There were 27 values of ratio in each task, ranging from 1.125 to 9.0. For the non-symbolic task, the number of dots in each array was five times the value of the digits used in the corresponding symbolic task, to avoid arrays with fewer than five dots. Each stimulus pair was presented 10 times in each of two counterbalanced arrangements (i.e., 10 times with the larger on the left and 10 with the larger on the right), giving a total of 720 trials per participant. The data are available on the repository at osf.io.
3. A subset of results from Halberda, Ly, Wilmer, Naiman, and Germine (2012) was used. The original study involved over 10,000 participants in an online comparison task, in which a set of intermixed yellow and blue dots was presented on each trial. The number of dots of each

colour ranged between five and 20, with four different ratios between 1.14 and 2.00, each presented around 70 times. Trial-level experimental data for the first 501 participants (from the repository at osf.io) were used in the present analyses.

Parameter estimates for the experimental data

Estimates for the three main DDM parameters were calculated for each of the tasks in the secondary datasets outlined above. For the Agrillo et al. (2013) study, where only aggregate data were available, the EZ diffusion method was used, while both the EZ and the HDDM method were applied to the trial-level data from Krajcsi et al. (2018) and Halberda et al. (2012). Some of the resulting drift rate estimates are summarised in Table 1. Figure 1A shows examples of the estimated drift rates for each level of $1 - \text{ratio}_L$ (see below), calculated using the EZ method for the entire group and for each individual, as well as the HDDM hierarchical and non-hierarchical methods.

EZ-Diffusion estimates For each comparison task and ratio level in Agrillo et al. (2013), aggregate data on accuracy and RT were taken from tables 1 and 2 of the article. Two different metrics for indexing difficulty level were tested. The first used ratio defined using larger/smaller – this is the metric that has been used most commonly in non-symbolic comparison tasks (e.g., Halberda & Feigenson, 2008) and will be referred to here as ratio_H . The second metric (referred to here as ratio_L) utilised the reciprocal ratio, which was subtracted from one so that smaller values corresponded to more difficult comparisons (in keeping with the more traditional measure), giving $1 - \text{smaller/larger}$. This is equivalent to the distance/larger measure suggested by Krajcsi et al. (2018). The resulting estimates for drift rate were tested for deviation from linearity using a Wald-Wolfowitz test on the residuals, and a test based on sums of squares.

In each task, it was found that the ratio_L difficulty metric resulted in a more linear relation between difficulty and drift rate than ratio_H . Importantly, the intercept in each case was close to zero, which is consistent with one of the central tenets of the DDM, that the drift rate should approach zero as the information that can be extracted from the stimulus reduces (i.e., in this context, as the two quantities become indiscriminable). The highest drift rate estimate for the dots task was 0.259, with a slope of 0.330 (using $1 - \text{ratio}_L$). For comparison, in the line-length task, the estimated drift slope was 0.241 while that in the tone-duration task was 0.366 (the mean boundary separations were 0.118 and 0.110 respectively).

A similar pattern in drift rate was reported for the dot-array task by Krajcsi et al. (2018) (see below), but it is noteworthy that all of the non-symbolic tasks in Agrillo et al. (2013) followed this trend and, arguably, the linearity of the relation suggests that it captures an important aspect of the underlying decision processes. The mean boundary separation for the dots task was 0.129. However, the non-decision times showed signs of bias, such that the estimates for the most dif-

ficult trials appear to be unusually long (approaching 900ms in some tasks) while those in the easiest trials appear rather short (around 200ms). Although several studies have found that the the DDM parameters do not show pure "selective influence" (such that experimental changes that would be expected to affect only one parameter are often found to influence several, e.g., Rafiei & Rahnev, 2021), the range of non-decision times seen here appears to be excessive.

When applied to the aggregated group data for the non-symbolic dot-array task in Krajcsi et al. (2018), the EZ-diffusion functions yielded patterns that were qualitatively very similar to those seen in the Agrillo et al. (2013) data – a good linear fit when difficulty is indexed by $1 - \text{ratio}_L$, with an intercept close to zero. The highest drift rate estimate was 0.345, and the drift slope was 0.336. However, the mean boundary separation was 0.212, which is high compared to values that are typically reported. Furthermore, once again, the non-decision time estimates were problematic, with a mean of around 212ms and 42% being negative. In both datasets, the low and negative estimates for non-decision time are probably due to inflated values for RT variance, caused by the differences between individuals being incorporated into the aggregated data for the group, as well as contamination by outliers or lapses of concentration (see below).

EZ estimates based on group-level data for the dots task of Halberda et al. (2012) similarly revealed a linear relation between estimated drift rate and $1 - \text{ratio}_L$, with a slope of 0.316 and intercept of zero. However, the estimated non-decision time for the easiest ratio was just 67ms.

When trial-level data are available, the problem with inflated RT variance in the EZ method can be avoided by calculating the performance in each condition for every individual separately. In this way, the proportion correct, RT mean and RT variance values each apply to a particular stimulus and participant. With this approach, the estimated drift rates for the non-symbolic task in Krajcsi et al. (2018) are higher, but still show a good linear fit (using $1 - \text{ratio}_L$) with intercepts close to zero for all individuals. The drift slope was calculated for each individual, giving a mean of 0.365 and standard deviation of 0.0905. The mean boundary separation was 0.144, which is within the typical range. Although the mean non-decision times were mostly positive, nearly all participants showed at least one negative estimate (overall mean 266ms).

There are at least two potential issues remaining in the EZ estimates. The first relates to outliers and 'contaminant' RTs. There is a range of opinions on how to deal with outliers (e.g., Berger & Kiefer, 2021). In contrast, contaminant RTs may not even be identifiable. Ratcliff and Tuerlinckx (2002) described contaminant trials as those in which the RT is lengthened due to, for example, a momentary lapse of focus or the re-starting of the decision process on a given trial. Importantly, some of these extended response times may fall within the typical range, so that they do not stand out as outliers but may not be fitted well by a DDM. One imperfect but common and very simple strategy for dealing with out-

liers is to remove trials with an RT greater than a given value (although this will probably leave many contaminant trials in place). Excluding all trials with an RT of more than three seconds from the Krajcsi et al. (2018) data resulted in a general increase in drift rates such the the mean slope increased to 0.451. The mean boundary separation increased slightly to 0.153 while the mean non-decision time increased to 322ms, although some negative estimates remained.

The second issue relates to the edge correction that must be applied to the accuracy values in the EZ method. The functions are undefined in cases where there are no errors, so an adjusted value for proportion correct must be applied. Wagenmakers et al. (2007) suggested a replacement value of $1 - \frac{1}{2n}$ should be used, where n is the number of trials in the cell. For example, if there are 10 trials in the condition in question, the replacement value for proportion correct would be 0.95. This approach was adopted by Krajcsi et al. (2018). When data for the group as a whole were used (as above in the case of the Agrillo et al., 2013 data), few if any cells will require this correction. In contrast, when data are analysed at the individual level, a larger number of cells will reach 100% accuracy. This will be particularly true of trials in which the non-symbolic quantities are easy to discriminate (and is likely to apply to some extent at all ratio levels in a symbolic comparison task). In the non-symbolic task, over 60% of cells required an edge correction. Using a higher replacement value (e.g., 0.98) resulted in higher estimates for drift rate and boundary, and better fits to the data for the easiest trials.

When data from Halberda et al. (2012) were analysed at the individual level using the EZ functions, the mean slope of the drift rate estimates was 0.517 (higher than that for the group-level analysis). The mean non-decision times ranged from 106ms to 129ms, which still appear somewhat low. The very large number of trials in this task may have increased the number of contaminants, which would tend to bias the EZ estimates.

HDDM Estimates In using the HDDM package, there were three key aims. The first was to explore further the range of individual differences in drift estimates, while avoiding the biases in the EZ estimates described above. The second was to compare a hierarchical model with a non-hierarchical one. The third was to compare directly the variability of individual performance in the symbolic and non-symbolic variants, by analysing estimates from the symbolic data from Krajcsi et al. (2018).

For each task in the Krajcsi et al. (2018) dataset, both a non-hierarchical analysis (using each participant's data independently) and a hierarchical one were performed. In the hierarchical model, the overall performance of the group constrains the estimates for each individual. As noted above, the use of a hierarchical method can increase the statistical power of the Bayesian analysis, which may allow better detection of subtle effects. When there is sufficient evidence, the estimates for any given individual may still be allowed to vary substan-

	Mean drift slope			Mean drift SD		
	EZ indiv	Non-hierarchical	Hierarchical	EZ indiv	Non-hierarchical	Hierarchical
Krajcsi dots	0.365	0.504	0.575	0.0905	0.1	0.0192
Halberda dots	-	0.517	-	-	0.182	-
Krajcsi digits	0.158	0.242	0.205	0.0829	0.107	0.0198

Table 1: Mean drift slope estimates using various methods

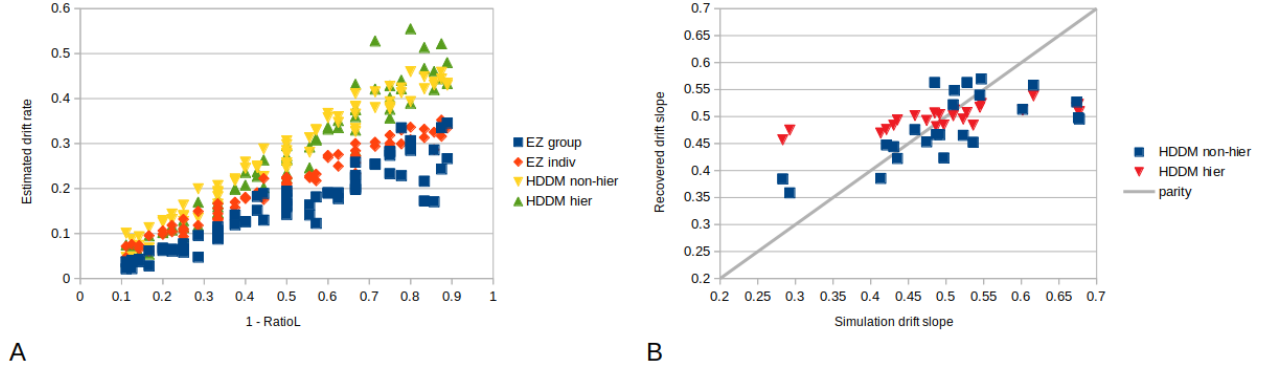


Figure 1: (A) Drift rate estimates for Krajcsi et al (2018) non-symbolic task using four different methods. (B) Recovered drift rates vs simulation drift rates for simulated non-symbolic task

tially from the group, as described in Wiecki et al. (2013). For the Halberda et al. (2012) dataset, only the non-hierarchical method was used. A hierarchical model was not run with this dataset due to its large size.

For the non-symbolic task of Krajcsi et al. (2018), when each individual was analysed separately, a linear relation between drift rate and ratio_L , with a zero intercept, was observed for all individuals. The slope of this relation was calculated for each participant, giving a mean of 0.504 (higher than the estimates from the group analysis above) and standard deviation of 0.100. The mean boundary separation was 0.173 and mean non-decision time 309ms.

When the non-symbolic task data analyses were repeated using a hierarchical model, as described above, linear trends were again observed for all individuals, with a slightly higher mean slope (0.575) but much reduced range (standard deviation 0.0192). The mean boundary separation was broadly similar (0.180) while the mean non-decision time was a little quicker (291ms).

When the symbolic comparison data were analysed, a linear drift slope was once again observed, but with a non-zero intercept (as reported by Krajcsi et al., 2018, using the EZ method). This finding is also consistent with the DDM approach, in that digit pairs with the closest ratios can still be discriminated, so the drift rate in this case would not be expected to reach zero. Using the non-hierarchical method, the mean drift slope was 0.242 (SD = 0.107), and the mean in-

tercept was 0.252 (SD = 0.0638). When using a hierarchical model, the mean slope decreased to 0.205 (SD = 0.0198), while the mean intercept was 0.208 (SD = 0.0121). As above, the hierarchical model tended to constrain the range of individual differences in drift rate. To determine whether a general processing speed for each individual may have influenced the drift rates in both tasks, the correlation between the drift rates in the two variants was computed. Because the drift slopes in the symbolic task are relatively low (due to the non-zero intercept), the maximum drift value was considered to be a more appropriate measure of an individual's processing speed. The correlation between maximum drift rate in the non-symbolic and symbolic tasks was significant ($r^2 = 0.238, p = 0.018$), indicating that the individuals' maximum sampling rates in the two variants were related.

Finally, data from Halberda et al. (2012) were analysed using the non-hierarchical approach with HDDM. The results from the individual-level analyses showed that the mean drift slope across all 501 participants analysed was 0.517 (SD = 0.182). The distribution of slopes was approximately normal (with a slight right-skew of 0.693). The mean intercept was -0.0260, with 95% of individuals having an intercept that was in the range ± 0.080 . The mean boundary separation and non-decision time were 0.161 and 246ms respectively.

For the non-symbolic tasks, given that each method of estimation yielded a linear drift mapping with an intercept that was close to zero for every individual and for each group as a

whole, it appears that the key source of individual differences is the slope of this relationship. In the case of the symbolic task, the relationship still appears to be linear but both the slope and intercept appear to vary from person to person. In both cases, the hierarchical HDDM estimates are constrained by the group mean, reducing the range of drift estimates between individuals.

In the non-symbolic task, there were signs that some individuals had shown a speed-accuracy trade-off (as evidenced by a reduction in boundary separation as the comparisons become easier, for example), although this did not seem to be a major factor for the group overall.

Study 2: Parameter recovery

In order to explore the differences between the individual and hierarchical estimates, a set of simulated data for the non-symbolic case was generated using the *rtlimits* package (Singmann, Brown, Gretton, & Heathcote, 2022) in R (R Core Team, 2022). This package uses a DDM to generate a set of simulated responses, with the accuracy and RT being predicted using the given parameters.

Each simulation was designed to mirror the experimental design of the non-symbolic task in Krajcsi et al. (2018), with 24 participants, and 27 different ratios. To simplify and speed up the analysis process, each simulated pair of quantities was repeated 10 times rather than 20 in the counterbalanced experimental design, and the analyses were grouped by $1 - \text{ratio}_L$. For each simulated individual, a linear function was used to generate a drift rate at each difficulty level (indexed by $1 - \text{ratio}_L$). The gradients for this mapping were drawn from a normal distribution with mean 0.504 and standard deviation of 0.1 (based on the individual estimates from the non-symbolic experimental data of Krajcsi et al., 2018). In order to focus purely on the drift rates, the boundary separation and non-decision time were fixed at 0.150 and 300ms respectively, and there was no trial-to-trial variation. The resulting data were analysed using the default priors in HDDM using two approaches as above: independent (non-hierarchical) analysis of each individual, and a hierarchical model.

Once again, each approach gave a linear relation between drift and $1 - \text{ratio}_L$ for every individual, with intercepts close to zero. The mean drift slope for the non-hierarchical estimates was lower (0.481) than that for the hierarchical method (0.499), where the true mean was 0.504. However, individual differences in drift rates were captured less well by the hierarchical method ($SD = 0.0196$) than the non-hierarchical method (0.0610), where the true value was 0.1. The mean boundary separation estimates were close to the simulation values (0.163 for the non-hierarchical and 0.153 for the hierarchical analysis, true value of 0.150). Similarly, the mean of the recovered non-decision times were very close to the true values (297ms and 299ms respectively, compared with 300ms). Thus, in common with the experimental data, both analyses allowed recovery of the boundary separation and non-decision time, but the hierarchical analysis produced a

narrower range of drift slopes across the simulated individuals, such that those whose ‘true’ slope was small tended to be overestimated and vice versa. This replicates the findings on “shrinkage” reported by Ratcliff and Childers (2015). The relation between the true (simulation) and recovered drift rates is shown in figure 1B.

General discussion

A series of datasets (experimental and simulated) relating to magnitude comparison have been analysed using a variety of readily-available and popular tools for estimating DDM parameters. In all cases, a linear relation was found between difficulty (indexed using $1 - \text{ratio}_L$) and estimated drift rate

In addition, for the non-symbolic tasks (involving dot arrays, line lengths and tone duration), the intercept was consistently close to zero. The robustness and linearity of these relationships, across different tasks and large numbers of individuals, suggests that there may be a common underlying factor at play. Note that the intention is not to suggest that the underlying representations are necessarily linked (c.f., Walsh, 2003), or that the non-symbolic representations somehow underpin other numerical representations (as argued by Dehaene, 2011, for example). In particular, it could be argued that the concept of “difficulty” is only applicable to the non-symbolic task (in which even the most skilled participant would not be able to distinguish between two quantities that have a ratio approaching unity). In contrast, the symbolic pair “8 9” is, arguably, not more difficult to discriminate than “1 9” but the decision process does take longer. Hence, it is conjectured that the DDM results may indicate a commonality in the decision process, based on the dynamic sampling of evidence, the rate of which is a linear function of $1 - \text{ratio}_L$. The correlation between the participants’ maximum drift rates in the two variants hints that the general sampling rate for an individual may drive some of the similarities in distance effects reported in different tasks. Furthermore, the decision process may be related to the pattern of weights in a putative connectionist network, which depends on the ratio of the quantities being compared but may also be influenced by their relative frequencies in the case of symbolic comparisons (see Verguts, Fias, & Stevens, 2005, for example). Further work is ongoing to explore this possibility.

Conclusions

The overall aim in the present study was to derive a simple baseline model for magnitude comparison, which may be used to explore the patterns found in various tasks, including the reliability across different individuals. As shown above, as far as non-symbolic comparison is concerned, the intercept of the linear relationship between difficulty ($1 - \text{ratio}_L$) and drift rate is consistently very close to zero (in line with the theory underpinning the DDM). This leaves the slope of the relationship as the main source of variation between individuals. For the symbolic task, the maximum drift rate may be a better index of individual variability.

The work reported here has both theoretical and methodological implications. From the theoretical perspective, when difficulty of magnitude comparison tasks is measured as $1 - \text{ratio}_L$, application of DDM parameter estimation results in a linear relationship between task difficulty and drift rate (and a zero intercept for non-symbolic tasks). This may suggest that difference in magnitudes and the magnitude of the larger item are the key factors affecting evidence accumulation.

A second conclusion from this work is methodological. Parameter recovery shows that the hierarchical technique applied to the data here was able to capture the mean parameters well but failed to recover the variability in those values between individuals. This may be because smaller datasets do not contain enough exemplars of each comparison to overcome the central tendency imposed by the hierarchical technique. Larger datasets are more likely to provide sufficient evidence for the individual estimates to deviate substantially from the mean. However, larger numbers of trials are more likely to lead to fatigue and other factors that could result in contaminants, which would tend to increase bias when using some estimation methodologies. Hierarchical recovery must balance these two opposing pressures. Although other estimation methods exist, and there are settings within the HDDM package that may be adjusted by the expert user, the issues raised here demonstrate that, when the research focus is differences between individuals, careful consideration must be given to both the experimental design and the method of parameter estimation.

References

- Agrillo, C., Piffer, L., & Adriano, A. (2013). Individual differences in non-symbolic numerical abilities predict mathematical achievements but contradict ATOM. *Behavioral and Brain Functions*, 9(1), 26. doi: 10.1186/1744-9081-9-26
- Alexandrowicz, R. W., & Gula, B. (2020). Comparing Eight Parameter Estimation Methods for the Ratcliff Diffusion Model Using Free Software. *Frontiers in Psychology*, 11.
- Berger, A., & Kiefer, M. (2021). Comparison of Different Response Time Outlier Exclusion Methods: A Simulation Study. *Frontiers in Psychology*, 12.
- Buckley, P. B., & Gillman, C. B. (1974). Comparisons of digits and dot patterns. *Journal of Experimental Psychology*, 103(6), 1131–1136. (Place: US Publisher: American Psychological Association) doi: 10.1037/h0037361
- Dehaene, S. (2011). *The number sense: How the mind creates mathematics*, Rev. and updated ed. New York, NY, US: Oxford University Press. (Pages: xxii, 316)
- Halberda, J., & Feigenson, L. (2008). Developmental change in the acuity of the "number sense": The approximate number system in 3-, 4-, 5-, and 6-year-olds and adults. *Developmental Psychology*, 44(5), 1457–1465. doi: 10.1037/a0012682
- Halberda, J., Ly, R., Wilmer, J. B., Naiman, D. Q., & Germine, L. (2012). Number sense across the lifespan as revealed by a massive Internet-based sample. *Proceedings of the National Academy of Sciences*, 109(28), 11116–11120. doi: 10.1073/pnas.1200196109
- Krajcsi, A., Lengyel, G., & Kojouharova, P. (2018). Symbolic Number Comparison Is Not Processed by the Analog Number System: Different Symbolic and Non-symbolic Numerical Distance and Size Effects. *Frontiers in Psychology*, 9, 124. doi: 10.3389/fpsyg.2018.00124
- Park, J., & Starns, J. J. (2015). The Approximate Number System Acuity Redefined: A Diffusion Model Approach. *Frontiers in Psychology*, 6. doi: 10.3389/fpsyg.2015.01955
- R Core Team. (2022). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from <https://www.R-project.org/>
- Rafiei, F., & Rahnev, D. (2021). Qualitative speed-accuracy tradeoff effects that cannot be explained by the diffusion model under the selective influence assumption. *Scientific reports*, 11(1), 45. doi: 10.1038/s41598-020-79765-2
- Ratcliff, R., & Childers, R. (2015). Individual Differences and Fitting Methods for the Two-Choice Diffusion Model of Decision Making. *Decision (Washington, D.C.)*, 2015, 10.1037/dec0000030.
- Ratcliff, R., & McKoon, G. (2018). Modeling numerosity representation with an integrated diffusion model. *Psychological Review*, 125(2), 183–217. doi: 10.1037/rev0000085
- Ratcliff, R., & Tuerlinckx, F. (2002). Estimating parameters of the diffusion model: Approaches to dealing with contaminant reaction times and parameter variability. *Psychonomic Bulletin & Review*, 9(3), 438–481. doi: 10.3758/BF03196302
- Singmann, H., Brown, S., Gretton, M., & Heathcote, A. (2022). rtdists: Response time distributions [Computer software manual]. Retrieved from <https://CRAN.R-project.org/package=rtdists> (R package version 0.11-5)
- Verguts, T., Fias, W., & Stevens, M. (2005). A model of exact small-number representation. *Psychonomic Bulletin & Review*, 12(1), 66–80. doi: 10.3758/BF03196349
- Voss, A., & Voss, J. (2007). Fast-dm: A free program for efficient diffusion model analysis. *Behavior Research Methods*, 39(4), 767–775. doi: 10.3758/BF03192967
- Wagenmakers, E.-J., Van Der Maas, H. L. J., & Grasman, R. P. P. P. (2007). An EZ-diffusion model for response time and accuracy. *Psychonomic Bulletin & Review*, 14(1), 3–22. doi: 10.3758/BF03194023
- Walsh, V. (2003). A theory of magnitude: common cortical metrics of time, space and quantity. *Trends in Cognitive Sciences*, 7(11), 483–488. doi: 10.1016/j.tics.2003.09.002
- Wiecki, T., Sofer, I., & Frank, M. (2013). HDDM: Hierarchical Bayesian estimation of the Drift-Diffusion Model in Python. *Frontiers in Neuroinformatics*, 7.

Uncovering Iconic Patterns of Syllogistic Reasoning: A Clustering Analysis

Daniel Brand (daniel.brand@psychologie.tu-chemnitz.de)

Predictive Analytics, Chemnitz University of Technology, Germany

Nicolas Riesterer (riestern@cs.uni-freiburg.de)

Cognitive Computation Lab, University of Freiburg, Germany

Marco Ragni (marco.ragni@hsw.tu-chemnitz.de)

Predictive Analytics, Chemnitz University of Technology, Germany

Abstract

Syllogistic reasoning is one of the core domains of human reasoning research. Over its century of being actively researched, various theories have been proposed attempting to disentangle and explain the various strategies human reasoners are relying on. In this article we propose a data-driven approach to behaviorally cluster reasoners into archetypal groups based on non-negative matrix factorization. The identified clusters are interpreted in the context of state-of-the-art theories in the field and analyzed based on the posited key assumptions, e.g., the dual-processing account. We show interesting contradictions that add to a growing body of evidence suggesting shortcomings of the current state of the art in syllogistic reasoning research and discuss possibilities of overcoming them.

Keywords: syllogistic reasoning; cognitive modeling; clustering; non-negative matrix factorization; dual-process theory

Introduction

The ability to reason about information is an essential skill for humans in almost all aspects of their lives. Consequently, the research of human reasoning has been a key field of study to advance our understanding about human cognition for an extensive time span. One of the core domains within the field is syllogistic reasoning, which is being investigated for over a century now (Störring, 1908). In its most common form, a syllogism consist of two quantified statements (premises) with first-order logic quantifiers (*All*, *Some*, *No*, and *Some ... not*), that interrelate three terms (commonly abbreviated by A, B, and C) via a middle-term as shown in the following example:

All A are B.

Some B are C.

What, if anything, follows?

The task is to conclude what the relation between the two end-terms occurring in only one of the premises (A and C) is. Additionally, there is the possibility that no valid conclusion (NVC) is possible resulting in a total of nine distinct response options.

For convenience, syllogisms are often abbreviated. Quantifiers are represented by the letters (*All*: A, *No*: E, *Some*: I and *Some ... not*: O). The arrangement of the terms is called *figure*. Throughout this paper, we will use the figure notation from Khemlani and Johnson-Laird (2012), which is shown in the table below:

Figure 1	Figure 2	Figure 3	Figure 4
A-B	B-A	A-B	B-A
B-C	C-B	C-B	B-C

Put together, the syllogism in the example above would be abbreviated as *AII*. Conclusions can be represented in a similar way, combining the quantifier and the direction (*ac* or *ca*). For example, *Some C are A* would be abbreviated by *Ica*.

Given the long history of research, it is not surprising that a large variety of competing theories and models exist. However, the field was unable to reach consensus: In a recent meta-analysis, twelve theories of syllogistic reasoning were compiled and evaluated, concluding that “none of the existing theories is correct. Investigators of reasoning need to develop a better theory of monadic reasoning.” (Khemlani & Johnson-Laird, 2012, p. 23).

Since the inferential mechanisms and strategies are substantially influenced by individual factors (e.g., working memory Gilhooly, Logie, Wetherick, & Wynn, 1993) and are susceptible to the influence of external factors (e.g., content and personal beliefs; Morgan & Morton, 1944), it is not surprising that the observed reasoning behavior shows significant inter-individual differences (e.g., Dames, Klauer, & Ragni, 2022) that current models struggle to capture (Riesterer, Brand, & Ragni, 2020a). As human reasoning behavior seems to be highly individual, the idea that no single inferential account may be able to capture every individual suggests itself (Khemlani & Johnson-Laird, 2012). Consequently, it seems to be more sensible to try to disentangle the different reasoning strategies.

From a more abstract data-driven perspective, the observed behavior of a human reasoner can be represented as a task-response-pattern. A model accounting for the behavior then specifies a process that generates the respective pattern. Thereby, it is restricted by its assumptions and the corresponding parameter space (for a model-evaluation based on this principle, see Riesterer et al., 2020a). From this perspective, the question of disentangling different strategies can be reformulated as a question of uncovering a set of latent (iconic) patterns that are suitable for capturing the patterns of most individuals to a satisfying degree. In this work, we utilize clustering methods to uncover the latent response patterns of individual reasoners and present a way to determine the number of central reasoning strategies.

The rest of the article is structured as follows: First, the background relevant to this work will be introduced. Second, our dataset and the clustering approach used to extract the iconic patterns are described. Third, the obtained patterns are interpreted with respect to their meaning for the state of the art in syllogistic reasoning. Finally, the results are discussed and a general outlook is given.

Background

A common approach aiming at describing the behavioral differences observed in reasoning is the dual-processing account (Evans, 2008), which proposes two mechanisms: a fast-and-frugal heuristic approach (*System 1*; S1) and a deliberative, more logical mechanism (*System 2*; S2). In the field of syllogistic reasoning, models often fall clearly into one of the two categories, with heuristics (e.g., PHM; Chater & Oaksford, 1999) belonging to S1 while approaches closer to logic (e.g., PSYCOP; Rips, 1994) would generally be considered to rely on S2.

Probably the most prominent theory incorporating the idea of dual-processing is the Mental Model Theory (MMT; Johnson-Laird, 1975) and its implementation mReasoner (Khemlani & Johnson-Laird, 2013). At its core, MMT assumes that syllogistic inference is a three-step procedure (Bara, Bucciarelli, & Johnson-Laird, 1995). In the first step, the premises are interpreted to construct a mental model representation of the information. This model is then extended to also incorporate the information of the second premise. In the second step, the constructed model is used to derive a conclusion candidate. This candidate is then put to the test in the third step, which consists of a search for counterexamples, i.e., models that contradict the conclusion but are still consistent with the premise information. If no counterexample is found, the candidate will be responded as the conclusion. Otherwise, a new conclusion candidate is generated, which is then subjected to the search for counterexamples again, or it is concluded that “no valid conclusion” is possible if no new candidates can be created.

The expensive search for counterexamples in MMT is assumed to be a S2 process, while conclusions directly inferred from the initial mental model reflect the more intuition-based strategy associated with S1.

It is important to note that while the average correctness of a participant’s responses typically increase with a higher number of NVC responses (Dames et al., 2022), seemingly corroborating the notion of S2 being responsible for NVC responses, invalid syllogisms are over-represented in the syllogistic domain with more than half of the syllogisms being invalid despite NVC being only one out of nine possible responses. Furthermore, recent work found that the response times did not increase for NVC responses as it would be assumed when engaging in an exhaustive search for counterexamples (Brand, Riesterer, & Ragni, 2022), sowing doubt if the proposed distinction into S1 and S2 truly reflects the processes underlying syllogistic reasoning.

Method

Dataset

The foundation of our analysis is a publicly available dataset by Dames et al. (2022), which contains the response data of 106 participants to all 64 syllogistic tasks. In the original analysis, participants were asked to complete all 64 tasks twice to investigate potential retest effects. However, these effects are out of scope for the present work and the respective data from the syllogistic retest is therefore excluded. Additionally, a variety of individual information about the participants is provided, out of which the *Cognitive Reflection Test* (CRT; Frederick, 2005) including additional questions by Toplak, West, and Stanovich (2014) and the participants’ Need for Cognition (NFC; see Cacioppo & Petty, 1982) are relevant for this work.

Clustering

Clustering refers to an unsupervised learning process of grouping objects together that are similar with respect to some similarity measure (for an overview, see Aggarwal, 2015). The clustering methods used in this work are thereby partitioning approaches that group objects into disjoint sets by minimizing a cost function (e.g., euclidean distances between objects and cluster centroids in k-Means clustering). For our analysis, we compare the performance of k-Means, k-Medoids and a clustering method based on Non-Negative Matrix Factorization (for a similar method, see J. Kim & Park, 2008). As k-Means and k-Medoids are standard procedures, they will only briefly be discussed with respect to potential strengths and weaknesses for the specific analysis.

Of the three methods, k-Means is probably of the most prominent approach for cluster analyses. As the name suggests, k-Means divides objects into k clusters that are defined by centroids representing the mean of the respective objects in the cluster. Thereby, it behaves similar to an aggregation of the data that is commonly performed to investigate response distributions, with the difference that k distributions are obtained instead of a single one, thereby having the potential to provide a better fit for individuals. However, aggregation of data has been criticized to be problematic when investigating individual processes (Riesterer, Brand, & Ragni, 2020c), as different strategies might be entangled by the aggregation process.

In contrast to k-Means, k-Medoids uses actual datapoints as the centroids of the clusters. Hence, no aggregation is performed, eliminating the problems associated with it. However, as the number of participants in reasoning experiments is very limited compared to typical datasets used in machine learning, the approach might not find an optimal centroid for each cluster. Since human data is inherently prone to noise, a pattern found by k-Medoids might contain artefacts that were introduced by confounders unrelated to reasoning processes.

Clustering using NMF Non-Negative Matrix Factorization has the goal of finding a decomposition for an input-matrix X . To this end, a basis matrix $W = m \times k$ and a co-

efficient matrix $H = n \times k$ for a given k need to be found such that:

$$X \approx WH^T \quad (1)$$

These matrices can be obtained by using a variety of solvers, including the commonly used non-negative least squares solver (H. Kim & Park, 2008).

Formally, clustering can also be understood as a problem of matrix decomposition (e.g., J. Kim & Park, 2008). The columns in the W -matrix then represent the centroids of a cluster, while the H -matrix contains the assignment of a data point to the respective cluster.

To use NMF clustering on the syllogistic data, it needs to be represented as a matrix X of shape $m \times n$, where n corresponds to the number of participants and each column corresponds to an m -dimensional vector representing the respective participant's response pattern. To transform the data accordingly, we first represented the responses of each participant as a 64×9 matrix (for the 9 possible response options), meaning that each task is represented as a one-hot-encoded vector. The matrices were subsequently flattened into a 576-dimensional vector, out of which the final data matrix X containing all participant vectors was created (leading to $X = 576 \times 106$). In order to find the matrices W and H , we used the non-negative least squares solver included in the Python package SciPy¹ which is based on the algorithm proposed by Lawson and Hanson (1995).

In order to realize clustering via NMF, additional constraints on the coefficient matrix H are necessary. As the coefficient matrix contains the assignment of the participants to the respective patterns, it needs to be ensured that each participant is only assigned to a single pattern, i.e., that each row represents a one-hot-encoded vector. We realized the constraint by adjusting the H -matrix accordingly after each iteration of the solving algorithm instead of incorporating it into the minimization function, which has the advantage of guaranteeing that the constraint is satisfied.

While constraints on the W -matrices are not necessary, they can be used to enforce properties that are tailored to the specific domain. Each column in the W -matrix represents a complete pattern for all 64 syllogisms, which means that chunks of 9 values belong to a single syllogism. Therefore, we normalized each chunk of a column in each iteration of the NMF algorithm to the Euclidean norm in order to obtain results that are more distinct compared to the wider distributions of k-Means. To ensure that the reconstruction remains unaffected, we adjusted the corresponding column of the H -matrix accordingly.

Note that these constraints are not applied to the final results, but after each iteration of the algorithm instead. This ensures that the final result is optimized with respect to the given constraints, which is a major advantage of methods like NMF.

Determining k Given the strong inter-individual differences and noise that become apparent in syllogistic reasoning, it is challenging to determine an optimal (but low) number of clusters since a higher number of clusters would always allow to capture certain individuals better.

To assess this problem, we used a repeated hold-out validation (for different values for k with 1000 iterations each), i.e., we repeatedly divided the data in random subsets (training-set and test-set). Both sets had the same number of participants. We used four metrics to determine the number of clusters and compare the different clustering methods:

The first metric used is the *Inter-Similarity* and assesses the stability of the found patterns with respect to the specific set of participants. If k is too high, patterns might start to represent outliers. In these cases, it is unlikely that the results are stable, as they are likely to jump between different local minima depending on the dataset at hand. Therefore, clustering is performed on both, the training- and the test set. The resulting patterns of both clustering runs are then compared to each other (pattern vs. pattern) using cosine-similarity:

$$\text{sim}(w_1, w_2) = \frac{w_1 \cdot w_2}{|w_1||w_2|} \quad (2)$$

Inter-Similarity corresponds to the mean similarity between the patterns obtained from applying clustering to the training- and test set. Since the order of patterns might differ between both runs, the result is only based on the most optimal ordering of the patterns.

The second metric is the *Intra-Similarity*, which has a similar reasoning behind it: If k is too high, patterns might start to be too similar to each other. Therefore, the cosine similarity is used to compare the patterns obtained from a single run of clustering. *Intra-Similarity* is then defined as the maximum similarity between two patterns obtained from the same clustering run. However, the *Intra-Similarity* is unable to distinguish between the occurrence of multiple distinct patterns that are similar to each other and generally less distinct patterns, that have a high similarity because of a more blurry appearance.

For the above-mentioned reasons, we use our third metric, the *Entropy*, which indicates how distinct the pattern is: the more "blurry" a pattern is, the higher the entropy. Therefore, by definition, k-Medoids has a perfect score, as it uses a real participant pattern which always has a distinct response to each task. The entropy for the response distribution for a specific task is calculated as follows:

$$H = - \sum_i p_i \cdot \log_2 p_i \quad (3)$$

We use the mean entropy of all tasks of a pattern as the resulting entropy of a pattern.

Finally, we used the *Test-Accuracy*, which is defined as the mean accuracy achieved when using the k patterns obtained from clustering on the training-set as predictors for the participants in the test set. Thereby, the best pattern is selected for each participant.

¹<https://scipy.org>

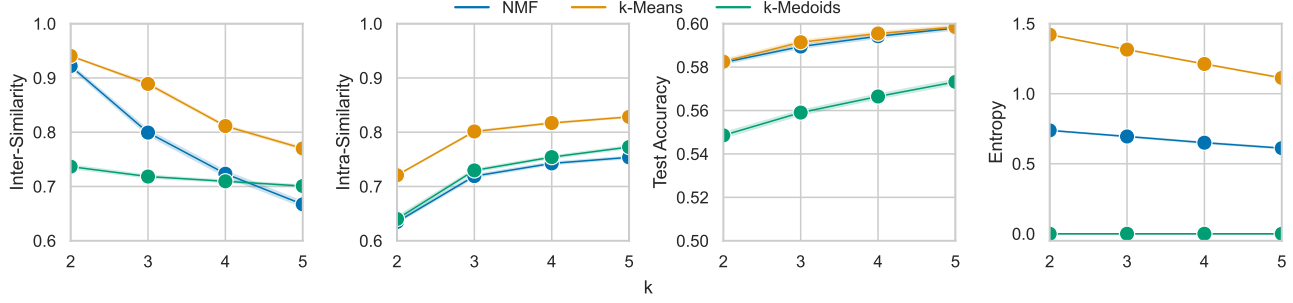


Figure 1: Results of a crossvalidation for kMeans, kMedoids and clustering based on the NMF in terms of inter- and intra-similarity, mean accuracy on the test set and entropy for different numbers of clusters (k).

The results of the metrics for different values of k are shown in Figure 1. For the *Inter-Similarity*, the disadvantage of k-Medoids becomes apparent: The resulting patterns are directly based on the participants, which makes it highly susceptible to changes of the dataset. For NMF and k-Means, a substantial decrease of stability is noticeable with higher levels of k , with k-Means being more robust to the changes overall. However, the downside of k-Means is clearly visible in the *Intra-Similarity*, where its mean-based centroids are substantially less distinct compared to the other methods. For all methods, a substantial change from $k = 2$ to $k = 3$ is apparent, indicating that even a third pattern already leads to a higher similarity between the patterns. However, higher values of k seem to not further increase the similarity to the same extent. This gets confirmed by the *Entropy*, where k-Means also shows to produce less distinct patterns compared to the other methods. This indicates that the worse score of k-Means in *Intra-Similarity* is not due duplicated patterns, but rather an effect of the aggregation. Both, the NMF and k-Means, show an improvement with higher values of k , since the additional clusters allow to build more homogeneous groups. However, as the *Intra-Similarity* indicates, this could also lead to overfitting in the form of almost identical patterns.

For the *Test-Accuracy*, k-Means and NMF show almost the same performance, with k-Medoids falling behind slightly. Also, the differences for varying number of clusters are negligible, suggesting diminishing returns for higher values of k .

Overall, the analysis suggests that a total number of two clusters seems to offer the best trade-off between accuracy and stability. For $k = 2$, the NMF is best suited, since k-Means has a substantially worse *Intra-Similarity* and *Entropy*, while k-Medoids is lacking stability with and therefore also generalizability. Hence, the final patterns (see Figure 2) were obtained with $k = 2$ by using NMF clustering. In the following section, the patterns will be interpreted with respect to their meaning within the domain of syllogistic reasoning.

Interpreting the Patterns

In the following, we will take a closer look on the obtained patterns and the groups of participants that were assigned to

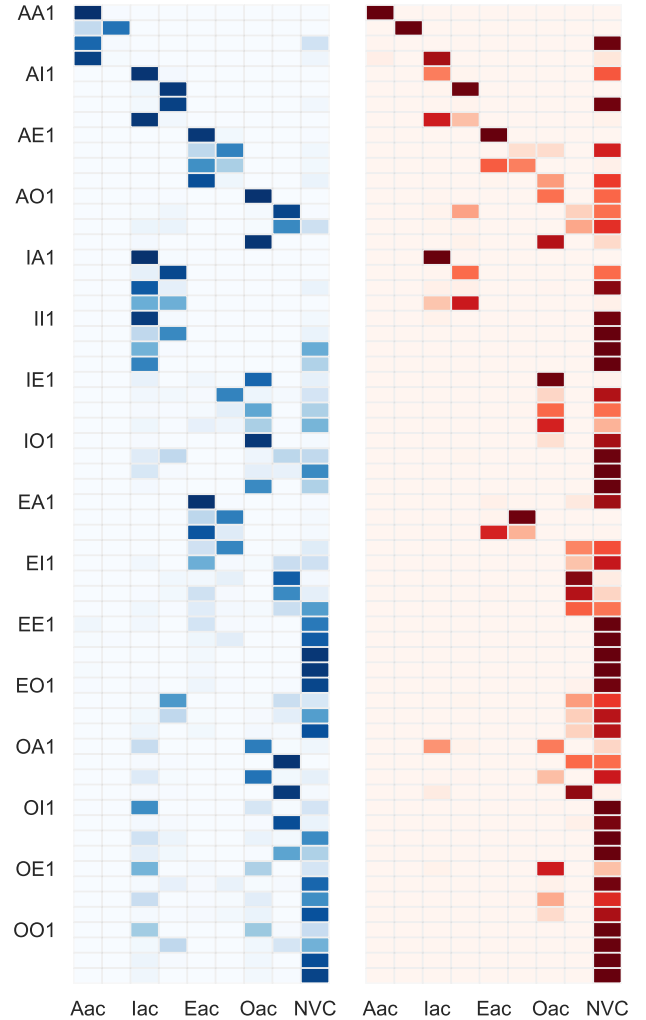


Figure 2: Both response patterns for the 64 syllogistic tasks found by clustering with Non-negative matrix factorization (for $k = 2$). Darker shades denote a higher weight of the respective response.

the respective patterns. For convenience and clarity, we will reference the two groups by *G1* (assigned to the blue pattern) and *G2* (assigned to the red pattern).

When comparing both patterns found by the NMF (see Figure 2), the main difference seems to be revolving around NVC. This is in line with previous analyses, which found the main inter-individual differences to be found with respect to NVC behavior (e.g., Riesterer, Brand, & Ragni, 2020b; Brand et al., 2022). With respect to the logical correctness, *G2* also shows a substantially higher correctness ($mean = .68$, $SD = .14$) compared to *G1* ($mean = .4$, $SD = .14$), which is expected since NVC is integral for a high correctness due to the high number of invalid syllogisms. While most differences between the patterns are just a shift towards NVC, a slight difference is also apparent for syllogisms with the quantifier *Some not* (O) in the first premise, as responses with the non-negative quantifier *Some* (I) are present for the blue pattern, while - if not NVC - only negative conclusions (*Oac* and *Oca*) are present in the red pattern. Besides these differences, the patterns seem to show identical response patterns.

Given that only two stable patterns were found that differ substantially with respect to their correctness, it is tempting to compare them with dual-processing accounts. Following the idea of dual processes and the respective implementation in mReasoner (Khemlani & Johnson-Laird, 2013), we classify the left (blue) pattern as being more likely to represent a strategy relying on *System 1* (S1), while the right (red) pattern be more frequently engaged in the search for counterexamples and thereby relying on *System 2* (S2). However, it is important to note that the correctness and number of patterns on their own do not corroborate a dual-processing account: Instead, since two stable patterns seem to emerge from the data that mostly differ with respect to NVC, it shows why models for syllogistic reasoning tend to converge to describe inter-individual effects with respect to NVC (i.e., confidences in the Probability Heuristics Model (PHM; Copeland, 2006; Riesterer et al., 2020a), NVC aversion in the model TransSet (Brand, Riesterer, & Ragni, 2020), and the search for counterexamples in MMT (Khemlani & Johnson-Laird, 2013)). Still, assuming a dual-processing account allows us to derive predictions about the groups of participants assigned to the respective patterns by the clustering method: First, it is expected that *G2* has a higher response time compared to *G1*, since relying on the deliberate inferences of S2 should be substantially slower than applying fast-and-frugal heuristics. Second, participants in *G2* should show a higher correctness in the Cognitive Reflection Test (CRT), since the test is designed to mislead participants relying on intuition. Furthermore, Need for Cognition (NFC), is also expected to be higher in *G2*, since participants with high NFC are more likely to engage in tasks that require cognitive effort and deliberative thinking.

With respect to our predictions, we investigated the differences in *Need for Cognition* (NFC) and the correctness in a *Cognitive Reflection Task* (CRT) as well as the mean response

Table 1: Overview and results of a Mann-Whitney-U test between the two groups as assigned by the NMF with respect to Need for Cognition (NFC), Cognitive Reflection Task correctness (CRT) and the mean response times (RT). Factors showing significant differences (with Bonferroni corrected $\alpha = 0.0167$) are written in bold.

	Mean		SD		U	p
	G1	G2	G1	G2		
NFC	4.65	4.73	.9	.84	1224.5	.536
CRT	.47	.7	.29	.28	747.0	< .001
RT	15803	13468	5969	6610	1697.0	.001

time needed for the 64 tasks. The results of the comparison are shown in Table 1. While NFC did not show any significant difference, the CRT differed significantly. With a mean correctness of .47, participants in *G1* were substantially more susceptible for the traps of the CRT compared to *G2* with a mean correctness of .7. This strengthens the assumption of the dual-process accounts, indicating that *G1* relies on a more intuitive process. However, the differences in response times showed that *G2* was significantly faster than *G1*. This contradicts the assumption of a slower, more logical approach using S2, but is in line with previous findings showing faster response times for NVC responses (Brand et al., 2022).

Finally, we checked how well the participants would be classified based on the NFC and CRT. To this end, we re-assigned the participants to the two patterns based on their NFC and CRT scores (using the median as a threshold). Subsequently, the accuracy of the respective pattern in predicting the participant's responses was calculated for each participant. Additionally, we included the original assignment as obtained from the NMF (*Fit*) and a post-hoc optimal assignment maximizing accuracy. Furthermore, the accuracy of the Most-Frequent Answer (MFA) strategy was added as a baseline. The MFA could thereby be understood as the result of a clustering with $k = 1$, making it useful to assess the additional gain by having an additional pattern. The results are depicted in Figure 3.

As expected from the previous analysis, NFC could not be used as an assignment strategy, even decreasing the accuracy (0.514) below the level of the MFA (0.552). However, the CRT only managed to improve the accuracy slightly (0.555), illustrating that a significant factor does not necessarily translate into being a powerful predictor on the level of individual response predictions. Finally, both data-driven assignments achieve an almost identical performance (0.599) which is a substantial improvement over both, the CRT and the MFA.

Discussion

The key goal of this article was to find and investigate stable patterns of human syllogistic reasoning behavior, which could be considered iconic for the task. Our analysis shows that first, only two patterns can be identified robustly, and sec-

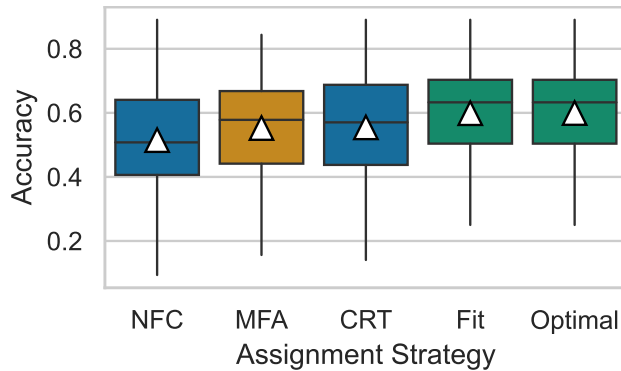


Figure 3: Accuracy achieved when comparing the individual patterns to the two iconic patterns for different assignment strategies. Individual traits are shown in blue, data-driven assignments in green. As a baseline, the most-frequent answer (MFA; orange) is added. Triangles denote the mean accuracy.

and that these patterns differ most predominantly in terms of the frequency of relying on NVC as a conclusion option.

The composition of the two identified patterns explains why most of the proposed models in syllogistic reasoning research (e.g., see Khemlani & Johnson-Laird, 2012) converge to a similar distinction between NVC-friendly and an NVC-averse participants in their response predictions. Instead, their key differences are mostly in the explanation of why these patterns emerge in syllogistic reasoning.

The fact that two patterns are found specifically seems to corroborate the dual-processing account assumptions underlying the search for counterexamples in MMT. This is further reinforced by the fact, that the reasoners associated with the more correct pattern also score high on the CRT, which is designed to assess the affinity of reasoning in a deliberative and logically correct manner. However, the observed response times associated with the patterns are contradictory to what is posited by the theory: the logically correct pattern is associated with faster instead of slower reaction times. Additionally, the CRT is known to correlate with various measures of cognitive ability (Frederick, 2005), which could also explain the a higher performance on syllogistic tasks. As a side-note, the marginal improvement achieved by using the CRT as an assignment strategy illustrated a pitfall in cognitive modeling: Even highly significant factors due not necessarily translate well to the level of predictors for individual patterns.

The results shown in this article raise the question if traditional modeling of syllogistic reasoning behavior has hit a dead end or will hit it soon. As models converge to the same patterns and only differ in their sets of explanatory assumptions, new experiments need to be designed and datasets acquired to more specifically investigate the validity or falsity of the underlying assumptions. One step towards this goal could be to integrate more auxiliary information about individuals for example via extended psychological test batteries. This could also make it possible to find additional patterns more

nuanced to smaller sub-populations of participants. Furthermore, the explanatory component of models and their underlying theories will be of greater importance, since a model comparison purely based on the general patterns will not suffice for a meaningful distinction between the models' capabilities. Instead, deriving specific hypotheses tailored to test certain assumptions of the model will become necessary.

On a technical level, our work showed that clustering, especially with flexible approaches like Non-Negative Matrix Factorization, can help to uncover expressive iconic patterns in human reasoning data. Paired with the proposed metrics, which allow to assess the robustness of the found patterns in domains where large inter-individual differences are to be expected, these approaches are valuable assets in cognitive modellers' toolkits, irrespective of the domain of interest.

Acknowledgements

This research was supported by the German Research Foundation, DFG (Grant RA1934/5-1 and RA1934/8-1) within the SPP 1921 "Intentional Forgetting" and by the Saxony State Ministry of Science and Art (SMWK3-7304/35/3-2021/4819) excellence initiative "Productive Teaming" on the basis of the budget passed by the deputies of the Saxony state parliament.

References

- Aggarwal, C. C. (2015). Cluster analysis. In *Data mining: The textbook* (pp. 153–204). Cham: Springer International Publishing.
- Bara, B. G., Bucciarelli, M., & Johnson-Laird, P. N. (1995). Development of syllogistic reasoning. *The American Journal of Psychology*, 108(2), 157.
- Brand, D., Riesterer, N., & Ragni, M. (2020). Extending TransSet: An individualized model for human syllogistic reasoning. In T. C. Stewart (Ed.), *Proceedings of the 18th International Conference on Cognitive Modeling* (pp. 17–22). University Park, PA: Applied Cognitive Science Lab, Penn State.
- Brand, D., Riesterer, N., & Ragni, M. (2022). Model-based explanation of feedback effects in syllogistic reasoning. *Topics in Cognitive Science*, 14(4), 828–844.
- Cacioppo, J. T., & Petty, R. E. (1982). The need for cognition. *Journal of personality and social psychology*, 42(1), 116.
- Chater, N., & Oaksford, M. (1999). The probability heuristics model of syllogistic reasoning. *Cognitive Psychology*, 38(2), 191–258.
- Copeland, D. E. (2006). Theories of categorical reasoning and extended syllogisms. *Thinking & Reasoning*, 12(4), 379–412.
- Dames, H., Klauer, K. C., & Ragni, M. (2022). The stability of syllogistic reasoning performance over time. *Thinking & Reasoning*, 28(4), 529–568.
- Evans, J. S. B. T. (2008). Dual-processing accounts of reasoning, judgment, and social cognition. *Annual Review of Psychology*, 59(1), 255–278.

- Frederick, S. (2005). Cognitive reflection and decision making. *Journal of Economic perspectives*, 19(4), 25–42.
- Gilhooly, K. J., Logie, R. H., Wetherick, N. E., & Wynn, V. (1993, jan). Working memory and strategies in syllogistic-reasoning tasks. *Memory & Cognition*, 21(1), 115–124.
- Johnson-Laird, P. N. (1975). Models of deduction. In R. J. Falmagne (Ed.), *Reasoning: Representation and process in children and adults* (pp. 7–54). New York, US: Psychology Press.
- Khemlani, S. S., & Johnson-Laird, P. N. (2012). Theories of the syllogism: A meta-analysis. *Psychological Bulletin*, 138(3), 427–457.
- Khemlani, S. S., & Johnson-Laird, P. N. (2013). The processes of inference. *Argument & Computation*, 4(1), 4–20.
- Kim, H., & Park, H. (2008). Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM Journal on Matrix Analysis and Applications*, 30(2), 713–730.
- Kim, J., & Park, H. (2008). *Sparse nonnegative matrix factorization for clustering* (Tech. Rep.). Georgia Tech.
- Lawson, C. L., & Hanson, R. J. (1995). *Solving least squares problems*. Philadelphia: SIAM.
- Morgan, J. J., & Morton, J. T. (1944). The distortion of syllogistic reasoning produced by personal convictions. *The Journal of Social Psychology*, 20(1), 39–59.
- Riesterer, N., Brand, D., & Ragni, M. (2020a). Do models capture individuals? Evaluating parameterized models for syllogistic reasoning. In S. Denison, M. Mack, Y. Xu, & B. C. Armstrong (Eds.), *Proceedings of the 42nd Annual Conference of the Cognitive Science Society* (pp. 3377–3383). Toronto, ON: Cognitive Science Society.
- Riesterer, N., Brand, D., & Ragni, M. (2020b). Feedback influences syllogistic strategy: An analysis based on joint nonnegative matrix factorization. In *Proceedings of the 18th international conference on cognitive modeling* (pp. 223–228).
- Riesterer, N., Brand, D., & Ragni, M. (2020c). Predictive modeling of individual human cognition: Upper bounds and a new perspective on performance. *Topics in Cognitive Science*, 12(3), 960–974.
- Rips, L. J. (1994). *The psychology of proof: Deductive reasoning in human thinking*. MIT Press.
- Störring, G. (1908). *Experimentelle untersuchungen über einfache schlussprozesse*. W. Engelmann.
- Toplak, M. E., West, R. F., & Stanovich, K. E. (2014). Assessing miserly information processing: An expansion of the cognitive reflection test. *Thinking & Reasoning*, 20(2), 147–168.

Modeling Change Points and Performance Variability in Large-Scale Naturalistic Data

Michael Collins (michael.collins.74.ctr@us.af.mil) ORISE at AFRL Dayton, OH, USA

Florian Sense (florian.sense@infinite-tactics.com) InfiniteTactics, LLC Dayton, OH, USA

Michael Krusmark (michael.krusmark.ctr@us.af.mil) CAE, Inc. Wright Patterson Air Force Base, Ohio

Tiffany Jastrzembski (tiffany.jastrzembski@us.af.mil) Air Force Research Laboratory Dayton, OH, USA

Abstract

To explain the performance history of individuals over time, particular features of memories are posited, such as the power law of learning, power law of decay, and the spacing effect. When these features of memory are integrated together into a model of learning and retention, they have been able to account for human performance across a wide range of both applied and laboratory domains. However, these models of learning and retention assume that performance is best accounted for by a continuous performance curve. In contrast to this standard assumption of models of learning and retention, other researchers have argued that, over time, individuals display sudden discrete shifts in their performance due to changes in strategy and/or memory representation. To compare these two accounts of memory, the standard Predictive Performance Equation (PPE; (Walsh, Gluck, Gunzelmann, Jastrzembski, & Krusmark, 2018)) and was compared to a Change PPE on fits to human performance in a naturalistic data set. We make several hypotheses about the expected characteristics of individual learning curves and the different abilities of the models to account for human performance. Our results show that performance that Change PPE was not only able to be better fit the data compared to the Standard PPE, but that inferred changes in the participant's performance was associated with greater learning outcomes.

Keywords: Cognitive Models; Learning and Forgetting; Change Detection; Naturalistic Data; Decay; Spacing Effect; Strategies; Long Term Learning; Individual Learning

Introduction

The study of human memory has long been a primary interest of psychology, focusing on how humans acquire, retain, and recall information over time. The findings from this research hold a great deal of promise for a variety of different applied domains, such as educational tutoring systems and adaptive training systems in medical, military and education domains. The main goal of these technologies is to attempt to determine what an individual's current ability or knowledge is on some task and then prescribe a relevant training schedule to improve or maintain their current ability over a period of time. To estimate and predict an individual's performance over time, models of learning and retention which incorporate specific features of memory are often used.

Three common features of memory which are used to account for performance over time are, the (1) power law of learning (Newell & Rosenbloom, 1980), (2) power law of decay (Rubin & Wenzel, 1996), and (3) spacing effect (Bahrick, Bahrick, Bahrick, & Bahrick, 1993). Power law of learning posits that an individual's performance improves with additional exposures to a task. The power law of decay states that

an individual's ability on a task decreases as a function of the time between instances of practice. The spacing effect states that if practice is spaced apart, knowledge is acquired at a slower rate but will be retained at a higher rate when compared to a massed learning schedule. Furthermore, models which take these regularities of memory into account often assume that individuals develop a singular representation of the task or knowledge which has a particular memory strength manipulated over time according to temporal distribution of learning schedule leading to a continuous performance curve (Walsh et al., 2018; Pavlik Jr & Anderson, 2005; Raaijmakers, 2003).

Though the assumption of a continuous learning curve is common in many different models of learning and retention and has been found to account for large amounts of empirical data (Walsh et al., 2018; Pavlik Jr & Anderson, 2005; Raaijmakers, 2003; Kumar, Benjamin, Heathcote, & Steyvers, 2022). Others have suggested that individuals display discrete and sudden shifts in performance over time, due to changes in memory representations (Collins, Tenison, Gluck, & Anderson, 2020), strategies (Gray & Lindstedt, 2017), or heuristics (Hintzman, 2011). To account for these discrete performance shifts models of learning and retention have been augmented to incorporate change points over time to account for shifts in performance. (Collins et al., 2020; Tenison & Anderson, 2016; Gray & Lindstedt, 2017).

We will now offer an overview of the predictive performance equation (PPE) as well as review the assumptions made by models of continuous learning models. Finally, we make several hypotheses about the properties of individual learning performance in a naturalistic data set collected from Luminosity and made available by Kumar et al. (2022). Understanding the variability in learning behavior in naturalistic data offers good test for psychological models attempting to account for performance over long periods of time, days, months, and years, due to the fact that most laboratory studies are conducted over a short period of time (i.e., days or weeks).

Predictive Performance Equation

The core of the PPE is composed of six individual equations, which attempt to account for the effect of the learning schedule on performance based on three features of memory previously discussed, (1) power law of learning, (2) power law of

decay, and (3) the spacing effect.

The center of the PPE revolves around the Activation term (M_i , 1), which is product of a learning (N_i^c) and a decay mechanism (T^{-d}). The learning mechanism encapsulates the number of exposures (N) raised to a constant learning rate (c)¹.

$$M_i = N_i^c * T_i^{-d} \quad (1)$$

PPE's decay mechanism is a product of model time (T_i , 3) which is a weighted sum (2) of wall clock time, raised to a decay rate (d_i , 4). PPE's decay rate accounts for the spacing effect through the use of the Stability term (ST_i , 5) which is a weighted average of all previous lags between instances of practice. When lags are closer together (i.e., massed training schedule) decay increases, and when lags increase (i.e., spaced presentation schedule) decay decreases. The effect of the stability term on decay is modified by two free parameters b and m , which modify the decay intercept and slope, respectively (4).

$$T_i = \sum_{j=1}^{n-1} w_j * t_j \quad (2)$$

$$w_i = t_i^{-x} \sum_{j=1}^{n-1} \frac{1}{t_j^{-x}} \quad (3)$$

$$d_i = b + m * ST_i \quad (4)$$

$$ST_i = \left(\frac{1}{n-1} * \sum_{j=1}^{n-1} \frac{1}{\ln(\text{lag}_j + e)} \right) \quad (5)$$

Finally, to transform the Activation term (M_i) into a probability the activation term is nested within a logistic equation (6), which is modified by free parameters τ and s controlling the performance intercept and slope. Although nesting PPE's Activation term within the logistic function is the most common use of the PPE in domains where PPE is used to track an individual's performance accuracy, it can be further modified by two more parameters to be used in events where there is no task defined maximum performance value (7). In this additional transformation, two more parameters A defines the maximum performance value and a defines the initial performance coming into a task.

$$\text{Prob}_i = \frac{1}{1 + \exp\left(\frac{\tau - M_i}{s}\right)} \quad (6)$$

$$\text{Performance}_i = a + \text{Prob}_i * A \quad (7)$$

¹For historical reasons the learning rate of the PPE is commonly held at 0.1 (Walsh et al., 2018)

Continuous or Segmented Learning

Traditionally, to explain learning over time, models of learning and retention posit a single continuous performance curve to account for an individual's learning history. These continuous performance curves can be modified according to different features of an individual learning history, such as decay and spacing (Walsh et al., 2018). Assuming a continuous performance curve has been the predominate approach to account for learning across both laboratory and real-world applications. However, positing a continuous performance curve makes several assumptions about learning and retention. First, a continuous performance curve assumes that each measurement of performance comes from the same mechanism. Second, a continuous performance curve assumes that individuals have stable sub-symbolic parameters (i.e., learning and decay rates) that account for their performance over time.

Despite the practicality of using these commonly used models of memory, the complexity of memory suggests that they might not be adequate for explaining performance over time at the individual level. Gray and Lindstedt (2017) has shown that individuals often show systematic changes over time when acquiring a skill. These systematic changes are often observed only at the individual level of performance and are lost when performance is averaged over multiple participants. Lee, Gluck, and Walsh (2019) has shown that individuals often change decision making strategies over time—to either improve their overall performance, adjust to new environment, or explore different decision making strategies. Tenison and Anderson (2016) have shown that, given enough experience, individuals transition through different phases of learning changing the mechanisms used to represent the problem over time moving from declarative to procedural.

However, each of these studies examined performance in laboratory conditions, where tasks were designed to be amenable to multiple different strategies or be consolidated by different memory representations. It is currently unknown to which degree individuals freely follow either a continuous or segmented learning curve in a naturalistic domain. Given these two different perspectives to account for an individual's performance over time, we compare these two different learning features in a naturalistic data set collected over a period of several years on the website Luminosity (Kumar et al., 2022). We fit two different versions of the PPE using either continuous or change learning assumptions to account for the individuals performance and make several hypotheses about capability of these two models.

First, **(H1)** we predict that a majority of the participants will be identified as having at last one change point in their observed performance. A change in performance could occur for a variety of different reasons, such as a change in strategy, an inability to recall their previous strategy due to a long lag between trials, a lapse in attention, or strategy exploration. Second, **(H2)** we predict that there will be a positive association between the number of inferred change points and total

number of performance opportunities. If changes in performance are associated with an increase in experience then we should expect to see more change points with increased attempts. Third (**H3**), we predict a majority of the inferred change points will be inferred after a brief lag between attempts. If changes in performance occur due to an individual's use of strategy or change in mechanism, then change points should occur close together in the learning history and not frequently occur after long lags. Finally (**H4**), we predict a positive correlation between the number of inferred change points and an individual's improvement in performance. If individuals are changing how they are completing a task in order to improve their performance, then we should see a positive relationship between the number of inferred change points and the average performance of participants.

To evaluate these predictions, we conducted a model comparison fitting two versions of the PPE. The first version is a standard version of the PPE (Continuous PPE), which estimates a continuous performance curve based on the three features of memory previously discussed. The second version of the PPE uses the same PPE model with the addition of a change detection algorithm which allows for the detection of multiple change points given an individual's performance history (Change PPE). The rest of the paper is structured as follows: we provide an overview of the Luminosity data set, the change detection algorithm, and the results from our model comparison. Finally, we review the implications of our finding to adaptive scheduling systems.

Method

Data set

A random subset ($N = 1200$) of individuals completing the Lost in Migration game on Luminosity were collected between December 2012 and October 31, 2017. The full data set was collected by Kumar et al. (2022), who formatted the data from Luminosity for research purposes². The Lost in Migration game was inspired by Erickson's Flanker Task (Eriksen & Eriksen, 1974), where individuals were shown a set of birds moving across the screen and had to report the direction that the birds were moving while ignoring the distractor signals. The data from each participant was organized into instances of performance (i.e., trials) and sessions (i.e., a set of continuous game play with a delay no longer than 1 hour). Performance was measured by the total number of correct game plays per trial. A full explanation of the organization of the data can be found in (Kumar et al., 2022).

Model Procedure

To infer the number of change points within an individual's performance data, a simple unsupervised change detection algorithm was combined with the Standard PPE (Serre, Ch  telat, & Lodi, 2020). Before the change detection algorithm could be run, a maximum number of change points had

to be determined. For this paper a maximum of 5 change points were chosen. This number of maximum change points is similar to previous research articles (Collins et al., 2020; Lee et al., 2019). Next, a genetic optimization algorithm was used to determine both the number and location of change points within an individual's performance data. Genetic algorithms are a type of optimization algorithm which are based on the features of natural selection. Genetic algorithms work by specifying a "population" of potential parameter values and then determining the "fitness" of each proposed parameter set by a user defined fitness function (i.e., RMSD, r , Likelihood). After the fitness of a set of parameters has been determined a next generation of population parameters are selected, according to mutation, cross over, and fitness values. This process is repeated until the algorithm settles on a solution that maximizes the user defined fitness function.

Once the genetic algorithm proposed a set of potential change points, the PPE's 6 parameters (b, m, s, τ, a, A) were fit to each of the individual performance segments using maximum likelihood. After the PPE was fit to each performance segment, the fitness values of fit proposed change points was evaluated. For this paper, the *BIC* of the PPE's overall fit to the participants' performance data was chosen (8). Choosing the *BIC* as a fitness function ensures that the change detection model does not over fit to the participants' data and incorporates two aspects commonly implemented in change detection algorithms. First, *BIC* takes into account the model's fit to the participants' performance using a likelihood function (*LL*), which was measured using a gamma distribution. Second, *BIC* incorporates two different penalty parameters based on the number of data points (N) and the number of free parameters (p_i) in the model. By minimizing the *BIC* the number and location of change points can be inferred and ensure the each additional change point and model parameters does not add unwarranted complexity.

$$BIC = -2 * LL(\frac{Pred_i^2}{\sigma^2}, \frac{Pred_i}{\sigma^2} | Perf_{i:N}) + p_i * \log(N) \quad (8)$$

In addition to the Change PPE we also fit the Standard PPE (Continuous PPE) to each participant's performance data using *BIC*. Note that the Continuous PPE is equivalent to the Change PPE with zero detected change points.

Results

Model Fit First, to compare the each model's fit the participants' performance, we calculated the correlation (r) and Root Mean Squared Error (*RMSD*) between both models' fit to the participants' individual performance. Overall, the Change PPE ($r = .95$, *RMSD* = 2.97) was found to better fit the participants' data compared to the Continuous PPE ($r = .89$, *RMSD* = 4.46; see Figure 1).

A visual inspection of Figure 1 shows that, overall, both models fit the average data of participants in the Luminosity data fairly well, with slight differences in the models' fits being observed. First, though both the Continuous and

²The dataset is freely available on https://osf.io/zkyr8/?view_only=cb500b45c76f448ea486dd0ec2e6ea4a.

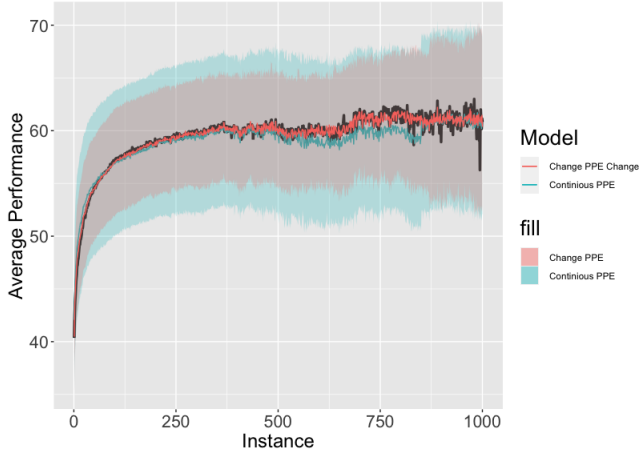


Figure 1: The average performance of participants (Black line), the average Change PPE (blue line +/- 95%CI), and Continuous PPE (red line, +/- 95%CI) fit to the participants' first 100 trials of the Luminosity data.

Change PPE fit the the participants' initial performance, differences arise between each models' ability to capture participants' later performance. The Continuous PPE has difficulty capturing the participants' variability in later performance, while the Change PPE is able to capture this variability in performance. Second, a large difference between the Continuous and Change PPE's estimates of uncertainty are observed. The Continuous PPE estimated larger confidence interval compared the the Change PPE over participants' entire performance history. Though the differences between the two models fit is minimal at the average level, larger differences between the Continuous and Change PPE are apparent at the individual participant level. For example, Figure 2 shows a participant for whom the Change PPE inferred 5 change points. From this example, the difference between the Change PPE (left panel) and the Continuous PPE's (right panel) ability to capture the participant's performance can clearly be seen. While the the Change PPE is able to capture the specific changes in the participant's performance over time, the Continuous PPE can only capture the participant's initial performance and under fits the participant's later performance.

Taken together the analysis of each models fit is unsurprising due the the fact that the Change PPE had the potential of using a greater number of parameters to fit the participant's Luminosity performance data. To ensure the inferences from the the change detection are model are warranted, the *BIC* of the Change and Continuous PPE were evaluated for each participant. For all participants where the Change PPE inferred one of more change points, the Change PPE's *BIC* was lower compared the Continuous PPE's. When no change points were inferred the *BIC* between the Continuous and Change PPE were the equal. The comparison of the *BIC* between the two models shows that the Change PPE algorithm

did not over fit the participant's data.

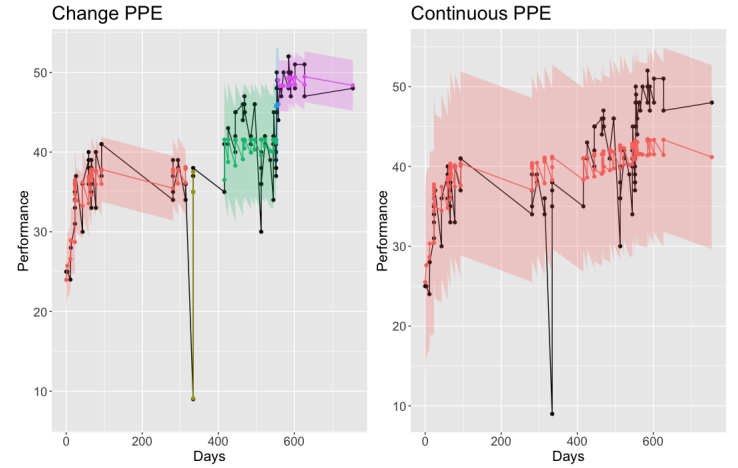


Figure 2: The performance of a single participant (black line) and the average fit (+/- 95%CI) of the Change PPE (left panel) and the Continuous PPE (right panel).

Change Detection Analysis Now we will go on to evaluate our hypotheses about the characteristics of the participants' learning profile. First, our analysis showed that 52% of participants were found to have one or more change points within their recorded performance. As can be seen in Figure 3, there is some variability in the number of change points inferred across participants having either 1 or 5 change points, with 2 change points being the least inferred number.

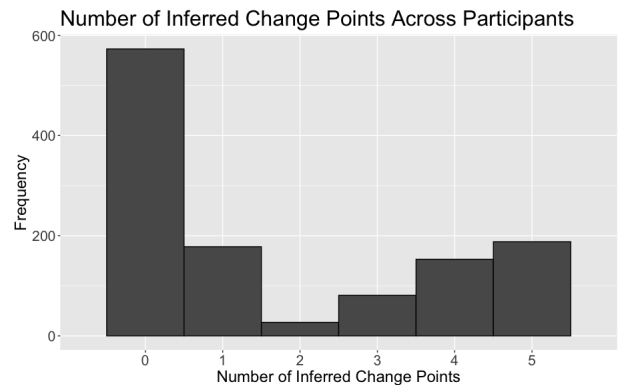


Figure 3: A histogram of the number of inferred changes points per participant across the Luminosity data set.

To investigate the relationship between number of instances of performance and number of inferred change points, we examined the correlation between total number of trials and the number of inferred change points in the participants' performance. We found that there was an overall positive correlation ($r = .48$) between the total number of trials and number of inferred change points. This suggests that as participants gain more experience with the task they either change

their sub-symbolic representation of the task or their strategy over time.



Figure 4: The total number of trials completed by a participant as a function of inferred change points

To assess when changes were inferred, we examined the time between the last trial of the previous learning segment at the first trial of a new segment (segment lag). If change points are inferred due to participants' changing their strategy or representation of the task then we would expect the lag between segments to be small, less than a day. In contrast, if change points are the results of participants' forgetting, then the lag between segments should be large. Our results show that for participants who were found to have a minimum of 1 change point, 58% of the data had a segment lag of less than one day. Finally, to assess the effect of inferred changes on the participant's overall performance we examined the relationship between the proportional change in participants performance between their initial and final performance value (i.e., improvement) and the number of change points that were inferred. We found a positive correlation ($r = .95$) between the average proportional change and the number of change points that were inferred across participants. This finding suggests that change points inferred in the participants data are associated with greater improvements in the participants' performance.

Discussion

In this paper we compared the ability of two different versions of the PPE (i.e., Continuous and Change) to account for the performance of individuals in a naturalistic data set collected from Luminosity (Kumar et al., 2022). Comparing the fit of both versions of the PPE allowed for a test of different assumptions of learning and retention. The Continuous PPE assumes that an individual's performance will follow a continuous performance curve, where the entire history of an individual is informative to determining their performance over time. In contrast the Change PPE, assumes that individual performance is more dynamic and includes sudden

change points where an individual's performance shifts due to changes in strategy, memory representation, or heuristic. In addition to comparing each model's fit to the data, several hypotheses about the expected inferences made by the Change PPE were evaluated.

Overall, the results reported in this paper support the notion that a majority of individuals in the Luminosity data show evidence for having sudden changes in their performance over time. Furthermore, we found evidence to confirm each of our four hypotheses. First, over half of the participants examined in this paper were found to have at a minimum of one significant change point within their recorded performance. This result supports the notion that a majority of participants' performance are better accounted by the Change PPE compared to the Continuous PPE. However, it should be noted though a majority of participants were inferred to have a minimum of one change point, there were still large portion of participants whose performance was best fit by a single continuous curve. The proportion of continuous to non-continuous performance curves suggests that individuals naturally display a wide range of performance curves. Second, we found a positive relationship between the amount of recorded instances of performance and the number of inferred change points. This finding supports the notion that the sudden changes in an individual's performance arise when an individual is given enough experience to refine their strategy or memory representation. Third, we found that a majority of the inferred change points occur recently after previous experience. The minimal time between instances of performance support the notion that inferred changes are caused by changes in strategy (Lee et al., 2019) or memory representation (Collins et al., 2020; Tenison & Anderson, 2016). If it was found that most inferred change points occurred after a long delay then it would support that idea that participants had failed to recall how to complete the game and had to relearn how to perform the task. Finally, we found an overall positive relationship between the number of inferred change points and the participants' improvement in performance, which strengthens our claim that performance changes are made in order to improve on the overall task and not necessarily due to lack of attention or mind wandering. Taken together the results of this paper support the idea that a majority of the participants' performance observed from Luminosity could not be accounted by a continuous performance curve and that they contain sudden and discrete shifts in performance.

Finally, the findings from this paper have implications for adaptive education systems. Most models used in adaptive educational systems attempt to utilize an individual's full performance history to generate a predictions of their future performance. However, the results presented in this paper suggests that using an individual's entire performance history might not always be the most appropriate methodology for making predictions and may hinder a model's predictive ability. Instead adaptive systems should attempt to incorporate change point detection mechanisms within a model to deter-

mine when or if an individual's performance has significantly changed as not to calibrate the model to performance which is not longer informative for their future performance.

Limitations and Future Research

Although the findings presented in this paper were informative, several limitations and future research directions should be addressed. First, due to the chosen change detection methodology used in this paper we had to choose the maximum number of possible change points which could be inferred from the participants' data. Having to choose a potential maximum change points limits the type of inferences that could potentially be made from the participants data. Future research should attempt to explore if allowing for more change points allows the Change PPE to better account for the dataset presented in this paper. Second, the results in this paper explored the ability of two models' ability to simply fit the data of participants post-hoc. If the Change PPE is going to be incorporated into adaptive scheduling systems, then these models will have to develop methods to generate predictions based on the previous instances that have been made. One possible way to incorporate predictions into our model is to utilize cross-entropy into the change detection approach (Serre et al., 2020), which measures the discrepancy between two segments of performance and can be used as a predictive weights. Finally, given the nature of this data we can not make any strong claims about the why or what mechanisms lead to an individual's performance changes, only that it is more likely that the underlying performance changed. In order to make more mechanistic inferences, we would have to use reaction time data (Collins et al., 2020).

Conclusions

Many different models of learning and retention have been developed with each model accounting for similar aspects of human memory. Though these aspects of memory have been identified through the use of laboratory studies where there is a great degree of control of both the content and history of the learning schedule. Other research has shown that as the task become more complex or the control of participants is weakened, participants will use a variety of different strategies, mechanisms, or heuristics to solve a task and the effectiveness of using only standard memory phenomena to account for behavior starts to diminish. Our research here supports these claims and highlights that in a naturalistic environment individuals' performance shows a wide range or learning and performance curves.

Acknowledgments

This research was supported by the U. S. Air Force Research Laboratory. The contents have been reviewed and deemed Distribution A. Approved for public release. Case number: AFRL-2023-2867. The views expressed are those of the authors and do not reflect the official guidance or position of the United States Government, the Department of Defense, or the United States Air Force.

References

- Bahrick, H. P., Bahrick, L. E., Bahrick, A. S., & Bahrick, P. E. (1993). Maintenance of foreign language vocabulary and the spacing effect. *Psychological Science*, 4(5), 316–321.
- Collins, M. G., Tenison, C., Gluck, K. A., & Anderson, J. (2020). Detecting learning phases to improve performance prediction. In *Proceedings of the 18th international conference on cognitive modeling*.
- Eriksen, B. A., & Eriksen, C. W. (1974). Effects of noise letters upon the identification of a target letter in a non-search task. *Perception & psychophysics*, 16(1), 143–149.
- Gray, W. D., & Lindstedt, J. K. (2017). Plateaus, dips, and leaps: Where to look for inventions and discoveries during skilled performance. *Cognitive science*, 41(7), 1838–1870.
- Hintzman, D. L. (2011). Research strategy in the study of memory: Fads, fallacies, and the search for the “coordinates of truth”. *Perspectives on Psychological Science*, 6(3), 253–271.
- Kumar, A., Benjamin, A. S., Heathcote, A., & Steyvers, M. (2022). Comparing models of learning and relearning in large-scale cognitive training data sets. *npj Science of Learning*, 7(1), 24.
- Lee, M. D., Gluck, K. A., & Walsh, M. M. (2019). Understanding the complexity of simple decisions: Modeling multiple behaviors and switching strategies. *Decision*, 6(4), 335.
- Newell, A., & Rosenbloom, P. S. (1980). *Mechanisms of skill acquisition and the law of practice*. (Tech. Rep.). CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE.
- Pavlik Jr, P. I., & Anderson, J. R. (2005). Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cognitive science*, 29(4), 559–586.
- Raaijmakers, J. G. (2003). Spacing and repetition effects in human memory: Application of the sam model. *Cognitive Science*, 27(3), 431–452.
- Rubin, D. C., & Wenzel, A. E. (1996). One hundred years of forgetting: A quantitative description of retention. *Psychological review*, 103(4), 734.
- Serre, A., Chételat, D., & Lodi, A. (2020). Change point detection by cross-entropy maximization. *arXiv preprint arXiv:2009.01358*.
- Tenison, C., & Anderson, J. R. (2016). Modeling the distinct phases of skill acquisition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 42(5), 749.
- Walsh, M. M., Gluck, K. A., Gunzelmann, G., Jastrzembski, T., & Krusmark, M. (2018). Evaluating the theoretic adequacy and applied potential of computational models of the spacing effect. *Cognitive science*, 42, 644–691.

Metacognitive Threshold: A Computational Account

Brendan Conway-Smith (brendan.conwaysmith@carleton.ca),

Robert L. West (robert.west@carleton.ca)

Department of Cognitive Science, Carleton University, Ottawa, ON K1S 5B6 Canada

Abstract

This paper will explore ways of computationally accounting for the metacognitive threshold — the minimum amount of stimulus needed for a mental state to be perceived — and discuss potential cognitive mechanisms by which this threshold can be influenced through metacognitive training.

Keywords: metacognition; threshold; metacognitive threshold; Common Model; proceduralization

Introduction

The ultimate goal of Cognitive Modeling is to build a Unified Cognitive Architecture that can simulate most, if not all, human cognitive abilities (Newell, 1994). Cognitive architectures like ACT-R and SOAR (Anderson & Lebiere, 1998; Laird, 2012) have achieved notable success in modeling knowledge-driven behaviour, however there is a scarcity of models related to phenomena surrounding metacognition. The ability for cognition to monitor and control its own processes, “metacognition,” has risen to the forefront of research in psychology, psychiatry, and AI. Modeling the results of empirical studies of metacognition is important for making progress toward accurately describing human cognition.

This paper will address a cognitive phenomenon referred to as the metacognitive threshold, i.e., the minimum level of stimulus needed for a mental state to be perceived. Specifically, we will address the variability of the metacognitive threshold, which can be reliably lowered to allow an agent improved perceptual access to their own internal cognitive states (Pauen & Haynes, 2021). The degree of an individual’s introspective acuity is also referred to as “metacognitive sensitivity”. This can be reliably improved and the metacognitive threshold lowered by way of metacognitive training such as employing mindfulness techniques (Fox et al., 2016). In cognitive psychology, mindfulness is defined as deliberate attention directed toward perceptible mental experiences, i.e., affect, sensations, thoughts, etc. (Holas & Jankowski, 2013). Greater access to and control of one’s own mental states have shown to strongly correlate with improved psychological health and overall cognitive functioning (Grossman et al., 2004; Tang et al., 2015; Rigby et al., 2014).

While decades of research strongly support the effectiveness of metacognitive techniques to influence one’s metacognitive threshold, the underlying cognitive

mechanisms have remained poorly understood. Presently there exists little or no account of this phenomena — the cognitive and computational underpinnings by which the metacognitive threshold is raised or lowered.

This paper will investigate potential computational mechanisms that may contribute to the lowering of the metacognitive threshold. In particular, we will discuss metacognitive techniques that have shown to increase metacognitive sensitivity, and explore various frameworks for clarifying their underlying cognitive constituents.

For this purpose, we will employ the Common Model of Cognition (CMC), originally the ‘Standard Model’ (Laird, Lebiere, & Rosenbloom, 2017) which provides a unified framework for investigating the fundamental elements of cognitive and metacognitive phenomena. By utilizing the Common Model and specifically ACT-R in this endeavor, we intend to address unanswered questions regarding the architecture and particularly concerning the nature of production rules.

Metacognition

Metacognition refers to the monitoring and control of cognitive processes (Flavell 1979; Fleming, Dolan, & Frith, 2012). It also involves a wide range of introspective attitudes such as confidence ratings and judgments of learning (Frazier, Schwartz, & Metcalfe, 2021; Rhodes, 2016).

Metacognitive control involves the active regulation of cognitive states or processes (Proust, 2013; Wells, 2019). This involves engaging in mental actions to either access or suppress cognitive states. Mental actions are distinct from world-oriented actions. The control of cognitive activity can involve a range of processes such as attention, emotion, planning, reasoning, and memory (Slagter et al., 2011; Efklides, Schwartz, & Brown, 2017; Pearman et al., 2020).

Metacognitive monitoring refers to the ability to recognize and identify cognitive states. It involves the perception of internal mental states such as thoughts and feelings in order to regulate those states or direct behavior. Research has demonstrated that metacognitive monitoring can be developed and improved through training (Baird, Mrazek, Phillips, & Schooler, 2014). For instance, attentional processes can be developed and enhanced through the repeated practice of attention-based tasks (Posner et al., 2015). In particular, mindfulness as a form of attention has shown to

develop through the three stages of skill acquisition defined by Fitts and Posner (Kee, 2019).

Metacognitive training such as mindfulness techniques plays a significant role in the success rates of Cognitive Behavior Therapy (CBT) and Metacognitive Therapy (MCT). Both CBT and MCT instruct patients on metacognitive strategies to monitor and regulate their own thoughts and emotions (Dobson, 2013; Normann & Morina, 2018). Research has demonstrated that those with improved metacognitive skills are better equipped to identify and manage their own disruptive and harmful thoughts and emotions (Wells, 2011, 2019; Hagen et al., 2017).

Metacognitive monitoring as mindfulness

Metacognitive monitoring and mindfulness are often used interchangeably within cognitive psychology (Holas & Jankowski, 2013). Scientific interest in mindfulness practice has become a target of interdisciplinary research and has grown exponentially over the past few decades (Tang, 2017; Van Dam et al., 2018).

Mindfulness involves the deliberate focus on perceptible experiences (sensory, affective, thought-related) and the cultivation of a dispassionate awareness of mental states and processes (Brown & Ryan, 2003; Grossman, 2010). Studies indicate a technique called detached mindfulness to be a uniquely effective therapeutic practice in developing adaptive monitoring and control over maladaptive cognitive processes (Wells & Matthews, 1994; Wells, 2005).

Detached mindfulness is characterized by the awareness of internal states (thoughts and emotions) without reacting to them — without trying to maintain or suppress them. This is achieved by way of attempting to perceive the momentary changes in mental events (such as the minute fluctuations of emotions) and letting them pass without emotional response. Mindfulness psychology contends that a significant degree of emotional distress and pathological symptoms are caused by the illusory perception of affective experience being more permanent than it actually is. This illusory perception is explained as the result of a high metacognitive threshold (poor metacognitive sensitivity) that does not allow for the subtle detection of affective fluctuations. Training in detached mindfulness aims to improve metacognitive sensitivity and one's perception of affective impermanence, also referred to as equanimity. In mindfulness therapies that do not promote equanimity, awareness alone may not be sufficient to increase subjects' psychological well-being (Cardaciotto et al., 2008). The increased, more skillful, capacity to perceive the impermanence of affective experience is considered a key mechanism responsible for decreasing emotional reactivity (Tang et al., 2015).

Metacognitive threshold

A psychophysical threshold is the minimum amount of physical stimulus needed to evoke a perceptual response in a person (Rouder & Morey, 2009). Psychophysical thresholds and their variability have been researched in domains such as sound, vision, interoception, and others (Kingdom & Prins, 2009). In metacognition research, psychophysical thresholds have been studied in reference to the minimal level of a stimulus required for a person to be aware of some mental state and make a judgment about it (Charles, Chardin, & Haggard, 2020; Sherman, Seth, & Barrett, 2018; Pauen & Haynes, 2021). These include confidence ratings as well as the subtle fluctuations of affective experience.

Generally, it is believed that an individual's metacognitive threshold is variable and can be lowered by way of training attention to perceive the momentary variations of internal cognitive states (equanimity). The training of equanimity through detached mindfulness and meditation practice has shown to be effective at lowering one's metacognitive threshold and enhancing metacognitive sensitivity.

Metacognitive sensitivity is the extent to which one is able to perceive their own mental processes or states, including thoughts, feelings, and emotions (Fleming & Lau, 2014). Mindfulness training can increase metacognitive sensitivity, allowing one to better perceive the nuances of their own feelings and thoughts.

Various metacognitive strategies and meditation techniques can allow one to practice and improve certain cognitive processes. Meditation is an umbrella term for techniques that employ deliberate focus and engage neurocognitive processes that result in advantageous effects on brain and behavior (Fox et al., 2016).

Various meditation techniques have the reported effect of enhancing metacognitive sensitivity, enabling one to perceive a weaker signal strength from internal cognitive states. In the case of developing equanimity, one becomes more capable of detecting subtle variations within emotional stimuli, such as the rapid arising and passing of feelings, thoughts, and emotions.

Meditation can involve a variety of practices. We will use Vipassana meditation as an example. Vipassana meditation (in the tradition of S.N. Goenka) is an old and popular technique that largely focuses on cultivating equanimity — a refined perception and sensitivity to the momentary impermanence of affect and sensations. Regular practice of this technique has shown to result in various cognitive advantages, such as improving executive functioning, enhancing response inhibition, and control over automatic reactions (Chambers, Lo, & Allen, 2008; Andreu et al., 2019).

The Vipassana method engages practitioners in guided meditation that directs them to maintain

attention on the impermanence of their own sensations (Kakumanu et al., 2018). During this process, practitioners monitor their affective and bodily sensations moment-to-moment, without evaluation or emotional reactivity.

Following this technique, practitioners report being able to detect increasingly subtle properties of their own mental states, including improved perceptual access to fluctuations in affect that were previously inaccessible. In other words, subjects report greater metacognitive sensitivity and a concomitant lowering of their metacognitive threshold.

Modelling the phenomena

A computational model that accounts for the phenomena surrounding the metacognitive threshold must necessarily ask questions about the fundamental nature of the architecture. Which computational components might allow one to perceive subtler properties in internal signals? Does it require us to change the way we think about certain elementary units of the cognitive architecture? We discuss these questions with specific reference to ACT-R, however, the application is intended more generally to the CMC family of architectures (Note, because the focus is on ACT-R, references are made to production systems. Other CMC architectures, such as SOAR, use more complex mechanisms, but the issues raised here remain relevant).

The ACT-R cognitive architecture fundamentally distinguishes between procedural and declarative knowledge to explain the underlying components of skill, which accords with the literature in philosophy and psychology (Squire, 1992; Christensen, Sutton, & McIlwain, 2016). Declarative knowledge is formatted propositionally and structured within semantic networks. Procedural knowledge is commonly referred to by researchers as containing “procedural representations” (Anderson, 1982; Pavese, 2019). In Anderson’s ACT-R model, procedural representations are computationally specified as “production rules” which are a dominant form of representation within accounts of skill (Newell, 1994; Taatgen & Lee, 2003; Anderson et al., 2019). Production rules, or “productions”, transform information and change the state of the system to complete a task or resolve a problem. A production rule is modeled after a computer program instruction in the form of a “condition-action” pairing. Essentially, a production rule is a “pattern-directed invocation of action” (Stocco et al., 2021). It specifies a condition that, when met, performs a prescribed action. A production is also thought of as an “if-then” rule. *If* the condition is satisfied, *then* it fires an action. Production rules are considered to be central to human intelligence and fundamental to the realization of cognitive skills (Anderson, 1993). Neurologically, production rules are associated with the

50ms decision timing in the basal ganglia (Stocco, 2018).

Modelling the metacognitive threshold

How might production rules account for an enhanced ability to detect internal cognitive signals and their variations? This could be accomplished through increasing the speed of production rules. Essentially, making productions faster, particularly productions that notice internal states, would increase the chances of picking up fleeting or intermittent signals related to emotions and noetic (epistemic) feelings, such as confidence and feelings of knowing (FoK).

A complete model of this phenomena would involve modelling internal signals, how they are detected, how they break into one’s current awareness, and how metacognitive training can improve this. Here, we discuss only how production rule acceleration can occur (however, see West and Conway-Smith [2019] for an account of how affect and noetic feelings can be incorporated into this type of model).

With regard to the speed-up of production rules, there are at least four different mechanisms that could accomplish this:

1. The ticking clock mechanism

Production rules fire when a fixed amount of time is up. The use of this mechanism produces production timing that is analogous to the intervals of a ticking clock. The timing for production firing is generally estimated to be 50ms. During this interval, productions that match the buffer conditions are identified. When this time is up, the matching production with the highest utility will fire. The timing of this process is based on neural functions that are generally considered to occur within the Basal Ganglia. Using this mechanism, production time could be sped up by shortening the clock speed. This could possibly occur through top-down feedback related to attention, as its influence has been observed in other psychophysical thresholds, such as improving perceptual sensitivity (MacLean et al., 2010).

2. The fire when ready mechanism

Production rules fire when they are ready. ACT-R is essentially a fire-when-ready model. ACT-R assumes that it takes 50ms for a production to fire, but if no production rule matches the buffer conditions, ACT-R will wait until the buffer conditions change. For example, for memory retrieval, ACT-R waits for the knowledge chunk to be delivered into the declarative memory buffer and then fires the matching production. Hence, the overall time taken is the memory retrieval time plus 50ms. However, if an alternative production matches the buffer conditions before this occurs, then it will fire instead. Using this mechanism, production firing can be made faster by using productions that do not wait for information from memory or perception.

These types of productions can be generated through the production compilation mechanism in ACT-R.

3. The narrow focus mechanism

ACT-R is capable of multitasking and even mind wandering, if the appropriate productions are available. The simplest way of producing a faster rate of firing for a specific type of production is to maintain the buffer conditions such that only this type of production can fire. Under these conditions, ACT-R can be said to model a narrow focus of attention.

4. The faster production mechanism

Some productions may be faster than others. Productions range in the complexity of their internal actions (Taatgen, 2013). The consequences of this at the neural level could imply that more complex productions take longer than simpler productions. Stewart et al.'s (2010) neural model of the Basal Ganglia estimates that this would produce a range between approximately 34ms-44ms for simple productions, and 59-73ms for complex productions. If this is the case, then the use of simpler productions would speed up the firing time.

All of these mechanisms could lower the metacognitive threshold by speeding up productions and thus increasing fidelity. To be clear, this viewpoint does not require that a threshold exists *in fact*, only that the resulting effect would appear to be so. Hence, from an architectural standpoint, this particular issue is not regarding thresholds but rather the complexity and timing of production rules. We propose that increasing the rate of production rule firing can potentially account for reports of increased metacognitive sensitivity as a result of metacognitive training, and that Common Model type architectures can model this.

Metacognitive proceduralization

The process by which simpler, faster production rules are developed through metacognitive training can be largely explained by way of metacognitive proceduralization. Proceduralization is a concept used in the skill acquisition literature to explain the cognitive mechanisms involved. It refers to the process by which a task or skill becomes automated, allowing it to be performed more efficiently and accurately, with minimal conscious effort or attention. The process involves the converting of slow declarative knowledge into fast procedural knowledge that is increasingly refined. Performance can be further improved by mechanisms such as time delayed learning, where faster productions are rewarded.

Proceduralization plays a significant role in the cognitive processes underlying skill learning in domains such as motor skill and cognitive skill (Ford, Hodges, & Williams, 2005; Beilock & Carr, 2001; Anderson, 1982; Tenison & Anderson, 2016).

Conway-Smith, West, and Mylopoulos (2023) propose that metacognitive skill develops largely through the process of proceduralization. Based on the skill acquisition model of Fitts (1964) and Anderson (1982), this model relies on the principle that skill learning within any domain is principally realized by the development and refinement of production rules. Metacognitive proceduralization proposes a mechanism by which human cognition can become more skillful at monitoring and controlling its own states, such as attention, emotion, and, we suggest, metacognitive sensitivity.

Within this framework, metacognitive skill develops through three stages (Figure 1) similar to those of Fitts and Anderson, from an early stage of instruction following to an expert stage that relies on refined, automatic procedural knowledge (production rules).

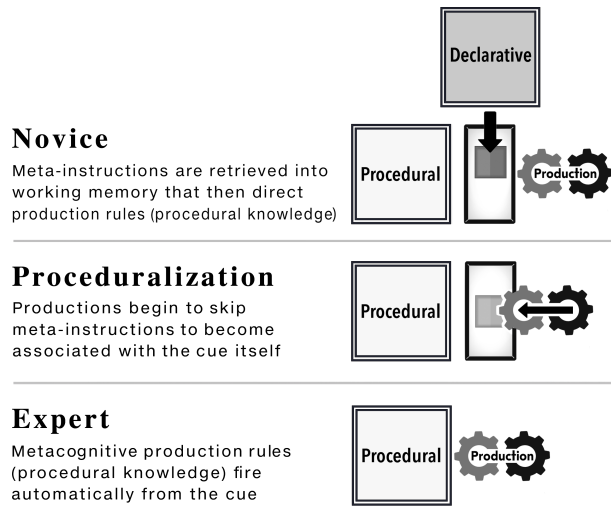


Figure 1: The three stages of metacognitive skill learning through proceduralization (Conway-Smith, West, & Mylopoulos, 2023).

The metacognitive practitioner progresses through the following three stages:

The novice stage begins with meta-instructions that direct monitoring and control resources in a specific way. In the case of metacognitive training in equanimity, meta-instructions direct the novice's attention toward the momentary fluctuations of affective experience (a feeling, sensation, or emotion). These meta-instructions are carried out by productions that retrieve them from declarative memory and execute them. Here, production speed-up could occur through mechanism 3 and possibly mechanism 1.

The intermediate stage of metacognitive training involves the process of proceduralization, where the practice of meta-instructions result in the creation of

faster production rules to accomplish the task. Specifically, repeated practice would lead to the compilation of task-specific production rules that bypass declarative knowledge. Because they are faster (due to bypassing declarative memory and possibly being less complex), these productions are more strongly rewarded and more likely to bypass the retrieval of instructions in the future. Here, speed-up occurs through mechanism 2 and possibly mechanism 4 (with mechanism 3 and 1 still in play).

The expert stage involves a robust accumulation of production rules that have been refined and stored in procedural memory. These productions can be deployed automatically to act out monitoring and control processes quickly and effectively. Here, it is possible that productions accelerated through mechanisms 2 and 4 are so deeply engrained that fast productions resulting in metacognitive monitoring and control occur spontaneously. This would result in an increased ability to monitor, even without using mechanism 3 or 1 (although 3 and 1 would still increase effectiveness if employed).

Discussion

This paper investigates the empirical phenomenon where metacognitive training can effectively lower an individual's metacognitive threshold, thereby increasing perceptual access to their own internal cognitive states. To explore the underlying cognitive and computational processes of this phenomenon, we have employed the Common Model of Cognition with a special emphasis on the ACT-R framework. In the course of this investigation we have proposed a novel method of explanation by way of metacognitive proceduralization.

References

- Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological review*.
- Anderson, J. R. (1993). Knowledge representation. *Rules of the mind*.
- Anderson, J. & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Anderson, J. R., Betts, S., Bothell, D., Hope, R., & Lebiere, C. (2019). Learning rapid and precise skills. *Psychological review*.
- Andreu, C. I., Palacios, I., Moënné-Loccoz, C., López, V., Franken, I. H., Cosmelli, D., & Slagter, H. A. (2019). Enhanced response inhibition and reduced midfrontal theta activity in experienced Vipassana meditators. *Scientific reports*.
- Baird, B., Mrazek, M. D., Phillips, D. T., & Schooler, J. W. (2014). Domain-specific enhancement of metacognitive ability following meditation training. *Journal of Experimental Psychology*.
- Beilock, S., & Carr, T. (2001). On the fragility of skilled performance: What governs choking under pressure? *Journal of experimental psychology: General*.
- Brown, K. W., & Ryan, R. M. (2003). The benefits of being present: mindfulness and its role in psychological well-being. *Journal of personality and social psychology*.
- Cardaciotto, L., Herbert, J. D., Forman, E. M., Moitra, E., & Farrow, V. (2008). The assessment of present-moment awareness and acceptance: The Philadelphia Mindfulness Scale. *Assessment*, 15.
- Charles, L., Chardin, C., & Haggard, P. (2020). Evidence for metacognitive bias in perception of voluntary action. *Cognition*.
- Chambers, R., Lo, B., & Allen, N. B. (2008). The impact of intensive mindfulness training on attentional control, cognitive style, and affect. *Cognitive therapy and research*.
- Christensen, W., Sutton, J., & McIlwain, D. J. (2016). Cognition in skilled action: Meshed control and the varieties of skill experience. *Mind & Language*.
- Conway-Smith, B., West, R. L. & Mylopoulos, M. (2023). Metacognitive skill: how it is acquired. In *Proceedings of the Annual Conference of the Cognitive Science Society*.
- Fitts, P. M. (1964). Perceptual-motor skill learning. In A. W. Melton (Ed.). *Categories of human learning*. New York: Academic Press.
- Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry. *American psychologist*.
- Fleming, S. M., Dolan, R. J., & Frith, C. D. (2012). Metacognition: computation, biology and function. *Philosophical Transactions of the Royal Society B: Biological Sciences*.
- Fleming, S. M., & Lau, H. C. (2014). How to measure metacognition. *Frontiers in human neuroscience*.
- Fox, K., Dixon, M., Nijboer, S., Girn, M., Floman, J., Lifshitz, M., & Christoff, K. (2016). Functional neuroanatomy of meditation: A review and meta-analysis of 78 functional neuroimaging investigations. *Neuroscience & Biobehavioral Reviews*.
- Frazier, L. D., B. L. Schwartz, and J. Metcalfe. (2021). The MAPS model of self-regulation: Integrating metacognition, agency, and possible selves. *Metacognition and Learning*.
- Dobson, K. S. (2013). The science of CBT: toward a metacognitive model of change?. *Behavior therapy*.
- Efklides, A., Schwartz, B. L., & Brown, V. (2017). Motivation and affect in self-regulated learning: does metacognition play a role?. In *Handbook of self-regulation of learning and performance*. Routledge.
- Ford, P., Hodges, N. J., & Williams, A. M. (2005). Online attentional-focus manipulations in a soccer-dribbling task: Implications for the proceduralization of motor skills. *Journal of motor behavior*.
- Grossman, P. (2010). Mindfulness for psychologists: Paying kind attention to the perceptible. *Mindfulness*.
- Grossman, P., Niemann, L., Schmidt, & Walach, H. (2004). Mindfulness-based stress reduction and health benefits: A meta-analysis. *Journal of psychosomatic research*.

- Hagen, R., Hjemdal, O., Solem, S., Kennair, L. E. O., Nordah, H. M., Fisher, P., & Wells, A. (2017). Metacognitive therapy for depression in adults: A waiting list randomized controlled trial with six months follow-up. *Frontiers in Psychology*.
- Holas, P., & Jankowski, T. (2013). A cognitive perspective on mindfulness. *International Journal of Psychology*.
- Kakumanu, R. J., Nair, A. K., Venugopal, R., Sasidharan, A., Ghosh, P. K., John, J. P., ... & Kutty, B. M. (2018). Dissociating meditation proficiency and experience dependent EEG changes during traditional Vipassana meditation practice. *Biological psychology*.
- Kee, Y. H. (2019). Reflections on athletes' mindfulness skills development: Fitts and Posner's (1967) three stages of learning. *Journal of Sport Psychology in Action*.
- Prins, N. (2016). *Psychophysics: a practical introduction*. Academic Press.
- Laird, J. (2019). *The Soar cognitive architecture*. MIT press.
- Laird, J. E., Lebiere, C., & Rosenbloom, P. S. (2017). A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. *Ai Magazine*.
- Stewart, T. C., Choo, X., & Eliasmith, C. (2010) Dynamic Behaviour of a Spiking Model of Action Selection in the Basal Ganglia. In *10th International Conference on Cognitive Modeling*
- MacLean, K. A., Ferrer, E., Aichele, S. R., Bridwell, D. A., Zanesco, A. P., Jacobs, T. L., ... & Saron, C. D. (2010). Intensive meditation training improves perceptual discrimination and sustained attention. *Psychological science*.
- Newell, A. (1994). *Unified theories of cognition*. Harvard University Press.
- Normann, N., & Morina, N. (2018). The efficacy of metacognitive therapy: a systematic review and meta-analysis. *Frontiers in psychology*.
- Rhodes, M. G. (2016). Judgments of Learning: Methods, Data, and Theory. In *The Oxford handbook of metamemory* (pp. 65–80). New York, NY, US: Oxford University Press.
- Rigby, C. S., Schultz, P. P., & Ryan, R. M. (2014). Mindfulness, interest-taking, and self-regulation. *The Wiley Blackwell handbook of mindfulness*.
- Rouder, J. N., & Morey, R. D. (2009). The nature of psychological thresholds. *Psychological Review*.
- Pauen, M., & Haynes, J. D. (2021). Measuring the mental. *Consciousness and Cognition*.
- Pearman, A., Lustig, E., Hughes, M. & Hertzog, C. (2020). Initial evidence for the efficacy of an everyday memory and metacognitive intervention. *Innovation in Aging*.
- Pavese, C. (2019). The psychological reality of practical representation. *Philosophical Psychology*.
- Posner M. I., Rothbart M. K., Tang Y.-Y. (2015). Enhancing attention through training. *Cognitive Enhancement*.
- Proust, J. (2013). *The philosophy of metacognition: Mental agency and self-awareness*. OUP Oxford.
- Sherman, M. T., Seth, A. K., & Barrett, A. B. (2018). Quantifying metacognitive thresholds using signal-detection theory. *BioRxiv*, 361543.
- Slagter, H. A., Davidson, R. J., & Lutz, A. (2011). Mental training as a tool in the neuroscientific study of brain and cognitive plasticity. *Frontiers in human neuroscience*.
- Squire, L. R. (1992). Declarative and nondeclarative memory: Multiple brain systems supporting learning and memory. *Journal of cognitive neuroscience*.
- Stewart, T. C., Choo, X., & Eliasmith, C. (2010, August). Dynamic behaviour of a spiking model of action selection in the basal ganglia. In *Proceedings of the 10th international conference on cognitive modeling*.
- Stocco, A. (2018). A biologically plausible action selection system for cognitive architectures: Implications of basal ganglia anatomy for learning and decision-making models. *Cognitive science*.
- Stocco, A., Sibert, C., Steine-Hanson, Z., Koh, N., Laird, J., Lebiere, C., & Rosenbloom, P. (2021). Analysis of the human connectome data supports the notion of a "Common Model of Cognition" for human and human-like intelligence across domains. *NeuroImage*.
- Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological review*.
- Taatgen, N. A., & Lee, F. J. (2003). Production compilation: A simple mechanism to model complex skill acquisition. *Human factors*.
- Tang, Y. Y., Hölzel, B. K., & Posner, M. I. (2015). The neuroscience of mindfulness meditation. *Nature reviews neuroscience*.
- Tang, Y. Y. (2017). Traits and states in mindfulness meditation. *The Neuroscience of Mindfulness Meditation: How the Body and Mind Work Together to Change Our Behaviour*.
- Tenison, C., & Anderson, J. R. (2016). Modeling the distinct phases of skill acquisition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*.
- Van Dam, N. T., Van Vugt, M. K., Vago, D. R., Schmalzl, L., Saron, C. D., Olendzki, A., ... & Meyer, D. E. (2018). Mind the hype: A critical evaluation and prescriptive agenda for research on mindfulness and meditation. *Perspectives on psychological science*.
- Wells, A. (2019). Breaking the cybernetic code: Understanding and treating the human metacognitive control system to enhance mental health. *Frontiers in Psychology*.
- Wells, A. (2011). *Metacognitive Therapy For Anxiety And Depression*. New York, NY: Guilford Press.
- Wells, A. (2005). Detached mindfulness in cognitive therapy: A metacognitive analysis and ten techniques. *Journal of rational-emotive and cognitive-behavior therapy*.
- Wells, A., & Matthews, G. (1994). Self-consciousness and cognitive failures as predictors of coping in stressful episodes. *Cognition & Emotion*.
- West, R. L., & Conway-Smith, B. (2019). Put Feeling into Cognitive Models: A Computational Theory of Feeling. In *Proceedings of ICCM 2019 17th International Conference on Cognitive Modelling*.

Extending Counterfactual Reasoning Models to Capture Unconstrained Social Explanations

Stephanie Droop¹ Neil Bramley²

Abstract

Human explanations are thought to be shaped by counterfactual reasoning but formal accounts of this ability are limited to simple scenarios and fixed response options. In naturalistic or social settings, human explanations are often more creative, involving imputation of hidden causal factors in addition to selection among established causes. Across two experiments, we extend a counterfactual account of explanation to capture how people generate free explanations for an agent's behaviour across a set of scenarios. To do this, we have one group of participants (N=95) make predictions about scenarios that combine short biographies with potential trajectories through a gridworld, using this to crowdsource a causal model of the overall scenario. A separate set of participants (N=49) then reacted to particular outcomes, providing free-text explanations for why the agent moved the way they did. Our final model captures how these free explanations depend on the general situation and specific outcome but also how participants' explanatory strategy is shaped by how surprising or incongruent the behaviour is. Consistent with past work, we find people reason with counterfactuals that stay relatively close to what actually happens, but beyond this, we model how their tendency to impute unobserved factors depends on the degree to which the explanandum is surprising.

1. Introduction

Suppose you see a friend crossing a car park, making a beeline for the end of an occluding wall behind which a food stand is often parked. But your friend stops abruptly at

the corner and changes direction. The tradition of Bayesian theory of mind uses the rationality assumption (that people act to achieve their desires given their beliefs) to work backwards to infer agents' beliefs or desires from their behaviour (Baker et al., 2007; 2017; Jara-Ettinger et al., 2020). A salient explanation for this behaviour could be the favourite food stand is absent today. But even if the stand is there, we have no problem coming up with alternative explanations: maybe seeing it reminded your friend of something more urgent she had to do; maybe she felt sick; maybe she saw someone she wanted to avoid. In everyday life, we seem to generate explanations easily and fluently, and readily draw on factors that go beyond the facts given. Natural human behaviour is complex and dynamic, driven by a hierarchy of short- and long-term goals. This presents a challenge for models of social and explanatory reasoning that often depend on a complete pre-existing model, and simplifying assumptions such as that people have stable goals and act in optimal ways to achieve them.

1.1. Explanations and Counterfactuals

A standard account of what it means to explain an event or outcome is to point to preceding event(s) that seem particularly causative for that event's occurrence on this occasion. A person might highlight a lightning strike to explain a fire in a barn over other less unique factors like the presence of hay and oxygen, while a data scientist might explain a model's classification decision on a particular fragment of its input or training data. Either way, explanations involve interrogating one's generative model of the causal relationships between the outcome and the various factors involved in producing it.

A critical component of explanation quality is whether the outcome depends on the highlighted factors not just in the actual world but also across *counterfactuals* — different ways that situation could have played out (Lagnado et al., 2013). Phrased differently, people frequently produce and find satisfying those explanations that pick out variables which robustly correlate with the outcome across a range of imagined counterfactual scenarios (Quillien, 2020; Gerstenberg et al., 2021). For instance, we more readily blame the lightning than the hay for the barn fire because many things

^{*}Equal contribution ¹Institute for Language, Cognition and Computation, University of Edinburgh, Scotland, United Kingdom ²Department of Psychology, University of Edinburgh, Scotland, United Kingdom. Correspondence to: Stephanie Droop <stephanie.droop@ed.ac.uk>.

in barns are flammable but in reality rarely catch fire without a spark. Counterfactual accounts which perturb the variables in a situation model to measure the explanatory power of different causes therefore pose a promising account of how people generate explanations. The next section discusses in detail one particular model which we build on in this work.

1.2. Counterfactual Effect Size Model

The *Counterfactual Effect Size Model* (CESM, Quillien & Lucas, 2023) operationalises the notion of simulating variations of what actually happens when selecting causal factors to mention in an explanation. The authors hold that when judging to what extent a cause C explains effect E, people first simulate counterfactual possibilities (as in the structural model proposed by Lucas & Kemp, 2015), and secondly compute the causal strength of C on E across these counterfactuals.

Important to how this works is the notion of effect size, a measure of correlation of cause with effect across counterfactuals, modelling how reliably an intervention on C would change E on average across a variety of possible background circumstances. *Ceteris paribus*, the theory is that the more strongly a cause correlates with an effect across counterfactuals, the more likely we are to posit it as an explanation for the effect.

In the CESM, the degree to which simulated counterfactuals depart from what actually happened is controlled by a stability parameter, s . When we simulate a counterfactual possibility, for each causal variable in the model, with probability s we leave the variable as it is the actual world. With probability $1 - s$, we instead sample the variable's value from its prior probability distribution, for each counterfactual we can then sample whether the effect occurs or not. Both Lucas & Kemp (2015) and Quillien & Lucas (2023) found that the value of s that maximised correlation with human selections was around 0.7.

The CESM works excellently for predicting explanations for outcomes in simple urn problems (e.g., “If I need two coloured balls to win, to what extent was drawing a blue ball from Urn 1 responsible for my win?”, etc., Quillien (2020)). The counterfactual effect size concept has also been shown to give a good account of people's estimates of how causative different states' results were for the overall 2020 US election outcome (Quillien & Barlev, 2022). However, like many probabilistic models of cognitive processes, the CESM assumes all possible explanans are enumerated from the start, leaving the cognizer to simply select which to point at. To make a start towards modelling more naturalistic explanations, such as explanations of the behaviour of other agents (“social explanation”), we investigated the free text explanations people generate when asked to explain the behaviours of agents in settings where there are a range of

potentially causal variables to latch onto.

1.3. Our Approach

We investigated social explanation using a novel paradigm where participants react to scenarios involving an agent with four salient personal and situational features, taking one of two trajectories to one of two food stands. We designed the scenarios to vary from over-determined (several variables are salient explanations for their behaviour), through singly determined (one good reason) to surprising or incongruous behaviour (no good reasons for, and several reasons against). We allowed participants to explain the agent's behaviour in each scenario using their own words and then developed a coding scheme to categorise the different explanations people gave. In this way, we explore how people generate explanations in a relatively unconstrained setting.

We are not only concerned with which and how many of the situational factors participants mention, but also with whether and when they posit additional latent causes not mentioned anywhere in the scenario. To model the human-coded responses, we implement CESM (Quillien & Barlev, 2022; Quillien & Lucas, 2023) based on a crowdsourced causal model of the general relationships between the variables manipulated in the scenarios. We build on the CESM by 1) basing our causal model on human intuitions about the relevant relationships 2) allowing for interactions between variables and 3) allowing for “other” responses that refer to latent exogenous factors.

2. Experiments

We ran two connected behavioural experiments (Figure 1). The first (Exp.1a) elicited participants' predictions about how likely a character was to take different paths to different food sources, given various biographical and environmental factors. We used these to construct a generative **situation model** that encapsulates laypeople's intuitions of the causal strengths and interactions between each factor. In the second experiment (Exp.1b) participants were shown a subset of the possible combinations of causal factors and path/food choice outcome. Participants provided a free text explanation for why each agent made that choice. We used the situation model derived from Exp.1a to predict what features of the scene participants would cite in their explanations.

2.1. Gridworld

Both experiments used the same “gridworld”, a simple graphic showing an agent walking around in a suburban environment before stopping to eat at a food stand. This was accompanied by a short biography text about the agent. We systematically varied three binary biography elements (Preference, Knowledge and Character) and one environmental

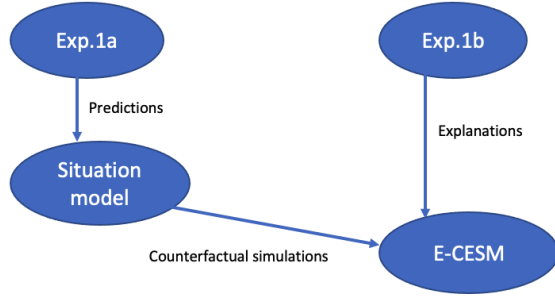


Figure 1: Design and flow of experiment and models.

Table 1: Gridworld Settings

	Factor	Values	Levels
1	Preference	0, 1	Absent, Hotdogs
2	Knowledge	0, 1	Doesn't know area, Knows area
3	Character	0, 1	Lazy, Sporty
4	Start position	0, 1	Hotdog visible, Pizza visible
5	Food choice	0, 1	Pizza, Hotdog
6	Path taken	0, 1	Short, Long

Note: Factors 1:4 describe the situational factors; 5:6 are the agent's choice that participants are asked to explain.

property (Starting position), yielding 16 scenarios, pairing these with two binary outcome variables (Food and Path), so four potential action outcomes each, or 64 explanation conditions in total (Table 1). Stimuli were the same for both experiments although their presentation was slightly different; see Section 3 and Figure 2 compared to Section 4 and Figure 3. This allowed us to systematically vary different combinations of factors and, first elicit predictions of how likely people found each outcome in each situation (Exp.1a), and second, elicit retrospective explanations for each action in each scenario (Exp.1b).

3. Experiment 1a: Predictions

The aim of Exp.1a was to crowdsource a **situation model**, a representation of people's intuitions of the causal strengths and interactions between factors 1–4 on characters' behaviour (factors 5–6).

3.1. Methods

3.2. Participants

We recruited 90 UK-based participants (42 female, 1 other, age Mean \pm sd 40.7 \pm 11.5, range 19–66) using the **Testable Minds** subject pool. They were paid \$1.60 and the experiment took Mean \pm sd 10.4 \pm 4.7 minutes.

3.3. Design

All participants saw all 16 scenarios (specified by Factors 1:4 in Table 1) one by one in a random order. For each trial participants rated the probability of the four possible outcomes. The presentation position on screen of these four was counterbalanced between participants to minimise any left-right bias.

3.4. Stimuli

3.4.1. BIOGRAPHIES

Biography stimuli were eight unique short texts about the agent in the gridworld, varying across three factors (Factors 1:3 in Table 1), each with two levels: *Preference*: {"X's favourite food is hotdog", absent}, *Knowledge*: {"X knows the area well", "X doesn't know the area well"} and *Character*: {"X is sporty", "X is lazy"}. An example is: "Jesse's favourite food is hotdogs. They do not know that area well and are sporty." The agent's name was different for each biography to ensure participants treated each trial and agent independently. Unisex names were used to minimise influence of gender stereotypes.

3.4.2. GRIDWORLD ENVIRONMENT

The gridworld stimuli depicted an agent in a stylised 2D world with houses in the middle and a road around the perimeter. The basic environment was always the same, with a hotdog stand at top right and a pizza stand at bottom left. Three factors were manipulated visually (Factors 4:6 in Table 1): starting location of the agent ({top left, bottom right} aka hotdog visible, pizza visible), and then the two outcome factors depicted with a red arrow: the agent's destination ({hotdog stand, pizza stand}), and the length of the path the agent takes ({long, short}). In Exp.1a, the four possible choices or action outcomes as shown by the red arrow ({short path to hotdog stand, short path to pizza stand, long path to hotdog stand, long path to pizza stand}) were presented all at the same time.

3.5. Procedure

The experiment was implemented in **Testable** and participants completed it in the browser on their own devices. After calibrating their computer screen, they were presented with the study's information sheet and consent form. See Figure 2 for trial flow schema. Once consent was accepted, participants were given instructions for completing the experiment and shown an example of the stimuli. Importantly, participants were informed that the depicted character could not see through the houses in the middle of the grid-environment. A button was then presented to begin the experiment.

Participants were first shown the agent's biography accom-

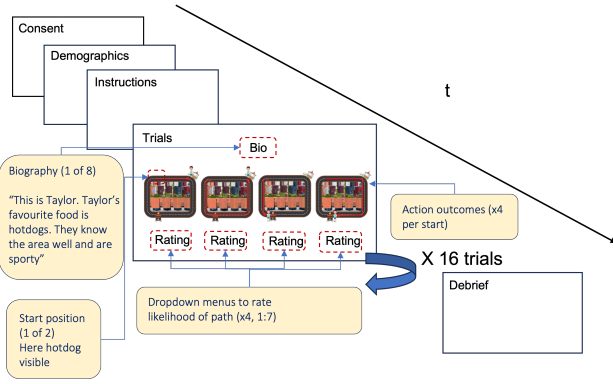


Figure 2: Experiment 1a

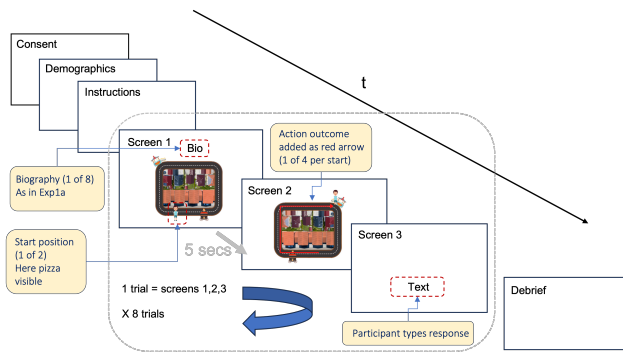


Figure 3: Experiment 1b

panied by the instruction, “They go for a walk and stop to eat at a food stand. Remember they cannot see through the houses or round a corner. Where do you think they will go? Show how likely each path is by rating each between 1 (not likely at all) and 7 (very likely)”. They were then shown each of the four outcomes in counterbalanced order, with a dropdown menu box below each offering the integers 1 to 7. Each participant rated each biography once for each starting position, completing 16 trials in random order.

3.6. Analysis

In Exp.1a, participants had rated how likely from 1 to 7 they thought each of the four possible choices (2 foods \times 2 paths), given the character’s biography and starting position. These crowdsourced likelihood ratings became the beta slopes of our situation model. To calculate them, we first normalised each participant’s ratings to sum to 1 across the four actions for each trial. For example, if they answered 7 for the short path to pizza and 1 for all others, on that trial the short path to pizza was rated 0.7. If they answered that each action was equally likely, then each action was normalised to 0.25, regardless of whether they were all rated

1, 7, or something in between. We then fit two separate generalised logistic mixed-effect regressions, one for each dimension of the action: **food choice** (whether the person went for a pizza or a hotdog) and **path choice** (whether they travelled the shorter or longer way). We included random intercepts for participants. We selected the final model for each dimension using a stepwise procedure implemented by [timnewbold/StatisticalModels](#). By combining the two regressions additively, we obtained the probability of each of the four actions for each situation. This gives rise to the **situation model** which is an intermediate stage in this paper.

3.7. Results: Predictions

Participants in Exp.1a saw each combination of factors in each scenario, and rated how likely was each outcome of food choice and path choice. This means we can use their responses to fit a structural equation model capturing the relationships between the causes and the potential outcomes. Concretely, this **situation model** was a combination of two logistic mixed effects regressions for which we selected the main effects and interaction terms using stepwise model selection. For simplicity, we assumed independent influences of the causes on the choice to take the longer or shorter path and the choice of destination. The resulting model, one outcome of one setting of which is visualised in Figure ?? and another outcome of the same setting in Figure ?? had main effects of Preference, Character and Start Position on food choice, as well as significant interactions, between area Knowledge and Start position, and Character and Start position. Only Knowledge and Character significantly influenced the predicted path length. The resulting model assigns a probability to all four outcomes in each of the 16 scenarios. The odds ratio parameters for each edge can be interpreted straightforwardly as causal influence strengths raising or lowering the probability of the different outcomes. For example, Preference’s weight of 5.2 means that, other things being equal, a preference for hotdogs increases the probability of the agent going to the hotdog stand by about a factor of 5.

4. Experiment 1b: Explanations

4.1. Participants

We recruited 49 UK-based adults (40 female, age Mean \pm sd 21.1 \pm 9.6, range 18-81) using SONA systems online recruitment and a [Reddit board](#) for recruiting experimental subjects. Participants were not paid. The task lasted Mean \pm sd 10.4 \pm 7.9 minutes.

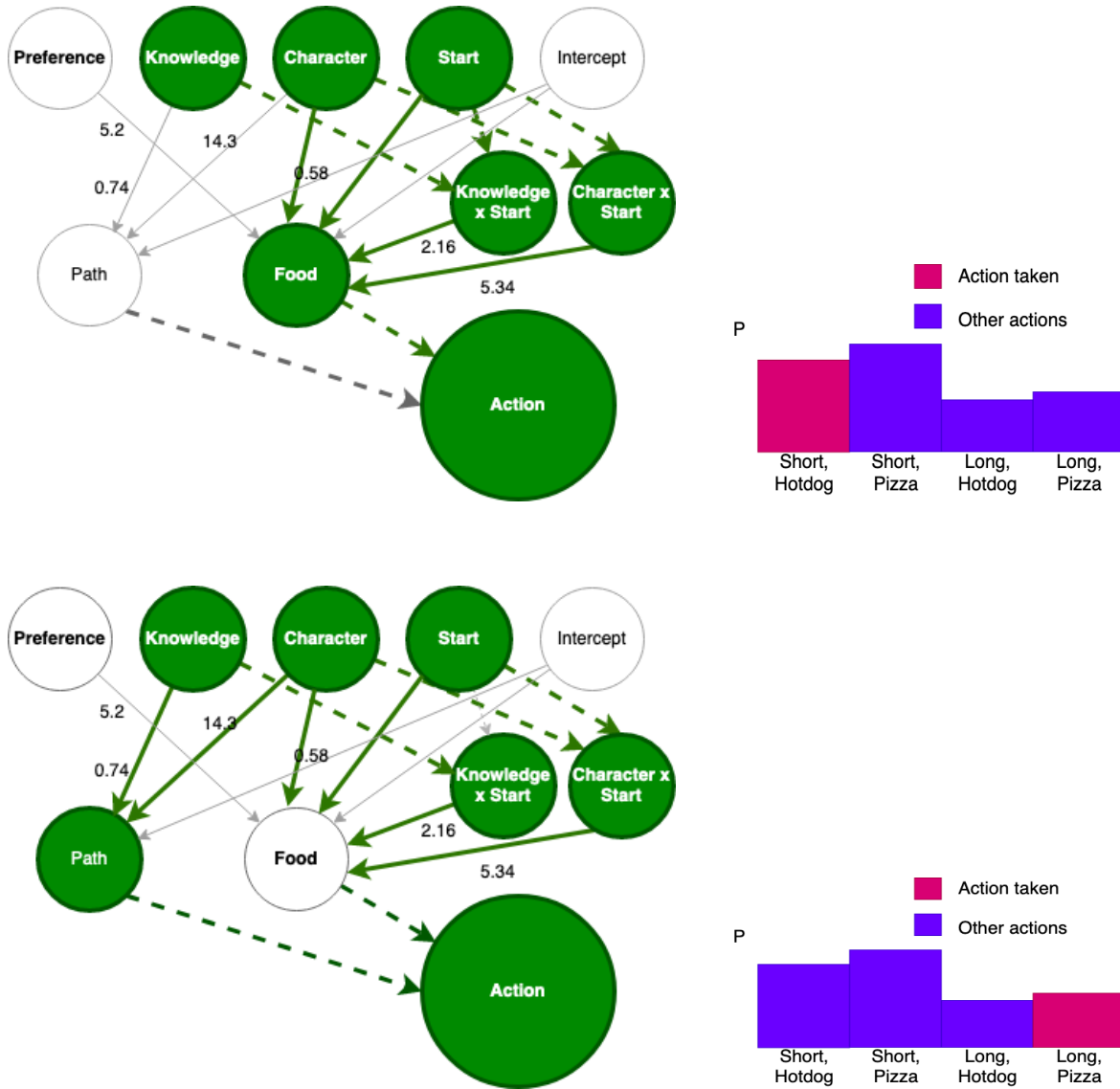


Figure 4: Schema of one setting of the situation model, with two outcomes. Nodes along top left are green for 1 and clear for 0. Here preference is 0, indicating no mention of liking hotdogs. Edge annotations are odds ratios. Solid edges show fitted regression slopes; dotted lines show logical and inferred relationships.

4.2. Design

The 64 stimuli were split into eight groups of eight using a pseudo Latin Square approach. Each participant thus saw one of each of the eight grid configurations and one of each of the eight biographies, but across the sample as a whole all combinations of biography and grid configuration appeared a roughly equal number of times.

4.3. Stimuli

Stimuli were the same gridworlds as Exp.1a.

4.4. Procedure

As per Exp.1a, until presentation of stimuli. Thereafter as per Figure 3. Participants were first shown the agent's biography and starting position in the grid environment. After a few seconds, a red arrow was added to show the agent's choice. At the same time, a text box appeared with the following question written above it: "What do you think is the single best explanation for the person's chosen path?". Once the participant typed their answer, they were presented with another trial with a different stimulus. Each participant saw eight separate trials.

4.5. Analysis

Data were analysed using R version 4.1.

4.5.1. TEXTUAL ANALYSIS

Free text responses were stripped of participant and trial data and coded by a research assistant naive to the experiment, by placing a “1” in the relevant column, for whether the participant cited explanations from the biography and situation. The categories were: the agent’s Preference (e.g. “They got a hotdog because they like hotdog”), Knowledge of the area, Character (e.g. “They went the long way because they are sporty so they probably wanted a walk before dinner”), their Starting position (a particular food stand was either closer or within sight), or Other which ranged from personal (“He just wanted hotdog today”) to situational (“The hotdog stand was closed that day”); see Section 4.7.1 for actual examples. These ratings were then compared to the model predictions after the modelling detailed in the next section.

4.5.2. EXTENDED COUNTERFACTUAL EFFECT SIZE MODEL (“E-CESM”)

We adapted the CESM to apply to our gridworld setup (Figure 5). To obtain model predictions, for each of the 64 gridworlds we simulated outcomes for 1000 counterfactual worlds, for whom the overlap of causal variable states with the actual scenario was governed by a stability parameter as in Quillien & Lucas (2023). For each simulation, the outcome was sampled according to the probability of that action given by the situation model (Figure ??), creating j rows of sampled counterfactuals for i columns of causes in matrix CF.

Then we calculated the correlation between each causal variable and the actual outcome across these counterfactual worlds. To do this, we looped over i , comparing the subset of the counterfactuals for which the variable in question is 1 ($CF[C_i == 1]$) to those where $CF[C_i == 0]$ and so matching relative proportion of getting the same effect as was actually obtained in the two subsets of counterfactuals.

We optimised stability parameter s through grid search (it was computationally expensive to optimise directly), generating predictions for the model separately for 19 values of s in steps of .05 from .05 to .95. Additionally, in fitting the model to the participant data from Exp.1b, we directly optimised two additional parameters: a parameter τ_1 controlling the probability of providing an explanation that pertains to something not manipulated explicitly, and a softmax temperature parameter on selection τ_2 . We additionally hypothesised that the probability of reaching beyond the provided dimensions could be related to how surprising the actual outcome was (i.e. how hard it was to explain in terms of the provided factors). As such we modelled

Table 2: Comparison of our Extended-CESM model (top) with the others explained in Section 4.5.3.

MODEL	τ_1	τ_2	s	NLL	BIC
E-CESM	.364	.168	.7	496.4	1010.7
CESM	.228	.127	.8	497.2	1012.3
DD	1.05	1.01	-	545.8	1103.5
BASELINE	-	-	-	630.9	1261.8

the probability of an explanation being classed as Other as $\tau_1(1 - P(Outcome))$. These parameters were optimised with Nelder-Mead as implemented by R’s `optim` function.

4.5.3. ALTERNATIVE MODELS

For comparison, we also ran the same model predictions through a modified function where propensity to cite Other causes was governed by just a flat τ_1 value rather than being modulated by $1 - P(Outcome)$, assigning the same probability to the other category for all explanations. This represents the classic CESM although that has no provision for outside factors. We also implemented a direct dependency model (“DD”) where counterfactual dependence was established just one factor at a time; Finally we calculated a baseline fit which is simply the log likelihood of falling into a category at chance ($\log(1/5)*392$). Results and model comparison are shown in Table 2. Code can be found in our [Repository](#).

4.6. Results: Explanations

The results of Exp.1b consisted of free text verbal explanations for why the character in the gridworld scenario acted the way they did. We used our E-CESM to predict what people would likely mention. Our model had a negative log likelihood of 496.4 and a Bayesian Information Criterion of 1010.7. See Table 2 for how this compares with the baseline and lesioned versions, and Figure 6 for how the final model predictions compare to actual participant data.

The stability parameter s fit best at 0.7, indicating that when simulating counterfactuals, variables kept their original value 70% of the time.

4.7. The Importance of “Other”

In some gridworld settings, participants predominantly answered “Other”; these were cases where the character’s choice was surprising given their biography and starting position (as can be seen towards the lower right of Figure 6, which is ordered by increasing unexpectedness of the character’s behaviour). For example, the subplot at the bottom right corner represents gridworld 110001, where the character chose the long path to pizza, despite having a preference

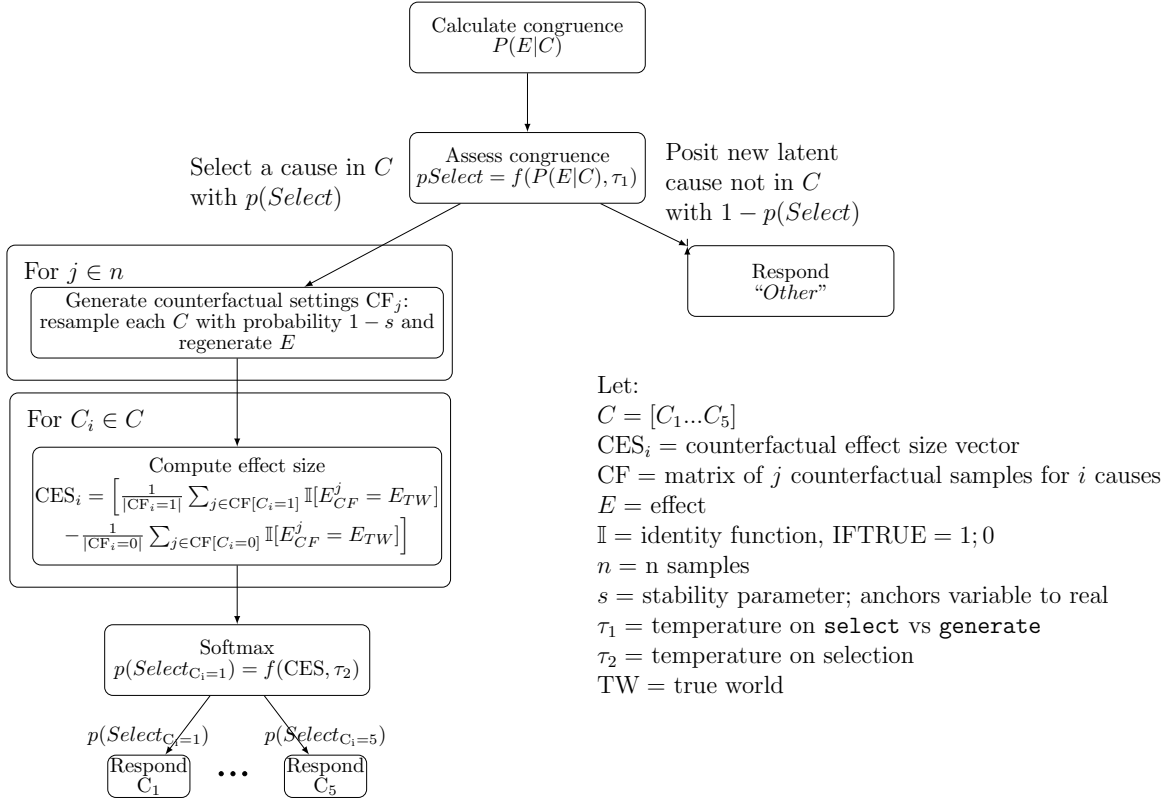


Figure 5: Mixture (process) model of E-CESM. The model either selects from available causes (left branch) or generates a new latent cause (right branch) as a function of how surprising (ie. improbable) the action was.

for hotdog, knowing the area, being lazy, and starting in a position near a visible hotdog stand. Their behaviour could therefore be seen as maximally incongruent with the given facts of the situation and so we would expect both the model and the participants to need to cite Other causes in order to adequately explain the character's choice.

4.7.1. EXPLORATORY ANALYSIS

We performed exploratory qualitative analysis of the “Other” column of the text responses to gain initial insight into any patterns of salient factors. We observed that participants often mentioned temporary changes in the more stable biography factors as well as aspects of the situation. An RA coded each response for mention of temporary *Food state*, *Character state*, *Other state* and *Other situation*.

Out of 392 text responses received, 219 mentioned some kind of Other factor, of which:

- 41 mentioned some kind of temporary desire for a food, e.g. “He had a hotdog recently and wanted a change”, “Changed their mind and wanted pizza”, “He wanted to catch the aroma of pizza to stimulate his tastebuds before a hotdog”.

- 28 mentioned a temporary character state related to those in the biographies but opposed to the current character's biography, e.g. “He was in a lazy mood”, “He decided to do exercise for a change”.
- 38 mentioned other temporary character states, e.g. “They are tired today”, “They changed their mind once they got there”.
- 52 mentioned something about the situation outside the person's goals, e.g. “Charlie was procrastinating an assignment”, “The hotdog van was closed that day”, “They had other things to do in the area”, “They got lost”.

5. Discussion

In this paper we explored how people explain behaviour in an ecologically richer and more open ended setting than has previously been analysed with experiments and causal models. One way our setup differs from many past scenarios set up is in its coverage: where [Lombrozo \(2006\)](#) and [Lucas & Kemp \(2015\)](#) used scenarios where the behaviour generally made sense under the intended causal situation model, we presented people with a “fully balanced” set of scenarios



Figure 6: Model predictions (red dots) against participant ratings for each of the 64 gridworld settings, ordered here by the predicted probability of the character’s choice. Facet names encode the condition in following sequence: Preference, Knowledge, Character, Start position, Food choice, Path taken following the level conventions in Table 1. For instance the first facet “100010” shows the condition in which the agent likes hotdogs, doesn’t know the area, is lazy, can see the hotdog stand, and goes to the hotdog stand by the shorter route.

where all variables were combined with all values of each other.

We first crowdsourced a general model of the situation by asking people to rate how likely the four possible outcomes were for each set of starting values. This revealed that certain behaviours are more or less surprising (why, for example, would a lazy person who has no special preference for hotdogs take the long way round to a hotdog stand they can already see?). Eliciting judgements from people in this way reduces some sources of experimenter-driven assumptions about how people understood the scenario in the task. We then showed new participants each situation-outcome pair, and elicited free explanations. These text responses often made reference to factors from the situation, but also often brought in imaginative reasons from outside the scenario, especially when the behaviour was incongruent. Our

extended model could capture that, to some extent, these Other factors tended to dominate the explanations when the behaviour was surprising under the model.

5.1. Comparison with the CESM

We generalise the CESM (Quillien, 2020; Quillien & Lucas, 2023) to a more open-ended setting. Our findings thus offer support for that model and an attempt to bridge the simple and quantified setting of sampling coloured marbles from urns (Quillien & Lucas, 2023) and real world issues like the 2020 US presidential election outcomes (cf. Quillien & Barlev, 2022). Like that study we attempt to bring explanation theories closer to the real world. Unlike it, however, our experimental dataset and modelling is based on human intuitions about the situation rather than a complex statistical model. Their situation model was not proposed to match

people’s mental models, whereas ours is.

In that light it is noteworthy that the the best fitting stability parameter $s = .70$, is numerically close to the .73 value found in Quillien & Lucas (2023) and .53 in Lucas & Kemp (2015) for their own experimental data and .77 for their reanalysis of Rips (2010). As such there appears to be some converging support for the idea that counterfactuals humans entertain involve resampling causal variables around a third of the time.

5.2. “Other” Causes

Our results also demonstrate that people are rarely unable to generate explanations, even for ostensibly unlikely or surprising behaviours. This shows everyday explanations are considerably richer and more creative than they might appear in tasks that fix the response options to a set of provided causes (e.g., Lombrozo, 2006; Pacer & Lombrozo, 2017). Our results mesh with research suggesting people both tend to overspecify causal relationships when explaining things, and often prefer comprehensive, overdetermined explanations (Zemla et al., 2017), i.e. referring to far more variables than necessary. Although every Bayes net has to simplify its corner of the world and draw artificial lines around the boundaries of a causal system, in reality no system is closed, and people are sensitive to this and able to cast around outside a presented option set.

Our exploratory text analysis suggests social explanations often reference a mixture of individual factors (e.g., personality, preference, etc.) and situational factors (e.g., environmental affordances, distance, convenience, etc.). This brings to mind the long history in social psychology of theories that seek to explain behaviour by making a distinction between *dispositional* factors (those internal to an agent, e.g. ability, knowledge, goals) and *situational* factors (outside the agent’s control, e.g., environment, societal pressure) (Heider, 1958/2013) in addition to the later *fundamental attribution error* (Ross, 1977) and *correspondence bias* (Gilbert & Malone, 1995) where people cite situational factors for their own failures, apparently unwilling to concede they may have acted irrationally, but happily cite character or disposition for incongruent behaviour in others. While the CESM does not make any predictions about which would take precedence, and our study was not set up to compare rates at which people are subject to the fundamental attribution bias, exploring self-other differences in what variables are selected in explanations is an avenue for future work.

5.3. Limitations

Limitations to this approach include our simplifying choice to treat the two components of participants behaviour as causally independent, and the relatively small sample size for Exp.1b. Once more data is collected, some of the noise

seen between the model predictions and the participant data in Figure 6 should dissipate. We acknowledge the age and gender imbalance between the participant samples of Exp1a and Exp1b, due to 1b being mostly undergraduates and unpaid, but this type of higher-level cognition is not known to have any age or gender differences. Finally and most importantly, although E-CESM is a step towards a model of higher-level cognition, it still only predicts a single “catch-all” Other category rather than truly modelling the flexibility and dynamism of human cognitive processes. However, modelling is in progress toward a generative explanation model which is able to impute hidden variables, similar to the edge replacement technique of Buchanan et al. (2010); Buchanan & Sobel (2014).

5.4. Conclusion

In this paper we presented a computational model of how people explain more or less surprising behaviour. This involved a small extension to an existing counterfactual model of causal selection, enabling it to cover the content of natural language explanations in a naturalistic setting. We combined this with an open-ended free text response format to obtain a richer view of spontaneous explanation, in addition to crowdsourcing a situation model, thereby minimising our need for experimenter-set parameters. Our extension to modulate Other by the probability of the outcome fit provided a modest but encouraging improvement in fit over CESM, a Direct Dependency counterfactual model and a Baseline, making it a promising start toward richer models of human explanation.

References

- Baker, C. L., Tenenbaum, J. B., and Saxe, R. R. Goal inference as inverse planning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 29, 2007.
- Baker, C. L., Jara-Ettinger, J., Saxe, R., and Tenenbaum, J. B. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):1–10, 2017.
- Buchanan, D. and Sobel, D. Edge replacement and minimality as models of causal inference in children. In *Advances in child development and behavior*, volume 46, pp. 183–213. Elsevier, 2014.
- Buchanan, D., Tenenbaum, J., and Sobel, D. Edge replacement and nonindependence in causation. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 32, 2010.
- Gerstenberg, T., Goodman, N. D., Lagnado, D. A., and Tenenbaum, J. B. A counterfactual simulation model

- of causal judgments for physical events. *Psychological Review*, 2021.
- Gilbert, D. T. and Malone, P. S. The correspondence bias. *Psychological bulletin*, 117(1):21, 1995.
- Heider, F. *The psychology of interpersonal relations*. Psychology Press, 1958/2013.
- Jara-Ettinger, J., Schulz, L. E., and Tenenbaum, J. B. The naive utility calculus as a unified, quantitative framework for action understanding. *Cognitive Psychology*, 123: 101334, 2020.
- Lagnado, D. A., Gerstenberg, T., and Zultan, R. Causal responsibility and counterfactuals. *Cognitive science*, 37 (6):1036–1073, 2013.
- Lombrozo, T. The structure and function of explanations. *Trends in cognitive sciences*, 10(10):464–470, 2006.
- Lucas, C. G. and Kemp, C. An improved probabilistic account of counterfactual reasoning. *Psychological review*, 122(4):700, 2015.
- Pacer, M. and Lombrozo, T. Ockham’s razor cuts to the root: Simplicity in causal explanation. *Journal of Experimental Psychology: General*, 146(12):1761, 2017.
- Quillien, T. When do we think that x caused y? *Cognition*, 205:104410, 2020.
- Quillien, T. and Barlev, M. Causal judgment in the wild: evidence from the 2020 us presidential election. *Cognitive Science*, 46(2):e13101, 2022.
- Quillien, T. and Lucas, C. G. Counterfactuals and the logic of causal selection. *Psychological Review*, 2023.
- Rips, L. J. Two causal theories of counterfactual conditionals. *Cognitive science*, 34(2):175–221, 2010.
- Ross, L. The intuitive psychologist and his shortcomings: Distortions in the attribution process. In *Advances in experimental social psychology*, volume 10, pp. 173–220. Elsevier, 1977.
- Zemla, J. C., Sloman, S., Bechlivanidis, C., and Lagnado, D. A. Evaluating everyday explanations. *Psychonomic bulletin & review*, 24:1488–1500, 2017.

Modeling a Human-AI Cooperation Task in ACT-R

Tanishca Sanjay Dwivedi (tfd5326@psu.edu)

Department of Industrial and Manufacturing Engineering
The Pennsylvania State University
University Park, PA 16801 USA

Christopher L. Dancy (cdancy@psu.edu)

Department of Industrial and Manufacturing Engineering & Department of Computer Science and Engineering
The Pennsylvania State University
310 Leonhard Building, University Park, PA 16801 USA

Keywords: ACT-R, Conceptnet, Human-AI Cooperation, Pig Chase

Introduction

How can we model the ways race-based systems of power and oppression impact the ways people interact with AI agents? To approach this question, we are developing a computational model of a human-AI interaction study that explores the impact of racialization on such interactions. There has been a variety of discussions related to ways in which implicit racial biases affect interactions between people and interaction between people and artifacts; some studies have shown this is not limited to phenotypical perception of racialized people (e.g., Atkins, Brown, & Dancy, 2021). Our initial perception and racialization of an individual is shaped by numerous cues, ranging from their physical appearance and social identity to their behavior (such as facial expressions, gestures, proximity), and even their scent. Additionally, situational factors can also play a role in how we interpret an individual's behavior (Kawakami, Amodio, & Hugenberg, 2017). Our understanding of an individual's actions is influenced not only by our perceptions of their behavior but also by the assumptions we make about how they are likely to act based on the impression we have formed (Macrae & Bodenhausen, 2000). Providing race as a cue not only influences how we interpret an individual's behavior but also the behavior of AI agents.

Atkins et al explored whether people's decision to cooperate with an AI agent during the Pig Chase Task (a modified version of the Stag hunt task) is affected by the knowledge that the AI agent was trained on behavioral data from people who identify and/or are racialized as [Black/African American, White/Caucasian]; in the *control* condition, racialization of data was not mentioned (though racialization can occur nonetheless, given the racialization of AI more broadly, Cave & Dihal, 2020). Unbeknownst to the participants, the AI agent used an A* algorithm to complete the task and hadn't been trained on any human behavior. The data showed that participants who identified as White performed the best when the agent was racialized

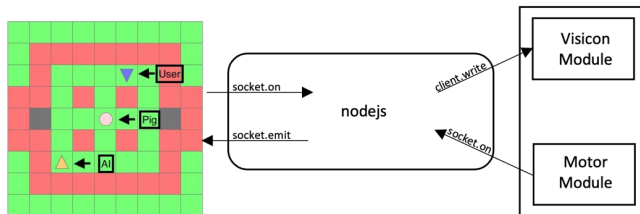
as White and not racialized at all, while participants who identified as Black achieved the highest score when the agent was racialized as Black. Qualitative data indicated that participants who identified as White were less likely to report that they believed that the AI agent was attempting to cooperate during the task and were more likely to report that they doubted the intelligence of the AI agent (when compared to participants who identified as Black) (Atkins, Brown, & Dancy, 2021).

Given the social and economic costs, how can we identify why the participants (based on their treatment group) may have exhibited certain responses? In this work we've been exploring how ACT-R can help us understand the socio-cognitive processing that leads to participants making certain decisions over others while performing the task.

Connecting ACT-R to the Environment

We're developing an ACT-R model that connects with the existing study infrastructure and code to complete the Pig Chase task. To establish a connection between the Pig Chase Task environment and ACT-R we use NodeJS, a backend JavaScript runtime environment, and particularly the Socket.IO library. Socket.IO is a library that enables low-latency, bidirectional and event-based communication between a client and a server. It opens a communication channel between ACT-R and the environment that allows the model to receive information about the location of the agents (in the visicon module) and respond to those movements (through the motor module).

The code sets up a server using the Node.js and Express framework, similar to the example server code included with recent versions of ACT-R. This server then communicates with the ACT-R environment, updating visicon features based on javascript objects provided by the Pig Chase task code. It also interprets incoming ACT-R messages and commands so that the model can *interact* with the environment. We were able to facilitate communication with small injections of code at key interaction points in the Pig Chase Task Javascript environment.



Conclusion & Future Work

Connecting the existing Pig Chase Environment to ACT-R provides us with an opportunity to understand and in turn lay out the process of recognizing the steps taken to make certain decisions, particularly without creating another environment, something that can be especially time consuming for the computational cognitive modeling process (Dancy & Ritter, 2017). The data comprising the movements of participants from the pig chase experiment in Atkins et al would be used to feed information into the ACT-R model. We have collected data from 1008 participants for 21 identity x treatment interactions. Those data will be used to test and validate aspects of the cognitive model. However, modeling of the cognitive aspects of the decision-making process is still limited by the local knowledge representations used in the typical ACT-R models. ACT-R can provide a computational account of the processes that lead to the steps taken to perform the task but does not (by default) delve deeper into the interaction the participants have with the information prompt provided about the race (Black, White, None) the AI agent has been trained on. Thus, there lack clear ways to include the influence of sociocultural systems that are important to everyday behavior. To develop a more complete simulation of related human behavior with greater resolution, we are developing a connection between ConceptNet (Speer et al., 2017) and ACT-R. ConceptNet is an open-source knowledge graph that combines knowledge from several sources including crowd sourcing, certain games, and existing online sources such as DBPedia, which makes it a potentially useful declarative knowledge source “out of the box” (Dancy, 2022). Integrating it with our model will provide the model with existing historical and sociocultural perspectives that we otherwise would have to build in manually, which would likely lead to a less reusable model for other tasks. This provides us with a more realistic ability to understand the interaction between the user and the environment after it has the knowledge of the race of the AI agent.

References

- Atkins, A. A., Brown, M. S., & Dancy, C. L. (2021). Examining the Effects of Race on Human-AI Cooperation. In *proceedings of the 14th International Conference on Social Computing, Behavioral-Cultural Modeling, & Prediction and Behavior Representation in Modeling and Simulation*, Virtual, 279-288.
- Cave, S., & Dihal, K. (2020). The Whiteness of AI. *Philosophy & Technology*, 33, 685-703.
- Dancy, C. L., & Ritter, F. E. (2017). A Standard Model of the Mind Needs a Body. *Common Model of Cognition Bulletin*, 1(2), 316–320.
- Dancy, C. L. (2022). Using a Cognitive Architecture to consider antiblackness in design and development of AI systems. In *proceedings of the 20th International Conference on Cognitive Modeling*, Toronto, Ontario, CA, 65-72.
- Kawakami, K., Amodio, D. M., & Hugenberg, K. (2017). Intergroup perception and cognition: An integrative framework for understanding the causes and consequences of social categorization. In J. M. Olson (Ed.), *Advances in experimental social psychology* (pp. 1–80). Elsevier Academic Press.
- Macrae, C. N., & Bodenhausen, G. V. (2000). Social cognition: thinking categorically about others. *Annual review of psychology*, 51, 93–120.
- Speer, R., Chin, J., & Havasi, C. (2017). ConceptNet 5.5: an open multilingual graph of general knowledge. In *proceedings of the Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California, USA, 4444-4451.

Comparing Classical and Quantum Probability Accounts of the Interference Effect in Decision Making

Christopher R. Fisher (christopher.fisher.27.ctr@us.af.mil)
Parallax Advanced Research
Beavercreek, OH 45324 USA

Lorraine Borghetti (lorraine.borghetti.1.ctr@us.af.mil)
Air Force Research Laboratory
Wright Patterson AFB, OH USA

Joseph W. Houpt (joseph.houpt@utsa.edu)
University of Texas at San Antonio
San Antonio, TX 78249 USA

Christopher Stevens (christopher.stevens.28@us.af.mil)
Air Force Research Laboratory
Wright Patterson AFB, OH 45433 USA

Leslie M. Blaha (leslie.blaha@us.af.mil)
Air Force Research Laboratory
Wright Patterson AFB, OH 45433 USA

Abstract

Prior research has found interference effects (IEs) in decision making. IEs violate classical probability theory (CPT), making them hard to model. Our primary research goals are to (1) determine whether a model called the probability theory + noise (PTN) model can produce IEs, and (2) compare the predictions of the PTN to an existing quantum probability-based model called the Belief-Action Entanglement (BAE) model that is able to model IEs. The PTN assumes that memory operates consistently with CPT, but noise in the retrieval process can produce violations of CPT. Using parameter space partitioning, we found that the PTN can produce IEs, and unconstrained versions of both models can produce all possible patterns. We also show the PTN (but not the BAE) predicts a relationship we term the conditional attack probability equality (CAEP) which is contradicted by previously reported data. Collectively, our results show that the PTN can produce interference effects, but the BAE is favored because it is not bound by the CAEP.

Keywords: Quantum Cognition; Interference Effect; Probability Theory; Decision Making

Introduction

An active debate concerns whether classical probability theory (CPT) or quantum probability theory (QPT) should serve as probabilistic foundation for models of cognition (e.g., Busemeyer, Potheos, Franco, & Trueblood, 2011). CPT is based on set theory and Kolmogorov axioms, whereas QPT is based on the logic of sub-spaces and a subset of the Kolmogorov axioms (Busemeyer et al., 2011). Research over the past 50 years has compiled a long list of violations of CPT in judgment and decision making, including the conjunction fallacy and order effects (see Busemeyer et al., 2011; Costello & Watts, 2014). Understandably, this has cast doubt on the utility of CPT for cognitive models and has led some researchers to propose QPT as an alternative to CPT. Proponents of QPT note that it provides a natural explanation of violations of CPT because its less restrictive axioms allow for the occurrence of order effects, interference effects, and the conjunction fallacy, among other possible CPT violations (Busemeyer et al., 2011).

Given the long list of violations of CPT, it would seem as though the debate could be easily resolved. However, resolving the debate has been challenging because many violations

can be explained by augmenting a CPT-based model with simple cognitive mechanisms. One prominent example is the probability theory + noise (PTN) model (Costello & Watts, 2014). According to the PTN, the organization of memory is consistent with the rules of CPT. However, violations of CPT stem from noise in the memory retrieval process, which has a systematic rather than random effect on judgments. Thus, the PTN predicts that judgments would conform to the rules of CPT if noise could be eliminated. Prior research has demonstrated that the PTN can account for a wide range of violations of CPT in joint (Costello & Watts, 2014) and conditional (Costello & Watts, 2016) probability judgments.

In this paper, we focus on a violation of CPT called an interference effect (IE). An IE occurs when a marginal choice distribution depends on the presence or absence of a preceding judgment (Wang & Busemeyer, 2016). IEs imply a violation of a law of CPT called the law of total probability (LOTP). Thus, IEs are incompatible with any model bound by the LOTP. According to the LOTP, when a judgment is made, the marginal choice probability can be divided into mutually exclusive and exhaustive partitions based on the possible judgment outcomes. These partitions sum to the original value, meaning the judgment should not alter the marginal choice distribution. Prior research has demonstrated that IEs occur in decision making. To date, a quantum model called the Belief-Action Entanglement (BAE) model has provided a superior account of IEs compared to a Markov model which is based on CPT (Wang & Busemeyer, 2016).

Our primary research question herein is whether the PTN can also account for IEs with its noisy memory retrieval process. In other words, is it necessary to use QPT to explain IEs? Or could IEs also be explained by a noisy memory system that otherwise satisfies CPT? To address this question, we will extend the PTN model to a sequential decision making paradigm previously used for studying IEs, and we will use a model analysis method called parameter space partitioning (Pitt, Kim, Navarro, & Myung, 2006) to compare the PTN and the BAE in terms of the qualitative IE patterns they can and cannot produce.

The remainder of the article is organized as follows. First, we begin by detailing the categorization-decision paradigm used to study IEs. Next, we introduce the PTN model and briefly describe the BAE. We then explore the predictions of the models using two methods: (1) parameter space partitioning to explore the qualitative patterns of IEs, and (2) a Monte Carlo simulation to measure variability in the magnitude of IEs across each model’s parameter space. We show that the PTN imposes a qualitative equality constraint on conditional attack probabilities which is violated in the data. We conclude by discussing limitations and future directions.

Categorization-Decision Paradigm

The categorization-decision paradigm (CDP) is commonly used to study how categorization interferes with subsequent decision making (Townsend, Silva, Spencer-Smith, & Wenger, 2000; Wang & Busemeyer, 2016). In this paradigm, participants are always asked to make a 2-alternative forced choice action decision; in some conditions, they are also asked to make an explicit categorization judgment of the stimuli prior to the action decision. On each trial in the Busemeyer and Wang implementation of the paradigm, participants are presented with a face in one of three instruction conditions: (1) a decision-only condition (*d*) in which participants decide to attack or withdraw, (2) a categorize-then-decide condition (*cd*) in which participants categorize the face as good or bad before deciding to attack or withdraw, and (3) an explicit category condition (*xd*) in which the true category is told to the participant before their attack/withdraw decision. IEs are measured by comparing the marginal attack probabilities in the *d* condition to those in the *xd* and *cd* conditions. Interference occurs when the comparisons are not equal.

Each face belongs to either the “good” category or the “bad” category. In the conditions where participants did not know to which category each face belonged, they could use facial features (e.g., width) to infer the most likely category, and by extension, whether a face was likely to be friendly or hostile. As a convention, we will use the terms *type-g* and *type-b* to refer to a facial feature associated with the good and bad categories, respectively. Half of faces were *type-g* and the other half were *type-b*. *Type-g* faces and *type-b* faces had a 60% chance of being in the good and bad categories, respectively. Participants received a reward on 70% of trials for attacking a face in the bad category, or withdrawing from a face in the good category (making correct action decisions). Similarly, participants were punished on 70% of trials for attacking a face in the good category or withdrawing from a face in the bad category (making erroneous action decisions). In this paradigm, it is assumed that people at least implicitly categorize the faces before they make their explicit action decisions; part of the categorization-decision paradigm is to ask participants to make their category decisions explicitly in the *cd* condition, to assess if making an explicit categorization interferes with the rates of action decisions.

Interference Effects

The LOTP requires the marginal probability of attacking (irrespective of category membership) to be equal across each condition. Using the *d* and *cd* conditions as an example, the LOTP can be stated formally as:

$$\Pr_d(a | t_q) = \Pr_{cd}(a | t_q, g)\Pr_{cd}(g | t_q) + \Pr_{cd}(a | t_q, b)\Pr_{cd}(b | t_q), \quad (1)$$

where *a* represents attack, *b* represents the bad category, *g* represents the good category, $t_q \in \{t_b, t_g\}$ represents a face type, and t_g and t_b represents *type-g* and *type-b* face types, respectively. Each probability statement is subscripted by its condition; for example, *cd* is the categorize-and-decide condition. An IE occurs when the equality above is violated. Table 1 shows a common IE pattern found in aggregated data known as the critical asymmetry. In the *xd* condition, the IEs for *type-b* and *type-g* are similar in magnitude but opposite in direction. However, in the *cd* condition, an IE only occurs for the *type-b* face, and is typically positive.

Table 1: IEs reported in Wang and Busemeyer (2016) Exp. 2, computed as the difference in marginal attack probabilities between the *d* condition and the comparison condition.

xd		cd	
type-b	type-g	type-b	type-g
0.03	-0.03	0.04	0.00

Probability Theory + Noise

As depicted in Figure 1, the memory retrieval process of the PTN can be illustrated as a processing tree. The core assumption of the PTN model is that memory operates in accordance with CPT, but violations of CPT arise from noise in the memory retrieval process (Costello & Watts, 2014). If noise could be eliminated, judgments and decisions would follow CPT. It is important to note that noise creates systematic rather than random departures from CPT because noise has a regressive effect on judgments. Thus, averaging noisy judgments does not necessarily produce results consistent with CPT. Previous research found the PTN is broadly consistent with the pattern of observed CPT violations for identities related to joint probabilities (Costello & Watts, 2014) and conditional probabilities (Costello & Watts, 2016), including the conjunction fallacy.

Judgments are made by estimating the relative frequency of an event encoded in memory (Costello & Watts, 2014). For example, the process of estimating the probability of event *A* involves the following four steps: (1) retrieve *n* instances from memory, (2) read the property flag of each memory, (3) count the number of instances flagged as event *A*, n_A , and (4) estimate the probability of *A* as $\frac{n_A}{n}$. The reading process is subject to random error, leading to violations of CPT in certain cases.

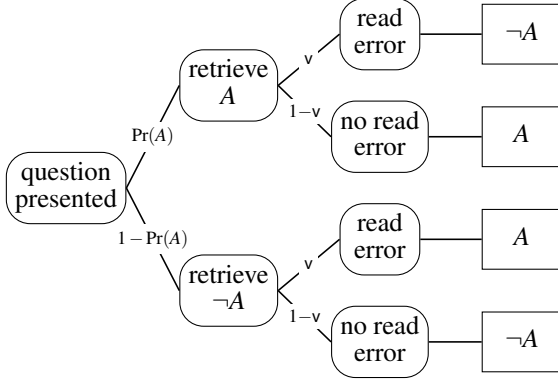


Figure 1: A process tree diagram of the PTN’s retrieval process for judging event A . $\Pr(A)$: probability of retrieving a memory for event A , v : probability of an error in reading the memory.

Figure 1 illustrates retrieval process for estimating the probability of event A as a processing tree. Starting at the root of the tree, the probability of retrieving a memory for event A is $\Pr(A)$. Similarly, the probability of retrieving a memory that is *not* for event A , denoted $\neg A$, is $1 - \Pr(A)$. After retrieving a memory, there is a chance $v \in [0, .5]$ that it is read incorrectly (e.g., A is misread as $\neg A$). The probability judgment for event A is found by multiplying the probabilities within both paths leading to A , and summing them together. Formally, this probability is given by:

$$J(A) = v\Pr(\neg A) + (1 - v)\Pr(A) = (1 - 2v)\Pr(A) + v \quad (2)$$

where $\Pr(A)$ is the true, subjective probability that a memory is for event A . To reiterate, $\Pr(A)$ refers to the probability of retrieving a memory for A , and v refers to the probability that the information is misread once it has been retrieved. Importantly, events described by $\Pr(\cdot)$ must conform to the rules of CPT, but events described by $J(\cdot)$ may violate CPT due to noise. If $v = 0$, then judgments and decision making obey CPT because $J(A) = \Pr(A)$.

A similar process is used for estimating the joint probability of multiple events occurring simultaneously. However, as noted in Costello and Watts (2016), subsequent versions of the PTN include an additional error term for complex events, such as conjunctions. The rationale is that complex events produce more errors in the reading process. We will denote the augmented error term as $\epsilon = v + \Delta$, where $\Delta \in [0, .10]$ is additional error for complex events, and $\epsilon \in [0, .50]$.

The judgment process for the conditional probability A given B involves a two-step reading process. Step one is reading whether a memory has a flag for event B . If the model reads the flag as B , the model increments a counter n_B for event B . Next, if the memory was read as B , the model reads whether a flag also indicates A , and if so, it increments $n_{A \wedge B}$, the counter for event $A \wedge B$. The conditional probability is estimated as $\frac{n_{A \wedge B}}{n_B}$.

Decision Process

One approach for extending the PTN to decision making in the CDP is to assume a response is based on the retrieval of a single memory (see also Borghetti, Fisher, Houpt, Blaha, & Gunzelmann, 2022). Importantly, Costello and Watts (2016) demonstrated that the expected value of the probability estimate is invariant to sample size. For example, consider a trial in the xd condition involving a *type-g* face in the good category. We assume that the decision is based on the two-step retrieval process described above for conditional probabilities. First, the model reads whether the memory has a flag matching the conditioning event *type-g* and good. Next, if the flag is read as *type-g* and good, it reads whether a flag also indicates “attack”. If so, the model decides to attack. Otherwise, it decides to withdraw.

Response Probabilities

Below, we outline the response probabilities each of the 12 CDP conditions which are based on the conditional probability equations presented in Costello and Watts (2016). We augment our notation to include the probability of retrieving instances from memory that contain information about multiple aspects of the stimulus by including more parameters in the $\Pr(\cdot)$ function. For example, $\Pr(a, t_g)$ is the probability that an instance of attack and a *type-g* face is drawn from memory.

Explicit Category Given In the xd condition, participants are given both the true category before deciding whether to attack or withdraw. Given face type $t_q \in \{t_b, t_g\}$ and category $z \in \{b, g\}$ the probability of attacking is:

$$J_{xd}(a | t_q, z) = \frac{[1 - 2\epsilon]^2 \Pr(a, t_q, z) + \epsilon [1 - 2\epsilon] [\Pr(a) + \Pr(t_q, z)] + \epsilon^2}{[1 - 2\epsilon] \Pr(t_q, z) + \epsilon} \quad (3)$$

We use the extended error rate term ϵ because the conditioning event is a conjunction. Note that the four conditional attack probabilities in xd can be formed by substituting each permutation face type and category into t_q and z .

Categorize and Decide Unlike the xd condition, the cd condition is divided into a categorization stage followed by a decision stage. Given a facial feature t_q , the probability of categorizing a face as z is:

$$J_{cd}(z | t_q) = \frac{[1 - 2v]^2 \Pr(z, t_q) + v [1 - 2v] [\Pr(z) + \Pr(t_q)] + v^2}{[1 - 2v] \Pr(t_q) + v}$$

After categorizing the face, a decision to attack or withdraw is made. The probability of attacking a face with feature t_q in category z is the same as defined in Eq. 3 for the xd condition:

$$J_{cd}(a | t_q, z) = J_{xd}(a | t_q, z) \quad (4)$$

The conditional attack probabilities in Eq. 4 are equal because PTN does not make a distinction between a judged category vs. a true category. Thus, all that matters at the time of the decision is the face type and the category values.

Decision Only

In the d condition, participants are presented with a face and decide whether to attack or withdraw. Participants are neither provided with the true category (as in xd), nor are they instructed to categorize the faces (like cd). The probability of attacking a face with the t_q feature is given by:

$$J_d(a | t_q) = \frac{[1 - 2v]^2 \Pr(a, t_q) + v [1 - 2v] [\Pr(a) + \Pr(t_q)] + v^2}{[1 - 2v] \Pr(t_q) + v}.$$

Table 2: Joint probability distribution for the probability matching PTN model paneled by face type.

	type-b		sum
	bad	good	
attack	0.21	0.06	0.27
withdraw	0.09	0.14	0.23
sum	0.3	0.2	0.50

	type-g		sum
	bad	good	
attack	0.14	0.09	0.23
withdraw	0.06	0.21	0.27
sum	0.2	0.3	0.50

The BAE Quantum Probability Model

In contrast to CPT models which must adhere to the axioms of set theory, the BAE model follows the logic of sub-spaces (Busemeyer et al., 2011). Accordingly, event probabilities are computed within a geometric Hilbert space across a field of complex numbers such that an inner-product indexes belief strength, and beliefs are described within a four-dimensional space spanned by an orthonormal basis. At any moment within this multidimensional structure, the cognitive system is in a superposition state—an uncertain, conflicted state with respect to the decision—that evolves throughout deliberation.

The initial state is superposed over four possible basis vectors, $\Psi_{t_q} = [\Psi_{GW}, \Psi_{GA}, \Psi_{BW}, \Psi_{BA}]^T$, where B represents the bad category and A represents the attack action. When presented with facial feature t_q , Ψ_{t_q} is modeled as a linear combination of the basis vectors. The parameter j governs the probability a *type-b* or *type-g* face will be judged as belonging to the bad or good category, respectively. For a *type-b* feature, we have:

$$\Psi_b = \frac{1}{\sqrt{2}} [\sqrt{1-j}, \sqrt{1-j}, \sqrt{j}, \sqrt{j}]^T.$$

In the xd and cd conditions, probabilities are updated when the category is provided by the experimenter or self-reported by the participant, respectively. For example, after a facial feature t_q is categorized as bad, the state is updated to

$$\Psi_{t_q} \rightarrow \Psi_b = \frac{1}{\sqrt{2}} [0, 0, 1, 1]^T.$$

The state does not update in the d condition as the category remains unknown.

Decisions are driven by utility values associated with facial feature and category, as well as the experimental reward rate. For example, the utility for a *type-b* feature categorized as bad, $\mu_{tb,b}$, increases the attack probability, whereas the utility for good categorization $\mu_{tg,g}$ decreases the attack probability. Similarly, the utility for a *type-g* feature categorized as bad, $\mu_{tg,b}$, increases the probability of attack in contrast to a good categorization $\mu_{tg,g}$. Further, utilities are influenced by the γ parameter which coordinates beliefs about categories and actions when the category remains unknown in the d and cd conditions but not when known in the xd condition. Interactions between utility and γ parameter allow the BAE model to produce the critical asymmetry in the cd condition.

Conditional Attack Equality Property

As shown in Eq. 4, the PTN predicts that the conditional attack probabilities in the xd and cd conditions must be equal given the same feature t_q and category z . We term this equality constraint the *conditional attack equality property* (CAEP). Importantly, the CAEP holds regardless of the parameter values assigned to the PTN, meaning a core assumption the PTN must be changed to predict otherwise. Wang and Busemeyer (2016) noted that a Markov decision model based on CPT also predicts the CAEP and found evidence that it was violated in their data. In contrast to the PTN, the BAE is not constrained by the CAEP.

Parameter Space Partitioning

Parameter space partitioning (PSP) is a method for identifying regions of a model’s parameter space associated with qualitative patterns (Pitt et al., 2006). PSP is useful for exploring the behavior of a model and assessing its flexibility. As noted in Roberts and Pashler (2000), a good fit not impressive if a model can produce any pattern of data. PSP is particularly useful for identifying potential critical tests in which one model predicts a pattern that is not predicted by another.

In the CDP, there are three qualitative patterns of IEs: negative, positive, and approximately equal. We consider an IE to be approximately equal if it is less than $|\cdot 005|$. In total, there are $3^4 = 81$ possible IE patterns because four conditions are formed by crossing face type and category.

As listed in Table 3, we developed a hierarchy of BAE and PTN models based on various parameter constraints. By constraining a parameter, it is possible to understand its role in producing IEs. An index of 1 indicates the presence of a constraint whereas an index of 0 indicates the absence of a constraint. For the BAE, we considered an unconstrained sub-model BAE₀ and a constrained sub-model BAE₁ in which $\mu_{tg,b} = -\mu_{tg,g}$ as described the original paper (Wang & Busemeyer, 2016). In the PSP analysis, we searched for patterns within the following parameter ranges except where constraints applied: $j \in [0, 1]$ and $\mu_{tb,b}, \mu_{tb,g}, \mu_{tg,g}, \mu_{tg,b}, \gamma \in [-2, 2]$.

For the PTN, we developed a hierarchy of eight models based on whether or not the following constraints apply: (1)

$v = 0$, (2) $\Delta = 0$, and (3) the true subjective probabilities are based on probability matching (i.e., the objective stimulus probabilities). We use three indices to indicate which of the three constraints apply. For example, PTN_{010} indicates a sub-model in which only the second constraint applies. Table 2 lists the parameters used for probability matching. In the PSP analysis, we searched for patterns within the following parameter ranges true subjective probabilities: $p_i \geq 0$, such that $\sum_{i=1}^n p_i = 1$. The ranges for the other parameters were $\epsilon = v + \Delta \in [0, .50]$, $v \in [0, .50]$, and $\Delta \in [0, .10]$.

Empirical Data

We used data from Experiment 2 in Wang and Busemeyer (2016) to test the CAEP. A total of 286 participants completed the d , cd and xd conditions in the categorize-decide paradigm. On 70% of the trials, participants received positive feedback for attacking a face in the bad category or withdrawing from a face in the good category. On 30% of the trials, participants received positive feedback for attacking a face from the good category or withdrawing from a face from the bad category. We excluded data from five participants whose z-score for missing trials was 2 or more standard deviations above the mean. After excluding data from these participants, 0.3% of trials were missing. Full details can be found in Wang and Busemeyer (2016).

Results

Parameter Space Partitioning Results

Table 3 lists the number of IE patterns each model can produce and whether a model can produce the critical asymmetry (see Table 1)¹. As expected, the BAE can predict the critical asymmetry. When $\mu_{t_g,b} = -\mu_{t_g,g}$, it cannot produce IEs for *type-g* faces in the cd condition. When this constraint is relaxed, the BAE predicts all 81 patterns. Importantly, the PTN can produce IEs, including the critical asymmetry. To predict the critical asymmetry, at least one noise parameter and the true probabilities must be free to vary. Comparison of the PTN sub-models shows that the noise parameters have a small effect on interference patterns by themselves, but they are necessary for producing the critical asymmetry.

Interference Effect Distributions

We generated IE distributions across the allowable parameter space of each model to better understand variance in the magnitude of the IEs. The parameter ranges used here are the same as those used in the PSP analysis. Table 4 provides the mean and standard deviations of the IE distributions for each sub-model. Two general patterns emerged for the BAE and PTN: (1) the mean IEs are close to zero, (2) the standard deviations are typically larger in the xd condition than the cd condition. One difference between the two models is that the BAE tends to produce IE distributions with a larger variance

Table 3: PSP results

Model	Constraints	n	Critical Asymmetry
PTN_{000}	none	81	yes
PTN_{001}	pm	3	no
PTN_{010}	$\Delta = 0$	81	yes
PTN_{011}	$\Delta = 0$, pm	2	no
PTN_{100}	$v = 0$	81	yes
PTN_{101}	$v = 0$, pm	2	no
PTN_{110}	$v = 0$, $\Delta = 0$	9	no
PTN_{111}	$v = 0$, $\Delta = 0$, pm	1	no
BAE_0	none	81	yes
BAE_1	$\mu_{t_g,b} = -\mu_{t_g,g}$	27	yes

pm: probability matching; n : number of IE patterns found with PSP.

compared to the PTN, suggesting that the BAE is more flexible in its predictions. An alternative way to assess the flexibility of the models is to examine the inter-correlations between IEs. Due to space limitations, we will focus on the unconstrained version of each model. The mean absolute correlation for the PTN_{000} was .56 ($sd = .25$) and .14 ($sd = .20$) for the BAE_0 , indicating that the BAE tends to produce a wider range of IEs.

Table 4: Mean (standard deviation) of IE distributions

Model	xd, t_b	xd, t_g	cd, t_b	cd, t_g
PTN_{000}	.000 (.046)	.000 (.045)	.000 (.036)	.000 (.035)
PTN_{001}	.003 (.003)	-.003 (.003)	.006 (.003)	-.006 (.003)
PTN_{010}	.000 (.033)	.000 (.034)	.000 (.012)	.000 (.012)
PTN_{011}	-.001 (.001)	.001 (.001)	.003 (.002)	-.003 (.002)
PTN_{100}	.001 (.093)	.000 (.092)	.001 (.053)	.000 (.052)
PTN_{101}	.005 (.003)	-.005 (.003)	.005 (.003)	-.005 (.003)
PTN_{110}	.002 (.101)	-.001 (.099)	.000 (.000)	.000 (.000)
PTN_{111}	.000 (.000)	.000 (.000)	.000 (.000)	.000 (.000)
BAE_0	.007 (.230)	-.002 (.229)	.000 (.094)	-.001 (.093)
BAE_1	-.003 (.228)	.002 (.152)	.001 (.093)	.000 (.000)

t_g : *type-g*, t_b : *type-b*, xd : explicit category condition, cd : categorize-and-decide condition

CAEP

We used a Bayesian hierarchical latent trait model (Klauer, 2010) to test the CAEP as predicted by the PTN (see Eq. 4). Briefly, the hierarchical latent trait model represents parameters in real space as a multivariate normal distribution which are mapped to a probability scale via a probit transformation to predict conditional attack probabilities for each condition. We compare the conditions by taking their difference: $\theta_{diff} = \theta_{xd} - \theta_{cd}$. The CAEP is tested by comparing the group-level posterior distribution of θ_{diff} , to the value of 0 predicted by the CAEP. The CAEP is considered to be contradicted if predicted value of 0 is outside the bulk of a posterior distribution, defined by the 95% highest density interval.

Figures 3 and 2 display the group-level posterior distributions for the difference in predicted conditional attack probabilities between xd and cd . Contrary to the PTN, the poste-

¹We used <https://github.com/itsdfish/ParameterSpacePartitions.jl> for the PSP analysis

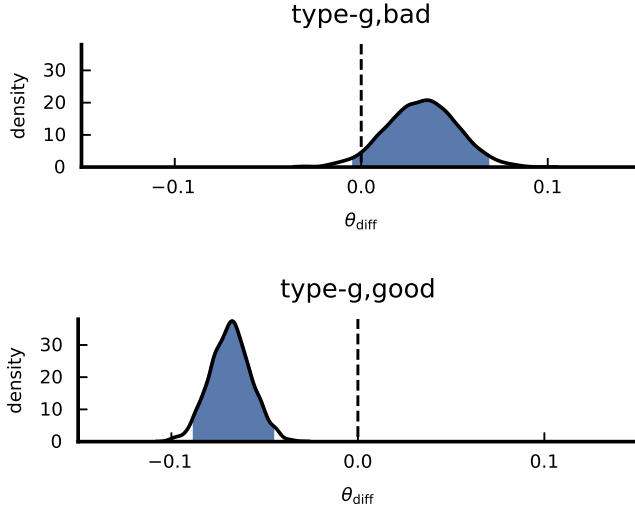


Figure 2: Posterior distributions of the difference in predicted conditional attack probabilities $\theta_{\text{diff}} = \theta_{\text{xd}} - \theta_{\text{cd}}$ for *type-g* faces. The vertical, dashed black line represents the CAEP prediction derived from the PTN model. Shaded area represents the 95% highest density interval.

rior distributions for *type-b* faces in Figure 3 are positive and shifted away from zero. In Figure 2, the highest density interval overlaps with the predicted value of zero for *type-g* faces in the bad category. However, for *type-g* faces in the good category, the posterior distribution is negative, thus contradicting the PTN.

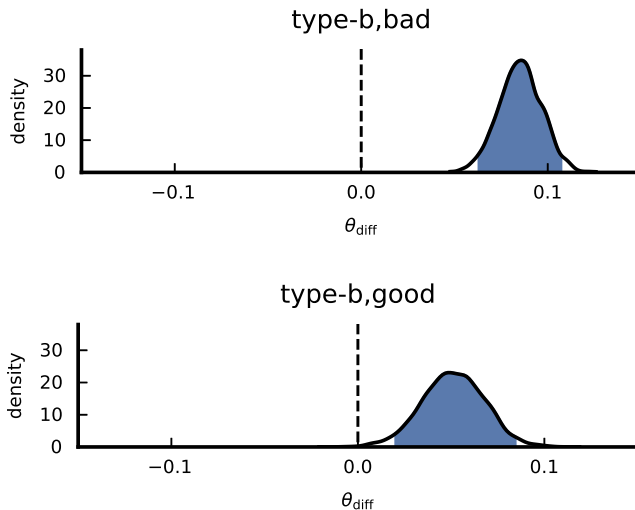


Figure 3: Posterior distributions of the difference in predicted conditional attack probabilities $\theta_{\text{diff}} = \theta_{\text{xd}} - \theta_{\text{cd}}$ for *type-b* faces. The vertical, dashed black line represents the CAEP prediction derived from the PTN model. Shaded area represents the 95% highest density interval.

Discussion

Our goal was to compare two competing accounts of IEs which differed in their underlying probabilistic foundations.

The PTN is based on CPT whereas the BAE is based on QPT. We found that the PTN is able to produce IEs despite being based on CPT. In the PTN, two conditions are required to produce the critical asymmetry: the true probabilities must depart from the objective probabilities, and there must be some degree of noise in the memory retrieval process. We found that both models can produce all possible IE patterns, but the BAE can produce the critical asymmetry with fewer qualitative patterns compared to the PTN. Another difference between the models is that the BAE produces IE distributions with a larger variance than the PTN.

Conditional Attack Equality Property

Perhaps the most striking difference between the PTN and the BAE was the CAEP rather than IEs. In contrast to the BAE, the PTN is constrained by the CAEP, which requires conditional attack probabilities to be equal in the *xd* and *cd* conditions when given the same stimulus. This prediction was not supported by the data. Given that this prediction holds for the PTN regardless of parameter values, an interesting question is whether alternative versions of the PTN are bound by the CAEP. If the degree of noise depends on how the category information is obtained (judged vs. given) and category, the CAEP would not hold. However, there are at least two potential challenges: (1) explaining how condition and category affect memory processes, and (2) ensuring this change does not eliminate the PTN’s ability to produce IEs.

Comparison to other models

Recently, Borghetti et al. (2022) proposed an ACT-R model and Fisher, Borghetti, Houpt, Blaha, and Stevens (2022) proposed the Judgment Revision Model (JRM) which can produce IEs. Somewhat similar to the PTN, IEs in the ACT-R model emerge from errors during memory retrieval. Variants of the ACT-R model without learning are also constrained by the CAEP. One challenge for this model would be to show that learning depends on condition, face type, and category in a way that matches the empirical data. The JRM is a multinomial processing tree which assumes IEs emerge from a category revision process. Unlike the PTN, the JRM is not constrained by the CAEP.

Conclusion

Our model comparison underscores the difficulty of distinguishing between models based on CPT and QPT. Although the PTN is based on CPT, it can produce IEs through errors in the memory retrieval process. Thus, a simple failure to produce interference effects cannot distinguish between the BAE and PTN. It turns out the CAEP—which is orthogonal to the underlying probability theory—is a distinguishing factor, and supports the BAE over the PTN.

Acknowledgments

This research was supported by Air Force Office of Scientific Research grant 21RHCOR080. The opinions expressed

herein are solely those of the authors and do not necessarily represent the opinions of the United States Government, the U.S. Department of Defense, the Department of the Air Force, or any of their subsidiaries or employees. Distribution A Approved for public release; distribution unlimited (AFRL-2023-2724).

References

- Borghetti, L., Fisher, C. R., Hout, J. W., Blaha, L. M., & Gunzelmann, G. (2022). Towards a method for evaluating convergence across modeling frameworks. In *Proceedings of the 20th international conference on cognitive modeling*.
- Bussemeyer, J. R., Pothos, E. M., Franco, R., & Trueblood, J. S. (2011). A quantum theoretical explanation for probability judgment errors. *Psychological Review*, 118(2), 193–218.
- Costello, F., & Watts, P. (2014). Surprisingly rational: Probability theory plus noise explains biases in judgment. *Psychological Review*, 121(3), 463.
- Costello, F., & Watts, P. (2016). People's conditional probability judgments follow probability theory (plus noise). *Cognitive Psychology*, 89, 106–133.
- Fisher, C. R., Borghetti, L., Hout, J. W., Blaha, L. M., & Stevens, C. (2022). A comparison of quantum and multinomial processing tree models of the interference effect. In *Proceedings of the 20th international conference on cognitive modeling*.
- Klauer, K. C. (2010). Hierarchical multinomial processing tree models: A latent-trait approach. *Psychometrika*, 75(1), 70–98.
- Pitt, M. A., Kim, W., Navarro, D. J., & Myung, J. I. (2006). Global model analysis by parameter space partitioning. *Psychological Review*, 113(1), 57–83.
- Roberts, S., & Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. *Psychological Review*, 107(2), 358–367.
- Townsend, J. T., Silva, K. M., Spencer-Smith, J., & Wenger, M. J. (2000). Exploring the relations between categorization and decision making with regard to realistic face stimuli. *Pragmatics & Cognition*, 8(1), 83–105.
- Wang, Z., & Bussemeyer, J. R. (2016). Interference effects of categorization on decision making. *Cognition*, 150, 133–149.

Using Neural Networks to Create Fast and Reusable Approximate Likelihood Functions for ACT-R

Christopher R. Fisher (christopher.fisher.27.ctr@us.af.mil)

Parallax Advanced Research
Beavercreek, OH 45324 USA

Taylor Curley(taylor.curley@us.af.mil) Christopher Stevens(christopher.stevens.28@us.af.mil)

Air Force Research Laboratory
Wright Patterson AFB, OH USA

Abstract

Likelihood functions form the basis for statistical inference techniques, including maximum likelihood estimation, and Bayesian estimation/model comparison. Unfortunately, deriving likelihood functions analytically for cognitive architectures such as ACT-R can be challenging, if not impossible in some cases, often requiring considerable time and expertise. Simulation-based approximations are computationally intensive, making them impractical to implement in real-time applications. We demonstrate how recently-developed techniques for learning intractable likelihood functions with neural networks can be applied to a visual search model based on ACT-R, and reused once trained. Our work extends prior applications in two ways: (1) we demonstrate that the technique can be scaled to a large number of conditions based on the size of the visual search array, and (2) we demonstrate that the technique is applicable to both unimodal and multimodal versions of the model. We conclude with a discussion for scaling up neural network techniques for approximating likelihood functions.

Keywords: ACT-R; neural networks; likelihood-functions; parameter-estimation

Introduction

One of the benefits of cognitive architectures, such as Adaptive Control of Thought -Rational (ACT-R; Anderson et al., 2004), is the ability to develop cognitively plausible models which scale to a wide range of complex tasks (Newell, 1990). An unfortunate trade-off is the difficulty of deriving likelihood functions due to their complex statistical structure. A likelihood function connects a model to data via probability distributions and describes how likely data are given a model with specific parameter values. Likelihood functions are important because they form the basis for parameter estimation and model comparison for both frequentist and Bayesian approaches (Kruschke, 2014). A practical benefit is that likelihood functions are typically orders of magnitude faster than Monte Carlo simulation approaches (e.g., Fisher et al., 2022), making the models easier to evaluate.

Recently, researchers have developed various techniques to approximate intractable likelihood functions using neural networks (Papamakarios et al., 2019; Fengler et al., 2021; Boelts et al., 2022). The basic idea is that a neural network can be given simulated data from a cognitive model and learn the mapping between the parameters and the likelihood function. We will focus on one such technique called likelihood approximation networks (LANs; Fengler et al., 2021). LANs learn the relationship between model inputs (e.g., parameters, stimulus values) and the likelihood function, which is

approximated from a distribution of simulated data using a kernel density estimator. Although generating training data and training the LAN can be computationally intensive, this is a one-time, upfront cost. Once the LAN is trained, evaluating the likelihood function is extremely fast, consisting of a simple forward pass through the network. Using LANs greatly speeds up the parameter estimation process because (1) evaluating the likelihood function is the primary bottleneck, and (2) the likelihood function must be evaluated hundreds or thousands of times during parameter estimation. In addition, the trained LAN can be saved, shared, and reused to quickly perform maximum likelihood or Bayesian parameter estimation.

Prior research has demonstrated the feasibility and utility of using LANs to approximate the likelihood function of various evidence accumulation models (Fengler et al., 2021; Boelts et al., 2022), such as the drift diffusion model (Ratcliff, 1978). Although LANs show promise as a proof-of-concept, little is currently known about the performance of LANs with respect to other types of models, such as cognitive architectures, which may have different model structures and likelihood typologies.

Our goal is to demonstrate how LANs can be used to approximate likelihood functions for the ACT-R cognitive architecture using a visual search model (VSM) as a test case. The likelihood function for the VSM is challenging to derive because it describes a complex mixture of visual fixations which becomes increasingly complex as the set of visual object grows. Our application of LANs extends previous work in two ways: (1) we demonstrate that LANs work for a large range of experimental conditions, and (2) we apply LANs to unimodal and multimodal versions of the VSM to showcase its flexibility.

Overview

The remainder of the paper is structured as follows. First, we will review techniques for developing and approximating likelihood functions, and describe their relative trade-offs. Next, we will describe a conjunctive visual search task followed by two VSMs based on ACT-R. We then apply LANs to these models and perform benchmarks for execution time and parameter recovery. Finally, we discuss future directions and challenges for using LANs with more complex models.

Approaches for Likelihood Functions

In this section, we review several approaches for developing and using likelihood functions: analytic derivations, probability density approximations (PDAs), lookup tables (LTs) and LANs. Each approach is characterized by different trade-offs. Before proceeding, we will formally introduce the likelihood function. A likelihood function for a vector of independent observations $\mathbf{Y} = \{y_1, y_2, \dots, y_n\}$ given a vector of parameters $\Theta = \{\theta_1, \theta_2, \dots, \theta_k\}$ can be written in terms of the probability density function (PDF), f :

$$\mathcal{L}(\Theta; \mathbf{Y}) = \prod_i^n f(y_i | \Theta). \quad (1)$$

To provide some intuition, Figure 1 shows the relationship between the top-down weight parameter ω_{td} (explained below) and the likelihood of data generated with $\omega_{td} = .50$. As one might expect, the likelihood is maximized near $\omega_{td} = .50$, and decreases rapidly as the value of ω_{td} moves away from .50.

Perhaps the most common method for developing a likelihood function is the analytic approach whereby the likelihood function is derived through a series of mathematical operations and theorems to produce a closed-form equation (e.g., Fisher et al., 2022). Typically, analytic likelihood functions can be evaluated quickly and impose few hardware demands. Unlike other methods discussed here, another benefit of the analytic approach is that it may provide mathematical insights about properties of the model and its relationship to other models. The primary drawbacks are (1) the time investment, (2) the required mathematical expertise, and (3) potential challenges scaling to some complex models.

In light of challenges with analytic approach, methods such as PDA have been developed to approximate likelihood functions through Monte Carlo simulation of the generative model (Turner & Sederberg, 2014). Using PDA entails the following three steps: (1) simulate the model thousands of times, (2) approximate the likelihood function with a kernel density estimator (KDE), and (3) evaluate the likelihood of the data with the KDE. One benefit of PDA is that it can scale to any model and the initial costs are low. Perhaps the most significant drawback of PDA is that it can be computationally intensive at run-time, which makes model evaluation slow and prohibits real-time model evaluation in practical applications. In addition, predictions generated with PDA are discarded rather than reused in future applications.

An alternative approach involves precomputing the predictions of the model and storing them in a lookup table (LT) for later use where they can be compared to observed data (Fisher et al., 2016). Although generating the LT requires a large initial computational cost, the benefits include ability to save and reuse the results, and the ability to quickly evaluate the fit of the model. One problem with LTs is the trade-off between accuracy and RAM usage. As the parameter space increases, more samples are needed to maintain the same level of accuracy, which eventually imposes high demands on RAM. This

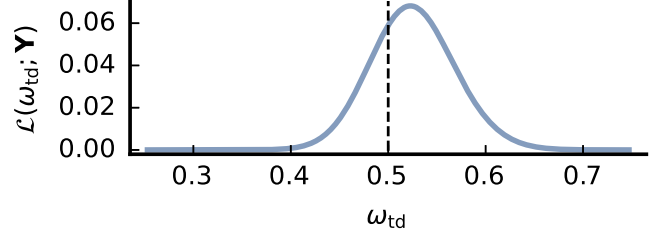


Figure 1: The likelihood of the data as a function of parameter ω_{td} . The data generating value of $\omega_{td} = .50$ represented by dashed vertical line.

trade-off stems from the fact that the input and outputs are stored rather than the function linking the two together.

As mentioned previously, LANs approximate the likelihood function of a model by learning the mapping between the inputs and output of the function. LANs involve three steps: (1) generate training data from the model using PDA, (2) train a neural-network on the relationship between model inputs and the log-likelihood, (3) evaluate the log-likelihood with a forward pass of the trained neural network. Much like the LT approach, LANs can be reused and are very fast at run-time. Given that LANs learn the relationship between model inputs and log likelihood, they have two significant advantages over LTs: (1) the trained neural network has a small RAM footprint, and (2) neural networks can interpolate the log likelihood of parameters values not used in the training data. One obvious drawback is the initial computational cost, which involves training the neural network in addition to generating a large set of training data from the model.

Conjunctive Visual Search Task

In a conjunctive visual search task (CVST), subjects must locate a stimulus that matches a target on multiple dimensions (Treisman & Gelade, 1980). The target stimulus is embedded within an array of scattered distractors without overlap. The set-size effect is a well established finding in which reaction time (RT) increases with the number of visual objects in the array (Treisman & Gelade, 1980). In our CVST, visual objects vary along two dimensions: color (black vs. grey), and shape (p vs. q). A target stimulus must match the target on both dimensions (e.g., black q), whereas distractors match on only one dimension (e.g., black p, or grey q). Visual objects of length $.86^\circ$ were placed randomly within a $11.33^\circ \times 11.33^\circ$ visual array, such that no overlapping occurred. Targets were present in 50% of trials and absent in the remaining trials.

Visual Search Model

We developed two visual search models (VSMs) based on ACT-R (Anderson et al., 2004) and an extension to its visual system called Pre-Attentive Attentive Vision (PAAV; Nyamsuren & Taatgen, 2013). One variant of the VSM produces a unimodal RT distribution and the other variant produces a multimodal RT distribution. As illustrated in Figure 2, RTs in

the VSM arise from a mixture of a different number of visual fixations. We leverage this fact to create unimodal and multimodal versions of the VSMs by manipulating the variance of the processing time of low level cognitive processes (e.g., conflict resolution, saccade, etc.). Although RT distributions are not typically multimodal in CVSTs (Palmer et al., 2011), we include a multimodal VSM to showcase the flexibility of LANs.

Visual Objects and Chunks

In ACT-R, visual objects, \mathbf{v}_j and chunks, \mathbf{c}_j , are represented as a set of feature-value pairs. For example, a visual object is represented as $\mathbf{v}_j = \{(f_{j,i}, b_{j,i})\}_{i \in I_j}$, where $f_{j,i}$ and $b_{j,i}$ are the feature and value of pair i , and I_j is the index set for slot-value pairs of visual object j . As a specific example, the target could be represented as $\mathbf{v}_t = \{(\text{color}, \text{black}), (\text{shape}, \text{q})\}$. We will use the set $Q_j = \{f_{j,i}\}_{i \in I_j}$, to denote a set of features (e.g. domain) in \mathbf{v}_j (or \mathbf{c}_j for a chunk). The mapping from features to values is defined as $v_i(f) = b_i$.

Visual Activation

In PAAV, visual activation is a weighted sum of three components: (1) top-down activation, (2) bottom-up activation, and (3) activation noise. Activation for visual object i is given by:

$$a_i = \omega_{td}ta_i + \omega_{bu}ba_i + \epsilon_i, \quad (2)$$

where ta_i and ba_i are top-down and bottom-up activation, ω_{td} and ω_{bu} are top-down and bottom-up weights, and $\epsilon_i \sim \text{normal}(0, \sigma)$ is activation noise. We fix $\omega_{bu} = 1.1$ and $\sigma = \frac{s\pi}{\sqrt{3}}$, with $s = .2$ as the default value.

Top-down activation reflects the accentuation of features based on the goal to find the target, which is encoded as a chunk \mathbf{c}_t in the goal buffer at the beginning of the trial. A visual object matching the target will have more top-down activation than one that does not match. Formally, top-down activation is given by:

$$ta_i = \sum_{k=1}^{n_f} \text{sim}(v_i(f_k), c_t(f_k)), \quad (3)$$

where n_f is the number of features, and sim is a binary similarity function, which returns 1 if $v_i(f_k) = c_t(f_k)$ and 0 otherwise.

Bottom-up activation is based on the contrast between a visual object and those surrounding it, such that the effect of contrast increases with decreasing distance. Formally, bottom-up activation is defined as:

$$ba_i = \sum_{j=1}^{n_v} \sum_{k=1}^{n_f} \frac{\text{dissim}(v_i(f_k), v_j(f_k))}{\sqrt{d_{i,j}}}, \quad (4)$$

where n_v is the number of visual objects, $d_{i,j}$ is the distance between visual objects i and j , and dissim is a binary dissimilarity function which returns 1 if $v_i(f_k) \neq v_j(f_k)$ and 0 otherwise.

Inhibition of Return

ACT-R briefly tracks a small number of visual objects to prevent multiple fixations of the same visual objects in quick succession. By default, the maximum number of objects (i.e., “finsts”) is 4 and the maximum duration is 3 seconds. If more than 4 visual objects are encountered within the maximum duration, the oldest visual objects are discarded in favor of more recently fixated visual objects.

Processing Times

In our VSM, we assume that the processing times of elementary cognitive processes, such as conflict resolution and motor execution, follow a gamma distribution with a standard deviation that depends on the mean. We reparameterized the gamma distribution as follows: $\text{gamma}(\mu_p, \sigma_p)$ where μ_p is the mean processing time of process p , and $\sigma_p = w\mu_p$ is the standard deviation of processing time of process p . The parameter w determines how large the standard deviation is relative to the mean. In all models, we used default values for each μ_p . In the multimodal model, we set $w = .20$ so that the component distributions are distinguishable. In the unimodal model, we increase w to $.50$ so that the component distributions sufficiently overlap. Under the assumption that processing times are serial and independent, the observed RT is the sum of individual processing times.

Response Rules

Departing from PAAV, our model uses a dynamic termination threshold based on Moran et al. (2013) to decide whether to fixate on the most active visual object or to terminate the search. The dynamic threshold is defined as:

$$\tau_r = \mu_r + \epsilon_r, \quad (5)$$

where μ_r is the expected threshold value after encountering r distractors, and $\mu_0 = 0$. After encountering a distractor, the threshold is updated according to:

$$\mu_r = \mu_{r-1} + \Delta, \quad (6)$$

where Δ is the updating factor. Thus, the probability of terminating and responding “absent” increases with each failed attempt to find the target. Given \mathbf{v}_m —the visual object with the highest activation—and \mathbf{c}_t —a chunk in the goal buffer representing the target—the response rules can be divided into three cases:

Case 1: $a_m \geq \tau_r$ and $\forall_k v_m(f_k) = c_t(f_k)$

All features match and the model responds “present”.

Case 2: $a_m \geq \tau_r$ and $\exists_k v_m(f_k) \neq c_t(f_k)$

The model updates the threshold according to Equation 6 after encountering a distractor and continues searching.

Case 3: $a_m < \tau_r$

The model responds “absent” and ends the search.

Challenges with Likelihood Function

Several characteristics of the VSM present challenges for deriving the likelihood function. Perhaps the most significant challenge is marginalizing over the large set of component distributions which comprise the observed RT distribution. As shown in Figure 2, the VSM can be characterized as a mixture of multiple unobserved component distributions corresponding to elementary cognitive processes (e.g., conflict resolution, saccades etc.). A general expression for the PDF of a mixture model with one mixing variable is given by:

$$f(x) = \sum_{i=1}^n p_i f_i(x) \quad (7)$$

where $f(x)$ is the PDF of the mixture model, n is the number of components, p_i is the probability of the i^{th} component, and $f_i(x)$ is the PDF of the i^{th} component. The biggest challenge with deriving a likelihood function is the large number of component distributions, n , which is large due to three unknown quantities: (1) the number of fixations, (2) which visual objects were fixated, and (3) the order in which visual objects were fixated. As the number of visual objects increases, the n increases exponentially, eventually making the PDF infeasible to compute.

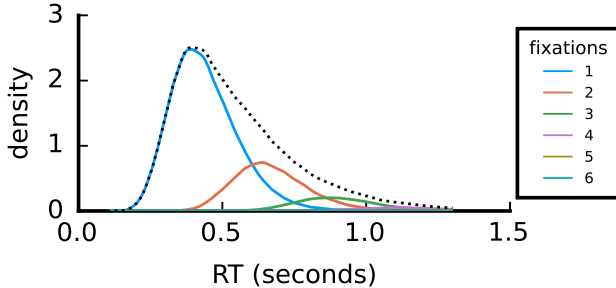


Figure 2: Densities of component distributions as a function of number of fixations. Black dotted line is the probability density for the observed RT distribution. Densities are based on a “present” response on a target present trial with a distractor set size of 6.

Simulation Methods

Likelihood Approximation Networks

As illustrated in Figure 3, we trained a LAN to learn the mapping between inputs (e.g., parameters, rt/choice, conditions) and the corresponding log likelihood. The LAN is a multi-layer perceptron consisting of an input layer for model inputs (e.g., parameters, stimulus values, data), three hidden layers of size [100, 100, 120], and an output layer that maps to a single node for the log likelihood (e.g., Fengler et al., 2021).

Training input vectors consisted of three types of information contributing to the log likelihood: (1) model parameters ω_{td} and Δ , (2) experiment parameters p —a target indicator variable—and n_d —the number of distractors in the array, and

(3) simulated data consisting of a response and RT. The label used for feedback was the log likelihood associated with the training input vector.

We generated training data for the model as follows. First, we sampled 15k parameter vectors from the following distributions: $\omega_{td} \sim \text{uniform}(0, 2)$, $\Delta \sim \text{uniform}(.25, .75)$, $p \sim \text{Bernoulli}(.5)$ and $n_d \sim \text{uniform}([2, 4, \dots, 20])$. For each parameter vector, we approximated the likelihood function by estimating the kernel density function from 50k simulated data points. Next, to create training data for each parameter vector, we generated 300 samples of data, and evaluated the log likelihood of each sample, resulting in $15k \times 300 = 4.5M$ training vectors. The test data were sampled using the same procedure, except we used 1,000 parameter vectors instead.

We trained the neural networks over the course of 50 epochs with a batch size of 1k, using the ADAM optimizer with a learning rate of .001.

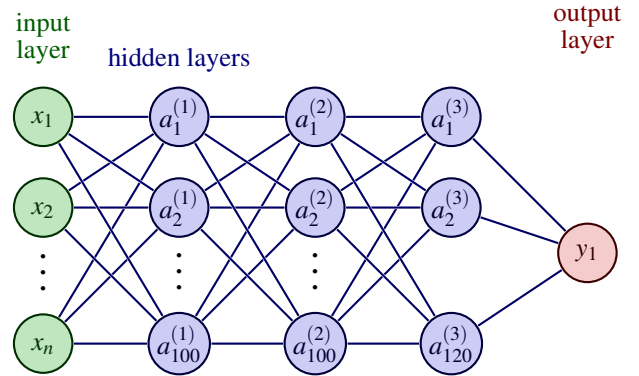


Figure 3: An illustration of the LAN for learning ACT-R likelihood functions. The input layer consists of parameters and data. The output layer emits the predicted log likelihood.

Parameter Recovery

We performed a parameter recovery simulation to assess the ability of the LAN to accurately estimate model parameters. A parameter recovery simulation involves three steps: (1) sample a vector of parameters θ_i from a distribution to serve as the ground truth, generate simulated data $\mathbf{Y}_{j|\theta_i} = [\mathbf{y}_{j,1|\theta_i}, \mathbf{y}_{j,2|\theta_i}, \dots, \mathbf{y}_{j,m|\theta_i}]$ from the model using θ_i , and (3) estimate the parameters from simulated data $\mathbf{Y}_{j|\theta_i}$, yielding $\hat{\theta}_i$. After repeating these steps multiple times, the true and estimated parameters are compared.

For each model, we generated 50 simulated trials from 100 simulated subjects, each represented by a different θ_i . Given that we are interested in point estimates rather than posterior distributions, we used differential evolution (DE) to find the maximum likelihood estimates (Storn & Price, 1997). The parameters were sampled from the same distributions used to generate training data for the LANs, except the ranges were decreased by 30% to prevent estimates from falling outside the training data.

Simulation Results

LAN Test Accuracy

We assessed the accuracy of the LAN using visual inspection and quantitative measures. Figure 4 shows that probability density from the LAN provides a good fit to the histogram of simulated data. The out of sample correlations and root mean squared errors were 0.99 and 0.07 for the unimodal VSM and 0.99 and 0.11 for the multimodal VSM.

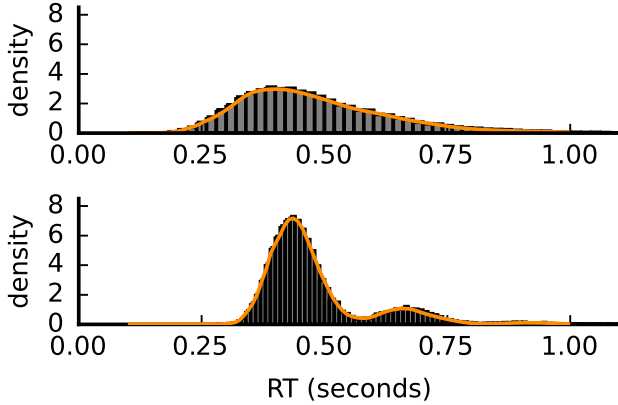


Figure 4: The probability density overlaid on histogram of simulated data for a “present” response on a target present trial with a distractor set size of 2. Top: the unimodal VSM. Bottom: the multimodal VSM.

Parameter Recovery

We evaluated the quality of parameter recovery with the correlation between true and estimated parameters and mean relative bias (e.g., $\frac{1}{n} \sum_{i=1}^n \frac{\hat{\theta}_i - \theta}{\theta}$). We only visualize the results for the unimodal model in Figure 5 due to space limitations. For the unimodal VSM, correlation and mean relative bias were .95 and .14 for ω_{td} and .94 and .08 for Δ . For the multimodal VSM, correlation and mean relative bias were .94 and .03 for ω_{td} and .94 and .02 for Δ . Given that the density overlay fit the simulated data well (e.g., Figure 4), its possible that the bias is an inherent property of the model. Overall, the parameters were recovered with an acceptable degree of accuracy for both versions of the VSM.

Timing Benchmark

We performed a benchmark to compare the execution times of LAN and PDA. Given that the evaluating the likelihood function is the primary bottleneck during inference, we decided to benchmark log likelihood of a single choice-rt pair. This means that the relative timing between PDA and LAN will be roughly constant across parameter estimation methods, but the absolute timing will depend primarily on the number of evaluations.

Unlike LAN, the PDA method is sensitive to the size of the distractor set and the number of samples is used to estimate the kernel density. For this reason, we varied both factors:

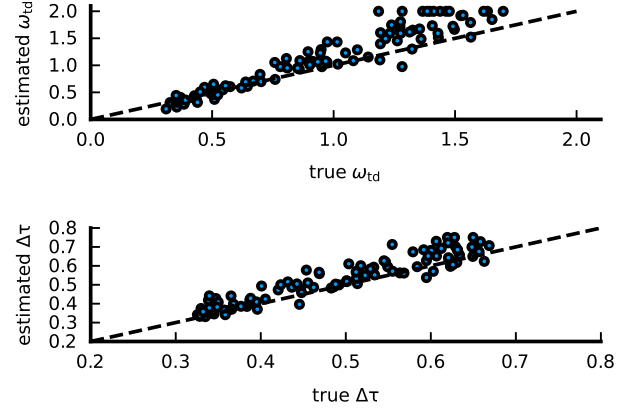


Figure 5: Scatter plot of the true and estimated parameters for the unimodal model.

the number of samples to construct the kernel density was 1k, 10k, 20k, and 50k, and the size of the distractor set was either 2 or 20. In each condition, we repeated the benchmark 1,000 times and computed the mean across replicates. The benchmark results in Figure 6 reveals three important findings: (1) LAN was 3-6 orders of magnitude faster the PDA, and (2) evaluation time for PDA increased with the number of samples, and (3) evaluation time for the PDA also increased with set size as expected. In addition, estimating the parameters for a single simulated subject using DE with LAN, three groups of 10 particles and 1,000 iterations required approximately 2.15 seconds. Parameter estimation would take significantly longer with PDA due to the use of Monte Carlo simulations.

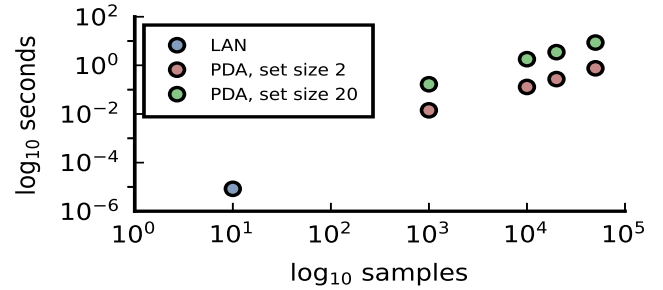


Figure 6: Mean time to evaluate log likelihood for LAN and PDA for distractor set size 2 and 20. Note that x-axis not applicable to LAN.

Discussion

Our goal in the present research was to determine to what extent LANs can approximate the likelihood function of models which differ from previous applications. As a test case, we used a VSM based on the ACT-R cognitive architecture. Deriving a likelihood function for the VSM is challenging because the model is a complex mixture with a large number of component distributions. Extending prior research, we

demonstrated that LANs (1) can scale up to a large number of experimental conditions, and (2) are sufficiently flexible to approximate the likelihood for both unimodal and multimodal distributions.

Consistent with prior research, we also found that LANs were multiple orders of magnitude faster than simulation based approaches such as PDA. In addition, our results highlight an important feature of LANs—namely, evaluation time is invariant to factors that affect PDA. In the VSM, for example, simulation times increase with set size due to the increase in the number of simulated visual fixations. Indeed, the increase was approximately 1 order of magnitude at the extremes (2 vs. 20). An increase in simulation time also occurs by decreasing the updating factor of the termination threshold. This effect is more pronounced during the simulation of target absent trials.

Many interesting questions remain for future research to address regarding the scalability of LANs to other more complex model structures commonly encountered in cognitive architectures. In prior investigations, including our own, the task consisted of a simple, repetitive trial structure involving a single goal and response mode. However, the structure of some tasks is less rigid and may require multiple response modes. Aviation tasks, for example, may require a person to prioritize multiple, potentially conflicting goals within a dynamic environment. Aviation controls are numerous and varied, ranging from buttons, dials, throttle levers, and rudder pedals. The resulting model structure for this type of task is multivariate and dynamic. In addition, unlike a simple laboratory task, the environment is *reactive*—responding modifies the environment, which in turn, creates new set of conditions with which a person must interact. The universal approximation theorem (Hornik et al., 1989) suggests that LANs can, in principle, be extended to accommodate more complex structures. The question, however, is whether this can be achieved in practice given limited computational resources.

Conclusion

In the past, mathematical intractability and computational limitations have impeded our ability to use likelihood based methods to evaluate many models based on cognitive architectures. We believe that LANs provide a promising method for approximating intractable likelihood functions. More research will be needed to identify boundary conditions and limitations of LANs. Nonetheless, our research provides evidence that the scope of applicable models is somewhat larger than previously known.

Acknowledgments

The opinions expressed herein are solely those of the authors and do not necessarily represent the opinions of the United States Government, the U.S. Department of Defense, the Department of the Air Force, or any of their subsidiaries or employees. This research was supported through Air Force internal funds. Distribution A Approved for public release. Case number: AFRL-2023-1410.

References

- Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111(4), 1036–1060.
- Boelts, J., Lueckmann, J.-M., Gao, R., & Macke, J. H. (2022). Flexible and efficient simulation-based inference for models of decision-making. *Elife*, 11, e77220.
- Fengler, A., Govindarajan, L. N., Chen, T., & Frank, M. J. (2021). Likelihood approximation networks (lans) for fast inference of simulation models in cognitive neuroscience. *Elife*, 10, e65074.
- Fisher, C. R., Hout, J. W., & Gunzelmann, G. (2022). Fundamental tools for developing likelihood functions within act-r. *Journal of Mathematical Psychology*, 107, 102636.
- Fisher, C. R., Walsh, M. M., Blaha, L. M., Gunzelmann, G., & Veksler, B. (2016). Efficient parameter estimation of cognitive models for real-time performance monitoring and adaptive interfaces. In *Proceedings of the 14th international conference on cognitive modeling*.
- Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359–366.
- Kruschke, J. (2014). Doing bayesian data analysis: A tutorial with r, jags, and stan.
- Moran, R., Zehetleitner, M., Müller, H. J., & Usher, M. (2013). Competitive guided search: Meeting the challenge of benchmark rt distributions. *Journal of Vision*, 13(8), 24–24.
- Newell, A. (1990). *Unified theories of cognition*. Harvard University Press.
- Nyamsuren, E., & Taatgen, N. A. (2013). Pre-attentive and attentive vision module. *Cognitive Systems Research*, 24, 62–71.
- Palmer, E. M., Horowitz, T. S., Torralba, A., & Wolfe, J. M. (2011). What are the shapes of response time distributions in visual search? *Journal of Experimental Psychology: Human Perception and Performance*, 37(1), 58.
- Papamakarios, G., Sterratt, D., & Murray, I. (2019). Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd international conference on artificial intelligence and statistics* (pp. 837–848).
- Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, 85(2), 59.
- Storn, R., & Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4), 341–359.
- Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, 12(1), 97–136.
- Turner, B. M., & Sederberg, P. B. (2014). A generalized, likelihood-free method for posterior estimation. *Psychonomic Bulletin & Review*, 21(2), 227–250.

Tetrad Fit Index for Factor Analysis Models

Víthor Rosa Franco (vithorfranco@gmail.com)

Graduate Program of Psychology, São Francisco University,
Rua Waldemar César da Silveira, Jardim Cura D'Ars, 105, 13045510 - Campinas, SP BRA

Rafael Valdece Sousa Bastos (rafavsbastos@gmail.com)

Graduate Program of Psychology, São Francisco University,
Rua Waldemar César da Silveira, Jardim Cura D'Ars, 105, 13045510 - Campinas, SP BRA

Marcos Jiménez (marcosjinezhquez@gmail.com)

Instituto de Ingeniería del Conocimiento C/ Francisco Tomás y Valiente, nº 11
Escuela Politécnica Superior (EPS), Edificio B, 5ª planta Universidad Autónoma de Madrid (UAM)
28049 - Cantoblanco, MAD ESP

Abstract

Traditional fit indices used in the context of factor analysis are based on the objective function of the Maximum Likelihood (ML), or modified ML, estimates of the free parameters. Therefore, these indices are an indication of how well the fitted model describes the observed correlation matrix. However, these indices do not provide a direct assessment of the validity of the assumed causal relations between the latent and observed variables. The objective of this study is to propose a tetrad fit index (TFI) that indicates how well the assumed causal relations in the model are reflected in the data. The TFI is defined as the complement of the average of the root-mean-squared difference between the tetrads of the observed correlation matrix and the correlation matrix implied by a fitted factor analytic model. A preliminary simulation study provides initial evidence in favor of using the TFI instead of other traditional fit indices to identify the correct factor model in comparison to concurrent models.

Keywords: Model comparison; causal inference; psychometrics

Introduction

In factor analysis and structural equation modeling applications one is interested in identifying how well a theoretical model reflects the data. Several fit indices have been developed to operationalize what “well” means in this context. Two main classes of fit indices were proposed to try to operationalize the “goodness” (or “badness”) of models (Xia & Yang, 2019): incremental fit indices, and absolute fit indices.

Absolute fit indices assess how far the fitted model is from a “perfect” model, whereas a “perfect” model is defined as the model that can perfectly predict the values of the observed correlation matrix. One of the most used absolute fit indices is the root mean square error of approximation (RMSEA; Steiger & Lind, 1980). Incremental fit indices, on the other hand, assess how the fitted model performs in comparison to a “baseline” model. The baseline model, in this context, is usually defined as the model where all variables are considered to be independent and, therefore, should be the model with the worst possible fit. The comparative fit index (CFI; Bentler, 1990) and the Tucker-Lewis index (TLI; Tucker &

Lewis, 1973) are two of the most commonly used incremental fit indices.

Independent of a fit index being incremental or absolute, the “quality” of the fit is defined according to the objective function of the factor model, which is usually defined in terms of some type of difference between the observed correlation matrix and the correlation matrix implied by the fitted model (or something alike). However, Spearman (1904) has shown that whenever a set of observed variables is linearly caused by a common latent variable, some more implicit patterns arise in the correlation matrix. More specifically, Spearman (1904) has shown that the difference between the product of some pair of correlations and the product of another pair among four random variables with a common latent variable should be equal to zero. This difference is known as the vanishing tetrad (Hart & Spearman, 1912) and it reflects a consequence of the causal assumption of factor models.

The vanishing tetrads have been known for a long time, but their applications in psychometrics have been quite sparse. In fact, it is not unreasonable to state that, after Spearman, vanishing tetrads have been ignored in mainstream psychometric (Bartholomew, 1995). Justifications for this include the fact that tetrads are more computationally expensive to calculate (as will be shown later, it involves the calculation the difference of products of products), procedures based on tetrads are not easy to use or interpret, and some of the existing methods are not readily accessible in statistical software (e.g., Spirtes et al., 2000). The objective of this study is to propose a tetrad fit index (TFI) that is easy to calculate and interpret, integrating the family of absolute fit indices with causal information that is not included in the current existing fit indices.

Factor Analysis and Traditional Fit Indices

The main objective of factor analysis is to find a structure of latent causes that can be used to explain the correlational structure of the observed data. In formal terms, the observed correlation matrix Σ is assumed to be the result of a linear combination of the factor loading matrix Λ , the matrix of

correlations between the latent causes Φ , and the diagonal matrix of uniqueness Ψ (i.e., the part of the variance of each item that is not shared with the other items):

$$\Sigma = \Lambda\Phi\Lambda^T + \Psi. \quad (1)$$

Computationally, the Maximum Likelihood (ML) estimator of the model in Equation 1 can be defined in several different ways, depending on the restrictions that one believes to be necessary for the estimation process in the given data (Xia & Yang, 2019). These restrictions are the properties that differentiates the traditional ML estimator from the DWLS, ULS, WLS, WLSMV, and other possible estimators (Flora & Curran, 2004). Therefore, all estimators are special cases of the traditional ML estimator (or objective/fit function) which can be expressed as

$$F_{ML} = \ln|\hat{\Sigma}| + \text{tr}(\Sigma\hat{\Sigma}^{-1}) - \ln|\Sigma| - p, \quad (2)$$

where $\hat{\Sigma} = \hat{\Lambda}\hat{\Phi}\hat{\Lambda}^T$, the hat operator ($\hat{\cdot}$) indicates estimated parameters, and p is the number of variables in the model. With a perfect model, $\hat{\Sigma} = \Sigma$ and $\text{tr}(\Sigma\hat{\Sigma}^{-1}) = p$; therefore, resulting in $F_{ML} = 0$.

In real applications, the model will be perfect usually only in the cases where the model is overidentified (i.e., the number of free parameters is larger than the amount of information in the model; Bamber & van Santen, 2000). Therefore, in most real applications, $F_{ML} > 0$ and the estimation method will find the smallest possible value of F_{ML} given the restrictions imposed over $\hat{\Lambda}$ and $\hat{\Phi}$. A consequence of this is that, apart from random error, misspecification of the restrictions imposed over the loadings and latent correlations matrices will also decrease the chances of finding a “good enough” model. Fit indices are, then, a way of checking if the identified model is, indeed, good enough to explain the data.

The RMSEA measures the “badness-of-fit” and is defined as

$$\text{RMSEA} = \sqrt{\frac{\hat{F}}{df}}, \quad (3)$$

where df are the degrees of freedom of the fitted model. The RMSEA represents the magnitude of the misfit given the number of free information (i.e., the dfs) of the model. The CFI measures the “goodness-of-fit” and is defined as

$$\text{CFI} = 1 - \frac{\hat{F}}{F_B}, \quad (4)$$

where F_B is the estimated objective function for the baseline model. The CFI is a likelihood ratio-like measure of goodness-of-fit, being equal to 1 only when the baseline model is infinitely worse than the fitted model. The TLI is defined as

$$\text{TLI} = 1 - \frac{\hat{F}/df}{F_B/df_B}, \quad (5)$$

where df_B are the degrees of freedom of the baseline model. The TLI also measures the “goodness-of-fit” and is interpreted similarly to CFI, but similarly to RMSEA it is also weighted by the relative free information of the fitted model in relation to the baseline model.

For a researcher to say if a model is “good enough” to explain the correlation structure of a set of data, the decisions based on fit indices are dependent on a set of cutoff criteria (Bentler & Bonett, 1980; Jöreskog & Sörbom, 1993). For instance, Hu and Bentler (1999) have shown, through simulation studies, that an RMSEA smaller than .06 and a CFI and TLI larger than .95 indicate a relatively good model-data fit for continuous observed variables. With nominal and ordinal data, however, these fit indices tend to be biased in the direction of a good fit. Therefore, with nominal and ordinal data, one should use more stringent criteria or yet another decision criterion for model selection (Xia & Yang, 2019).

One important procedure of model selection that is usually overlooked in the psychometric literature is that of model comparison. In this perspective, instead of depending heavily on the “good enough” fit indices, researchers compare theoretically competing models and, based on the relative differences of the fit indices, choose the model that provides the best possible fit to the data. We believe (and some other authors also hold this view; e.g., Xia & Yang, 2019) that model comparison can sometimes be more efficient than selecting models based on somewhat arbitrary cut-off criteria.

Apart from the discussion of the best way of using fit indices to make theoretically meaningful decisions, it is also a heated debate in the literature of what is the “best” fit index to decide on what model better describe the data (e.g., Heene et al., 2011; Sun, 2005). Because RMSEA, CFI, and TLI are all based on similar principles, they tend to be quite correlated. However, it is also not uncommon that one fit index indicates that the model is “good enough” for the given data, but another fit index indicates that the model is not good enough for the given data. Several simulation studies have then been conducted to show what fit index works better in what context (e.g., Heene et al., 2011; Hutchinson & Olmos, 1998; McNeish & Wolf, 2021; Shi & Maydeu-Olivares, 2020). Despite some interesting results, mathematically, the models are quite similar, and it is reasonable to state that RMSEA, CFI, and TLI are all some type of standardized effect size of the difference between the estimated correlation matrix and the observed correlation matrix (i.e., some type of residual-based measure).

In the recent psychometric literature, researchers have been discussing applications of probabilistic graphical models (known in this context as “network psychometrics”; Epskamp et al., 2018) as an alternative way to explain/describe the correlation patterns found among observed variables. For some of these models, latent variables are not considered. In fact, some simulation (e.g., van Bork et al., 2021) and theoretical (e.g., Kruis & Maris, 2016) studies have shown that network and factor analytic models can sometimes explain the same patterns of correlation. This highlights a limitation of fit indices such as RMSEA, CFI, and TLI for the assessment of the “quality” of factor models: they do not necessarily consider the causal assumptions embedded in factor models. Therefore, a fit

Implied Correlations

$$\begin{aligned}\rho_{12} &= \lambda_1 \lambda_2 \\ \rho_{13} &= \lambda_1 \lambda_3 \\ \rho_{14} &= \lambda_1 \lambda_4 \\ \rho_{23} &= \lambda_2 \lambda_3 \\ \rho_{24} &= \lambda_2 \lambda_4 \\ \rho_{34} &= \lambda_3 \lambda_4\end{aligned}$$

Implied tetrads

$$\begin{aligned}\tau_{1234} &= \rho_{12}\rho_{34} - \rho_{13}\rho_{24} = 0 \\ \tau_{1324} &= \rho_{13}\rho_{24} - \rho_{14}\rho_{32} = 0 \\ \tau_{1432} &= \rho_{14}\rho_{32} - \rho_{12}\rho_{34} = 0\end{aligned}$$

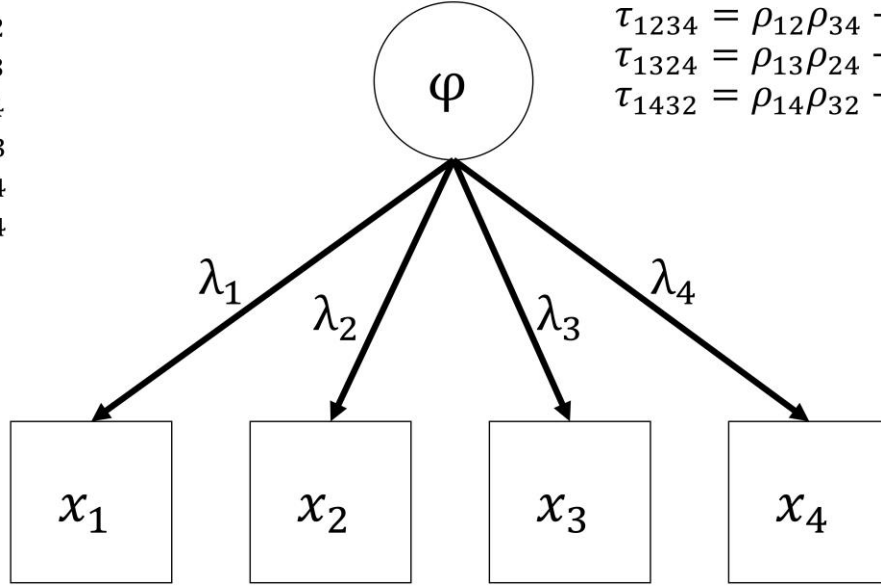


Figure 1: Basic factor model with implied correlations and tetrads

index that take into account the causal structure assumed by factor models could, in principle, provide additional information that is necessary to perform more adequate model selection.

Vanishing Tetrads and the TFI

The usual assumption of linearity was used by Spearman (1904), and later deepened by Bollen (1989) and Glymour et al. (2000), to show that the correlation matrix may “hide” some evidence regarding the presence of a common latent cause to a set of observable variables. In more technical terms, the common latent cause imposes some restrictions to the correlation matrix of observed variables in a way that it is possible to test (given some qualitative assumptions such as linearity and multivariate normality) if the data was generated by a common cause data generating process or not (Bollen & Ting, 1993). In fact, this was the method originally proposed by Spearman to assess the goodness-of-fit of his factor models. However, computing vanishing tetrads is expensive and, therefore, Spearman’s approach was quickly abandoned in exchange for principal component analysis and maximum likelihood estimation (Bartholomew, 1995).

The calculation of the vanishing tetrads is presented in Figure 1. In a data-generating process with a single common cause ϕ for the same four observed variables (x_1 to x_4), the implied correlation ρ_{ij} between two observed variables i and j is simply the product of the variables’ factor loadings λ_i and λ_j . For the sake of simplicity, here the procedure assumes that the variance of the latent variable is equal to 1. However, it could be similarly calculated with the covariance matrix of the latent variables, without additional restrictions to their variances. The tetrads τ_{hijk} implied by this data-generating

process is, therefore, defined as the difference between the product of a pair of correlations and the product of another pair among four random variables:

$$\tau_{hijk} = \rho_{hi}\rho_{jk} - \rho_{hj}\rho_{ik}. \quad (6)$$

One should note at this point why is the tetrad analysis computationally expensive in comparison to the other methods. For RMSEA, CFI and TLI, the necessary values are calculated with the optimization method, so are readily available right after the model was fitted. For tetrads, however, one still need to do some additional steps. First, one needs to calculate the products between all the correlations. Then, one needs to calculate the products between all the products of correlations. And finally, one needs to calculate the difference between the products of products of correlations. Despite the fact that modern computers can do these operations quite efficiently, in comparison to other fit indices, these calculations require exponential time, dependent on the number of variables in the dataset.

Regarding the usefulness of tetrads per se, the most important consequence of the common cause data-generating process is that all the tetrads implied by this model should be equal to 0; therefore, vanishing tetrads. Previous work with vanishing tetrads includes the exploratory tetrad analysis proposed by Glymour et al. (2000) and the confirmatory tetrad analysis proposed by Bollen and Ting (1993). Some other work has also studied asymptotic properties of test statistics derived from the vanishing tetrads analysis (e.g., Kenny, 1974), allowing the development of test statistics based on tetrads. However, all these previous works have at least one of some major limitations: they are computationally expensive (e.g., the exploratory tetrad analysis require the calculation of all the tetrads dozens or even hundreds of times

before reaching a valid result); they are overpowered (i.e., are biased against the factor model); or are not easy to interpret. Also, except for the exploratory tetrad analysis, which is implemented in the TETRAD software (Sirtes et al., 2000), to our knowledge, none of these methods are implemented in readily accessible statistical software.

Departing from Bollen and Ting (1993) and Kenny (1974), we will use the ML estimate of a factor model as the reference for the tetrad fit index (TFI). But instead of using \hat{F} as the reference value for the calculation, we will use a direct comparison of $\hat{\Sigma}$ and Σ . More specifically, from all the tetrads $\hat{\tau}$ implied by the estimated correlation matrix $\hat{\Sigma}$ and from all the tetrads τ implied by the observed correlation matrix Σ , the TFI is defined as

$$TFI = 1 - \frac{\sum \sqrt{(\hat{\tau} - \tau)^2}}{k}, \quad (7)$$

where k is the number of all the tetrads calculated for both of the correlation matrices. A value of TFI equal to 1 represents a model that perfectly fits the data. A value of TFI equal to 0 represents a model that fits the data as badly as possible. It is also worth noting that a similar index could be calculated using the estimated and observed covariance matrices instead. However, this would produce an index with ranges sensible to the variance of the variables in the model, and, therefore, more difficult to interpret.

The TFI represents an advantage to some previous tetrad methods especially because it only requires calculating all the tetrads twice. The exploratory tetrad analysis, for instance, can take hundreds of iterations to finish and, at each iteration, it needs to calculate all of the tetrads again. In comparison to other fit indices, the TFI provides not only an indication of how well the model describes or predicts the observed correlation matrix but also if (or how well) the implications of the causal model assumed in the factor analysis hold in the observed data.

Method

We ran a pilot simulation study to investigate if the TFI could be, indeed, a promising fit index for factor models. We simulated 100 datasets that were generated with a 5-dimensional factor model, with 5 observed variables per factor and a constant sample size of 500 respondents. The correlation matrix between the latent variables was sampled from a Wishart distribution with degrees of freedom equal to 5 and with the parameter matrix defined as a diagonal matrix of size 5. The factor loadings were sampled from a uniform distribution with a lower bound equal to .4 and an upper bound equal to .9. From the sampled latent correlation and loading matrices, we calculated the implied correlation matrix, and then, we sampled the observed continuous variables. Therefore, the data generating process was a factor model with five orthogonal factors that were the latent causes of each block of five variables in all the simulated datasets. For each simulated dataset, we fitted two models: a “best empirical model”, estimated using the exploratory graph analysis procedure (EGA; Golino & Epskamp, 2017); and a

model that reflected the data generating process (i.e., the model that truly reflects our simulation process). The final step for each iteration was to calculate the fit indices CFI, TLI, RMSEA, and TFI. After we calculated the fit indices for all the simulated datasets, we calculated the proportion of times that the right decision (regarding what was the correct model to choose) was made.

Results

The results regarding the proportion of times that each index would choose the correct model are shown in Table 1. The TFI fit index proposed in this study showed the highest performance for selecting the best model, with 100% accuracy. In addition, TLI was the second best among these fit indices, while the worst average performance happened with both CFI and RMSEA. In Figure 2, we present scatterplots of the 100 fit indices calculated for all the simulated datasets. Therefore, these values represent the “agreement” between the methods: stronger correlations indicate that the methods would suggest that the same model is the one to best describes the data. The fit indices ending with “1” represent the measures for the model estimated with EGA. The fit indices ending with “2” represent the measured for the model that correctly represents the true data generating process. Therefore, the upper diagonal plots show that the TFI exhibits comparatively lower correlations with the other fit indices, whereas the other indices display significant correlations with each other. The average absolute correlation of the TFI with the other indices is of about 0.320. Among the other indices, the average absolute correlation is of about 0.948. This suggests that the TFI may be evaluating something that the other indices do not consider. Furthermore, it also indicates that the other indices tend to converge in regards to what model will be considered as the most appropriate, given a specific dataset.

Table 1: Simulation Results For TFI, CFI, TLI, and RMSEA Fit Indices.

	Average Performance	Lower IC 95%	Upper IC 95%
TFI	1	1	1
CFI	0.76	0.67	0.84
TLI	0.86	0.79	0.93
RMSEA	0.76	0.67	0.84

Discussion

In this study, we proposed a fit index for factor models using vanishing tetrads, called the Tetrad Fit Index (TFI), which we demonstrated, at least providentially, to be more effective than other fit indices commonly used in the literature. Our simulation study shows that TFI is more accurate to select the best model, providing more accurate results than other fit indices. One potential explanation for the superior performance of TFI is that it takes into account the causal relations between the latent variable and its’ indicators, capturing some essential information that other fit indices

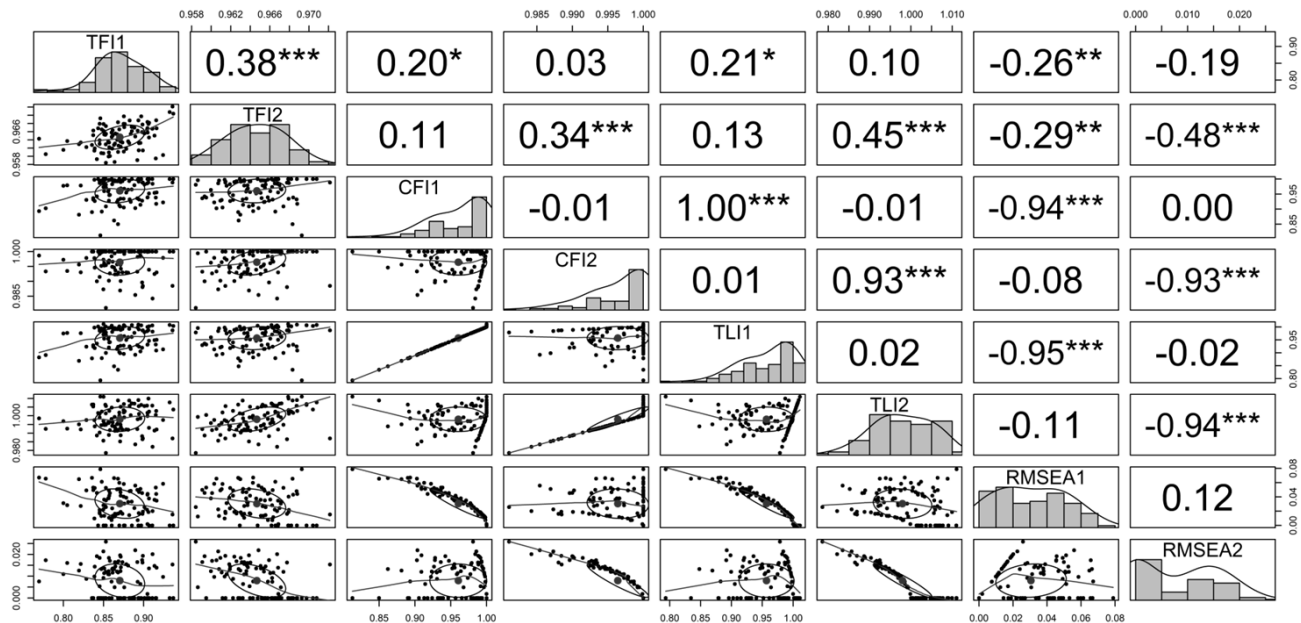


Figure 2: Correlations between fit indices

miss. It is worth noting that we still don't have sufficient evidence to state that TFI is the best-fit index for model selection in a variety of scenarios. Therefore, before a stronger conclusion is drawn, further studies with different simulation configurations are necessary. For instance, it would be particularly interesting to test if the TFI still works well when considering complex models such as second-order, bifactor, random-intercepts, and whatever combination of these models. However, our results suggest that researchers should consider testing the performance of the TFI as an alternative to other commonly used fit indices. We encourage researchers to use TFI in their future studies and to continue developing similar fit indices that can better capture the assumptions of the latent variable theory.

References

- Bamber, D., & Van Santen, J. P. (2000). How to assess a model's testability and identifiability. *Journal of Mathematical Psychology*, 44(1), 20-40. <https://doi.org/10.1006/jmps.1999.1275>
- Bartholomew, D. J. (1995). Spearman and the origin and development of factor analysis. *British Journal of Mathematical and Statistical Psychology*, 48(2), 211-220. <https://doi.org/10.1111/j.2044-8317.1995.tb01060.x>
- Bentler, P. M. (1990). Comparative fit indexes in structural models. *Psychological Bulletin*, 107, 238-246. <https://doi.org/10.1037/0033-2909.107.2.238>
- Bentler, P.M., & Bonett, D. G. (1980). Significance tests and goodness of fit in the analysis of covariance structures. *Psychological Bulletin*, 88, 588-606. <https://doi.org/10.1037/0033-2909.88.3.588>
- Bollen, K. A. (1989). *Structural equations with latent variables*. John Wiley & Sons.
- Bollen, K. A., & Ting, K. F. (1993). Confirmatory tetrad analysis. *Sociological Methodology*, 147-175. <https://doi.org/10.2307/271009>
- Epskamp, S., Maris, G., Waldorp, L. J., & Borsboom, D. (2018). Network psychometrics. *The Wiley handbook of psychometric testing: A multidisciplinary reference on survey, scale and test development*, 953-986. <https://doi.org/10.1002/9781118489772.ch30>
- Flora, D. B., & Curran, P. J. (2004). An empirical evaluation of alternative methods of estimation for confirmatory factor analysis with ordinal Data. *Psychological Methods*, 9(4), 466-491. <https://doi.org/10.1037/1082-989X.9.4.466>
- Golino, H. F., & Epskamp, S. (2017). Exploratory graph analysis: A new approach for estimating the number of dimensions in psychological research. *PloS One*, 12(6), e0174035. <https://doi.org/10.1371/journal.pone.0174035>
- Glymour, C., Scheines, R., Spirtes, P., & Kelly, K. (2000). *Discovering causal structure*. Academic Press.
- Hart, B., & Spearman, C. (1912). General ability, its existence and nature. *British Journal of Psychology*, 5, 51-84. <https://doi.org/10.1111/j.2044-8295.1912.tb00055.x>
- Heene, M., Hilbert, S., Draxler, C., Ziegler, M., & Bühner, M. (2011). Masking misfit in confirmatory factor analysis by increasing unique variances: A cautionary note on the usefulness of cutoff values of fit indices. *Psychological Methods*, 16(3), 319-336. <https://doi.org/10.1037/a0024917>
- Hu, L., & Bentler, P. M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling*, 6, 1-55. <https://doi.org/10.1080/10705519909540118>
- Hutchinson, S. R., & Olmos, A. (1998). Behavior of descriptive fit indexes in confirmatory factor analysis using

- ordered categorical data. *Structural Equation Modeling: A Multidisciplinary Journal*, 5(4), 344-364. <https://doi.org/10.1080/10705519809540111>
- Jöreskog, K. G., & Sörbom, D. (1993). *LISREL 8: Structural equation modeling with the SIMPLIS command language*. Scientific Software International.
- Kenny, D. A. (1974). A test for a vanishing tetrad: The second canonical correlation equals zero. *Social Science Research*, 3(1), 83-87. [https://doi.org/10.1016/0049-089X\(74\)90021-0](https://doi.org/10.1016/0049-089X(74)90021-0)
- Kruis, J., & Maris, G. (2016). Three representations of the Ising model. *Scientific Reports*, 6(1), 34175. <https://doi.org/10.1038/srep34175>
- McNeish, D., & Wolf, M. G. (2023). Dynamic fit index cutoffs for confirmatory factor analysis models. *Psychological Methods*, 28(1), 61-88. <https://doi.org/10.1037/met0000425>
- Shi, D., & Maydeu-Olivares, A. (2020). The effect of estimation methods on SEM fit indices. *Educational and Psychological Measurement*, 80(3), 421-445. <https://doi.org/10.1177/0013164419885164>
- Spearman, C. (1904). "General Intelligence" objectively determined and measured. *American Journal of Psychology*, 5, 201-293.
- Spirtes, P., Glymour, C. N., Scheines, R., & Heckerman, D. (2000). *Causation, prediction, and search*. MIT press.
- Steiger, J. H., & Lind, J. C. (1980). *Statistically based tests for the number of common factors*. Paper presented at the Annual Meeting of the Psychometric Society, Iowa City, IA.
- Sun, J. (2005). Assessing goodness of fit in confirmatory factor analysis. *Measurement and Evaluation in Counseling and Development*, 37(4), 240-256. <https://doi.org/10.1080/07481756.2005.11909764>
- Tucker, L. R., & Lewis, C. (1973). A reliability coefficient for maximum likelihood factor analysis. *Psychometrika*, 38, 1-10. <https://doi.org/10.1007/BF02291170>
- van Bork, R., Rhemtulla, M., Waldorp, L. J., Kruis, J., Rezvanifar, S., & Borsboom, D. (2021). Latent variable models and networks: Statistical equivalence and testability. *Multivariate Behavioral Research*, 56(2), 175-198. <https://doi.org/10.1080/00273171.2019.1672515>
- Xia, Y., & Yang, Y. (2019). RMSEA, CFI, and TLI in structural equation modeling with ordered categorical data: The story they tell depends on the estimation methods. *Behavior Research Methods*, 51, 409-428. <https://doi.org/10.3758/s13428-018-1055-2>

Single Neuron Distribution Modelling for Anomaly Detection and Evidence Integration

P. Michael Furlong¹ (michael.furlong@uwaterloo.ca)
 Madeleine Bartlett² (madeleine.bartlett@uwaterloo.ca)
 Terrence C. Stewart³ (terence.stewart@nrc-cnrc.gc.ca)
 Chris Eliasmith¹ (celiasmith@uwaterloo.ca)

¹Centre for Theoretical Neuroscience, University of Waterloo

²Cheriton School of Computer Science, University of Waterloo

³National Research Council of Canada, University of Waterloo Collaboration Centre,
 Waterloo, ON, N2L 3G1, Canada

Abstract

Probability theory is often used to model animal behaviour, but the gap between high-level models and how those are realized in neural implementations often remains. In this paper we show how biologically plausible cognitive representations of continuous data, called Spatial Semantic Pointers, can be used to construct single neuron estimators of probability distributions. These representations form the basis for neural circuits that perform anomaly detection and evidence integration for decision making. We tested these circuits on simple anomaly detection and decision-making tasks. In the anomaly detection task, the circuit was asked to determine whether observed data was anomalous under a distribution implied by training data. In the decision-making task, the agent had to determine which of two distributions were most likely to be generating the observed data. In both cases we found that the neural implementations performed comparably to a non-neural Kernel Density Estimator baseline. This work distinguishes itself from prior approaches to neural probability by using neural representations of continuous states, *e.g.*, grid cells or head direction cells. The circuits in this work provide a basis for further experimentation and for generating hypotheses about behaviour as greater biological fidelity is achieved.

Keywords: neural probability; spatial semantic pointers; anomaly detection; decision making; evidence integration

Introduction

Even without being tied to strict mathematical definitions, learning about the relative likelihood of different phenomena is useful for organisms operating in the world. Consequently, probability theory has become a useful tool in cognitive modelling. In this paper we use models of neural representations of continuous spaces to construct single neuron estimators of probability distributions. From those models we construct neural circuits for anomaly detection and evidence integration.

Approaches connecting neural activity to probability exist (see, *e.g.*, Doya et al., 2007). The Probabilistic Population Code (PPC; Ma et al., 2006, 2008) hypothesizes that neuron populations' activity represent posterior distributions conditioned on stimuli. Bayesian inference can be implemented using linear techniques, as can decoding distributions over the stimulus. PPC has been used to model cognition (*e.g.*, Ma et al., 2011; J. M. Beck et al., 2011; J. Beck et al., 2012; Hou et al., 2019; Walker

et al., 2020), including the forced decision task. Similarly, *convolutional codes*, in the terminology of Ma et al. (2008), posits that distributions over stimuli are encoded in neural populations' latent states, that can be linearly decoded (Anderson & Van Essen, 1994; Zemel et al., 1996; Eliasmith & Anderson, 2003; Barber et al., 2003). This technique has been employed to execute Bayesian inference in populations of spiking neurons (Sharma et al., 2017).

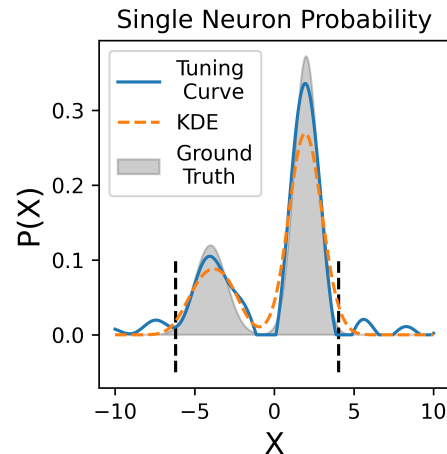


Figure 1: A probability distribution fit for 500 samples for a KDE and single-neuron SSP representation. Vertical dashed black lines indicate the bounds of the training data. Outside the training set the SSP estimator has peaks of probability due to the nature of the sinc quasi-kernel.

Alternatively, one can treat the activity of neurons not as encoding distributions, but samples from a distribution, (*e.g.*, Anastasio et al., 2000; Hoyer & Hyvärinen, 2002; Buesing et al., 2011; Huang & Rao, 2014; Kappel et al., 2015). In the case of Buesing et al. (2011) and Huang & Rao (2014), neuron activity is averaged to estimate the probability of the random variables the neurons represent, similar to the method presented in this paper. However, we differ in how we specify the synaptic weights of the neurons encoding probability.

More recently, work with the Semantic Pointer Archi-

texture (SPA; Eliasmith, 2013) has demonstrated the use of representations of continuous-valued data to model probability. Spatial Semantic Pointers (SSPs) are high-dimensional embeddings of data that have been used in high-level cognitive modelling (Komer, 2020; Voelker et al., 2021), as well as low-level state representations (Komer et al., 2019; Komer, 2020; Dumont et al., 2023), including grid cells in the medial entorhinal cortex (Dumont & Eliasmith, 2020). Furlong & Eliasmith (2022) showed that similarity between SSPs has a fundamental mathematical relationship with probability, and that algebraic manipulations of SSPs imply operations on distributions, extending prior results in hyperdimensional computing (Joshi et al., 2017; Frady et al., 2021).

In this paper we use SSPs to construct single neurons that encode probability distributions, and using those neurons, construct probabilistic algorithms. Figure 1 compares the tuning curve of such a neuron to a Kernel Density Estimator (KDE). To test SSP-based probability models, we use it to construct anomaly detection and evidence integration algorithms that are simple, but that have biological motivation. For anomaly detection, we simply threshold the output of a single neuron. This approach is similar to Dasgupta et al. (2018), who proposed single-neuron novelty detection circuit in the mushroom body of *Drosophila*. For the evidence integration task we employ a neural implementation of Wald’s sequential probability ratio test (SPRT). The SPRT has been suggested as a model of drift diffusion for forced-choice tasks, although it is not without its limitations (Drugowitsch & Pouget, 2012; Ratcliff et al., 2016). Further, a multi-hypothesis SPRT has been proposed as a framework for optimal decision making in the Basal Ganglia (Bogacz & Gurney, 2007; Bogacz & Larsen, 2011; Bogacz, 2015). What we show is how these cognitive representations can support the neural implementation of information theoretic algorithms.

We find that the neural distribution estimators have performance comparable to a non-neural KDE. While the neural anomaly detector does not reach the performance of a KDE baseline, it does achieve a high F1 score – a combined measure of precision and recall – relative to an analytical implementation of the anomaly detector. In the evidence integration task we find our model is statistically indistinguishable from a KDE baseline at a 95% confidence level. These results support the notion that one can, with confidence, combine cognitive representations like SSPs with neural circuits to implement probabilistic algorithms. Further, the particular circuits provide opportunities to generate hypotheses about neural structures and biological behaviour.

Methods

To illustrate the utility of single neuron distribution modelling we apply the technique in two settings. The

first application is novelty detection, which may have explanatory power in novelty-seeking behaviour. The second application is in decision making, framed using Wald’s sequential probability ratio test, which has been previously used as a metaphor for perceptual decision making and action selection in the basal ganglia.

Task description For the novelty detection task we present a sequence of observations drawn from a gaussian mixture model, $G_1(X) = 0.3\mathcal{N}(\mu_1 = -4, \sigma_1 = 1) + 0.7\mathcal{N}(\mu_2 = 2, \sigma_2 = 0.75)$. At some point t_{change} , the observation generating distribution switches to a second distribution, $G_2(x) = \mathcal{N}(\mu = 7, \sigma = 0.5)$. We run the experiment for $T = 2000$ observations and classify each observation as either “anomaly” or “non-anomaly”.

For the decision making task we required the network to classify a sequence of observations as being generated by one of two hypotheses, \mathcal{H}_0 or \mathcal{H}_1 , with distributions over observations, $P(x | \mathcal{H}_0) = \text{Beta}(\alpha_0 = 2, \beta_0 = 5)$, and $P(x | \mathcal{H}_1) = \text{Beta}(\alpha_1 = 5, \beta_1 = 2)$. Observations were then generated from distributions $G_{\text{obs}}(x) = \text{Beta}(\alpha_{\text{obs}}, \beta_{\text{obs}})$, where $\alpha_{\text{obs}} = \gamma\alpha_0 + (1 - \gamma)\alpha_1$, $\beta_{\text{obs}} = \gamma\beta_0 + (1 - \gamma)\beta_1$, and $\gamma \in [0, 1]$. Performance is assessed relative to an algebraic implementation of the SPRT algorithm for values of γ .

Algorithms and baselines: Fundamental to this work is the Spatial Semantic Pointer representation, which we use to model probability as described by Furlong & Eliasmith (2022). SSP encodings (eq. (1)) project lower dimensional data into a high dimensional vector representation, and are defined by a randomly selected encoding matrix, $\Theta_X \in \mathbb{R}^{d \times n}$, where d is the SSP dimensionality, and m is the dimensionality of the encoded data, $\mathbf{x} \in \mathcal{X}$.

$$\phi_X(\mathbf{x}) = \mathcal{F}^{-1} \left\{ e^{i\Theta_X \mathbf{x}} \right\} \quad (1)$$

Where \mathcal{F}^{-1} is the inverse Fourier transform. In this work we randomly sample the elements of the encoding matrix from the uniform distribution, $\mathcal{U}[-\pi, \pi]$. We use 1024 dimensional SSPs to encode samples from the probability distributions.

The dot product between m -dimensional data represented as SSPs approximates a product of sinc functions along each dimension of the m -dimensional data (eq. (2); Voelker, 2020). This function can be used as a quasi-kernel function for density estimation (Tsybakov, 2009).

$$\phi_X(\mathbf{x}/l) \cdot \phi_X(\mathbf{x}'/l) \approx \prod_{i=1}^m \text{sinc}(\|\mathbf{x}_i - \mathbf{x}'_i\|/l). \quad (2)$$

One can use the kernel trick (Rahimi et al., 2007) to approximate a kernel density estimator using simple linear methods, *e.g.*, $P(X) \approx \phi_X(\mathbf{x}) \cdot \frac{1}{n} \sum_i \phi_X(\mathbf{x}_i)$, given a dataset $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$. We can further equip the SSP encoding with a length scale parameter, l , which controls the bandwidth of the kernel function.

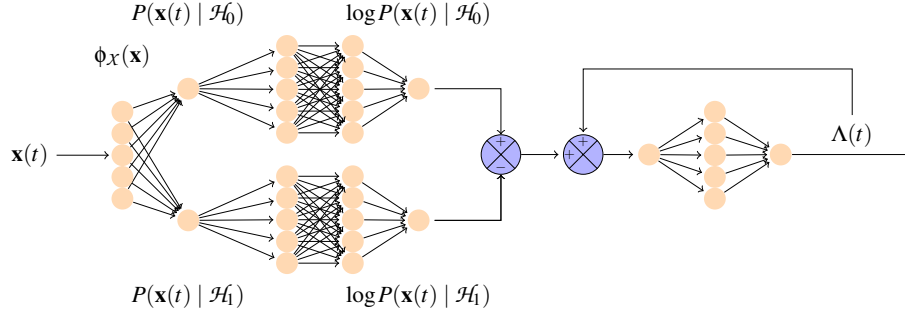


Figure 2: Schematic of neural circuit to compute the sequential probability ratio used in decision making. We use the principles of the NEF to represent probabilities and then compute log probabilities, and then use neural population dynamics to integrate log probability ratios.

The sinc function is a quasi-kernel because it can be negative, violating Kolmogorov’s axioms of probability. Glad et al. (2003) provide a correction for sinc kernel estimators – rectifying a biased version of the sinc-estimator (eq. (3)).

$$P(\mathbf{x}' | \mathcal{D}) \approx \max \{0, \mathbf{w} \cdot \phi(\mathbf{x}'/l) - \xi\}. \quad (3)$$

Where $\mathbf{w} = \frac{1}{n_l} \sum \mathbf{x}_i \phi(\mathbf{x}_i/l)$, and ξ is selected such that the function integrates to 1 over the input domain, \mathcal{X} . Equation (3) can be interpreted as a rectified linear (ReLU) neuron with synaptic weights, \mathbf{w} and bias ξ . Hence one can construct a single neuron that approximates a probability distribution.

Below we outline how to use this neuron model to make anomaly detection and evidence integration circuits. To do so we rely on the Neural Engineering Framework (Eliasmith & Anderson, 2003), the core principles of which are that we can: encode latent states in the activity of neural populations; linearly decode neural activity to transform it from one latent state to another; and use control theory to design neural circuits whose activity executes desired behaviour in said latent space.

Anomaly detection Novel observations are reported when the probability of the t^{th} observation is below the specified threshold. In this circuit a single neuron’s activity will reduce when it receives anomalous stimuli. Like Dasgupta et al. (2018), we posit its role as an inhibitory neuron that ceases inhibition in the presence of anomalies. We determine the synaptic weights for the neuron by first sampling $n = 5000$ observations from a distribution, $\mathbf{x}_i \sim G_1(X)$, and encoding the data as described above. We then define a neuron with activity, $a(t)$, given:

$$a(t) = \text{ReLU}(\mathbf{w} \cdot \phi_X(\mathbf{x}(t)) - \xi). \quad (4)$$

Anomalies are detected when $a(t) < \theta$, where θ is a threshold on the probability of an event. To determine the length scale parameter for the SSP encoding we used

the `estimate_bandwidth` function from the Scikit Learn clustering package with the quantile parameter set to 0.3.

As a baseline comparison we fit a kernel density estimator (KDE) using a Gaussian kernel, and selecting the length scale parameter using Silverman’s rule of thumb (Silverman, 1986). This method was fit on the same training set used for the single-neuron distribution model. To provide ground truth for the anomaly detection task we also implemented anomaly detection using the exact probability distribution, $G_1(X)$. To test the signal we generate 1000 samples from the data generating distribution, $G_1(X)$, which were then followed by 1000 samples from the anomalous distribution $G_2(X)$.

Decision making The second task is to integrate sequential observations to make a decision between two alternatives. This task has previously been formalized using Wald’s sequential probability ratio test (SPRT; Wald, 1945), a method for selecting hypotheses from sequential observations. The SPRT is defined in eq. (5), where $P(X | \mathcal{H}_0)$ and $P(X | \mathcal{H}_1)$ are probability distributions associated with the two possible decision outcomes.

$$\Lambda(t) = \sum_{\tau=1}^t \log(P(X = \mathbf{x}(\tau) | \mathcal{H}_0)) - \log(P(X = \mathbf{x}(\tau) | \mathcal{H}_1)) \quad (5)$$

The test integrates the log probability ratio, $\Lambda(t)$, of observations, $(\mathbf{x}(1), \dots, \mathbf{x}(t))$, until one of two decision thresholds are reached. If $\Lambda(t) > \theta_{\mathcal{H}_0}$, then hypothesis 0 is selected, and if $\Lambda(t) < \theta_{\mathcal{H}_1}$, then hypothesis 1 is selected. $\theta_{\mathcal{H}_0}$ and $\theta_{\mathcal{H}_1}$ can be specified using the desired false positive rate (fpr) and false negative rate (fnr) of the decision process, $\theta_{\mathcal{H}_0} = \log \frac{1-\text{fnr}}{\text{fpr}}$ and $\theta_{\mathcal{H}_1} = \log \frac{\text{fpr}}{1-\text{fpr}}$, respectively.

A diagram for a network that implements the SPRT is given in fig. 2. The first step of the circuit is to encode the observed point $\mathbf{x}(t)$ as an SSP. This SSP is then fed directly into two neurons, one neuron encoding $P(X | \mathcal{H}_0)$ and the other encoding $P(X | \mathcal{H}_1)$. Each probability neuron is fed into a population of 2000 neurons which also

represents the probability of the input observation. The neural population is then connected to another 2000 neuron population that approximates the quantity $\log(P(X | \mathcal{H}_k))$, $k \in \{0, 1\}$. The population size was chosen arbitrarily to ensure a good approximation of the log function. The $\log(P(X | \mathcal{H}_0))$ and $\log(P(X | \mathcal{H}_1))$ populations then converge on an integrator population (Eliasmith & Anderson, 2003, §8.2.1), with $\log(P(X | \mathcal{H}_1))$ scaled by -1 . The activity of this final neural population then represents the SPRT quantity, $\Lambda(t)$.

Because the log function is unbounded for values near zero, it is hard to approximate neurally. To mitigate this we use a “safe” log function for all algorithms, defined $\text{safe.log}(x) = \log \max\{\epsilon, x\}$, where $\epsilon = 10^{-3}$. To further ensure that log is well approximated for small values, the tuning curves of the neurons in the population representing probability were generated to disproportionately activate for small values of $P(X)$.

Analysis: Anomaly detection is a binary classification problem. We ran the system for 50 trials for 2000 randomly generated observations. The first 1000 samples are drawn i.i.d. from the true GMM distribution, while the second 1000 samples are drawn from a Gaussian distribution, $\mathcal{N}(\mu = 7, \sigma^2 = 0.25)$. To assess the ability of the single neuron distribution model to approximate a distribution we compare the probability of the generated samples under the true distribution, $P(X)$, with distributions estimated using a KDE, $\hat{P}_{\text{KDE}}(X)$, and using SSPs, $\hat{P}_{\text{SSP}}(X)$. Using a linear fit, we computed the coefficient of determination, R^2 . We also approximated the Total Variation (TV) goodness of fit, $\sup_{x \in \mathcal{X}} \|P(X) - \hat{P}(X)\|$. R^2 and TV were computed for each of the 50 trials, and were compared using a paired Student’s t-test. Where statistically significant differences were found, we assessed the effect size using Cohen’s d . We compared the novelty classifiers using the F1 score for each trial, as a function of the decision threshold, θ .

For the decision-making task we run 50 trials for each setting of the mixing parameter, γ . The circuit is tested on 200 observations drawn from the query distribution. We run the circuit until one of the decision boundaries have been crossed. In this setting there are three possible outcomes, either \mathcal{H}_0 or \mathcal{H}_1 are selected, or in the case of the SPRT quantity never crossing a decision threshold, then no decision is made (denoted N.D.). Because there is more than one possible classification we use the weighted F1 score, which is the F1 score computed one-vs-all for each class, weighted by the relative prevalence of the given class in the data set. Because each trial only computes one decision, we used the bootstrap method to compute the standard error for the F1 score.

Because there is a temporal aspect of the decision-making task we also report the error between the time the exact method arrives at a decision and when the tested al-

gorithms cross the decision threshold, $t_{\text{error}} = t_{\text{exact}} - t_{\text{alg}}$. We plot this quantity as a function of the mixing parameter γ with 95% confidence intervals computed over the corresponding 50 trials.

The above neural circuits were implemented using Nengo (Bekolay et al., 2014), the repository for the code is given in the Online Resources section. Baseline algorithms were implemented using SciKit Learn (Pedregosa et al., 2011). The experiments were executed on a computer with an Intel i5-8265U CPU, 8GB of ram, and running Ubuntu 22.04.1 LTS.

Results and Discussion

Anomaly Detection As can be seen in fig. 3, the KDE has a statistically significantly better linear relationship with the true probability than the SSP method, with a large effect size ($p < 0.001$, Cohen’s $d = 2.43$). However, we will note that the R^2 parameter is, on average, greater than 0.98, with 1 being a perfect linear relationship.

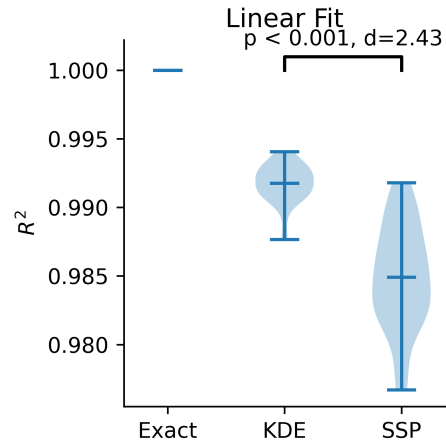


Figure 3: R^2 indicates the quality of a linear fit, with a value of 1 being a perfectly linear fit. While the KDE outperforms the SSP-based estimator, the single neuron estimator provides a reasonable level of performance.

The total variation scores for the KDE and the SSP estimator are statistically indistinguishable at a 95% confidence level, as shown in fig. 4. From this we can conclude that the upper bound on errors in probability estimates are similar for both the KDE and SSP estimator.

More interesting is the change in the F1 performance as the decision threshold is changed (fig. 5). For large values of the decision threshold, $\theta > 0.01$, the behaviour of the SSP estimator follows the KDE estimate, although both differ from the exact distribution. As the decision threshold approaches zero, we see that the F1 score for the SSP estimator decreases exponentially (linearly on the log scale). This is due to an increase in false negatives, as the SSP overestimates the probability of low-

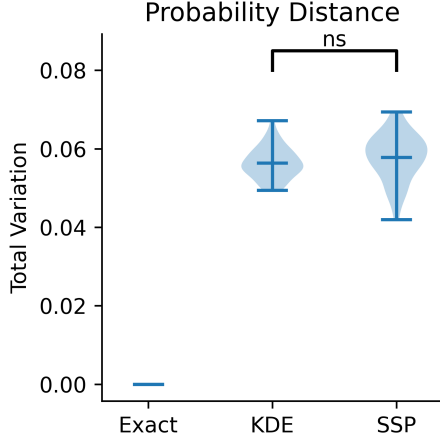


Figure 4: Total variation (TV) upper bounds element-wise error between two distributions, a TV of 0 indicates a perfect fit. In this test the total variation for the KDE and SSP estimators are indistinguishable.

probability events. Figure 1 shows the PDF estimated by the SSP estimator, and we see that outside the bounds of the input domain, the interference of the sinc kernel causes periodic but decreasing peaks of high-probability for what should be low-probability.

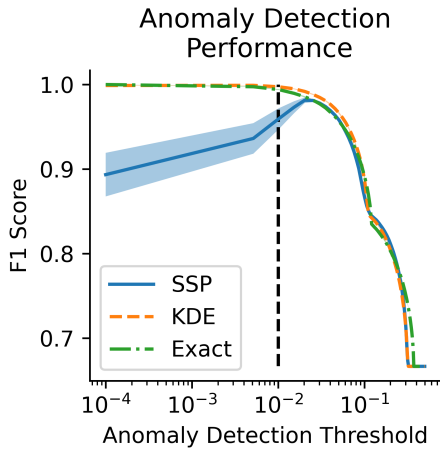


Figure 5: The above graph shows how the classification performance of the systems changes as a function of the decision threshold. While the SSP method follows the KDE reasonably closely, we see a decrease in performance as the threshold approaches zero. The dashed vertical line indicates a decision threshold of $\theta = 0.01$. Shaded regions indicate a 95% confidence interval.

Decision Making We compared the performance of the KDE and SSP estimators against the exact implementation of the SPRT algorithm. We found that both the KDE and SSP implementations were statistically indistinguishable from each other, and except for the ambigu-

ous distribution, where the mixing parameter $\gamma = 0.5$, they were in perfect agreement with the exact implementation, indicated by an F1 score of 1. The KDE is faster to

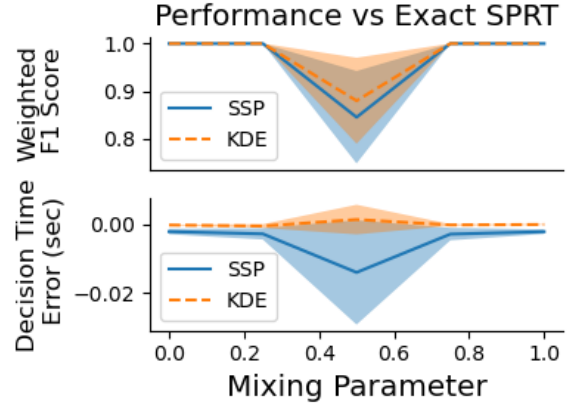


Figure 6: The top panel shows the F1 score for the multi-class classification of the decision-making task. The bottom panel shows the error in the decision time, relative to the exact implementation of SPRT. Shaded regions represent 95% confidence intervals.

reach a decision than the neural circuit, as can be seen in the bottom panel of fig. 6. Interestingly, in the ambiguous case of $\gamma = 0.5$, while it is not possible to draw firm conclusions due to the wide confidence intervals, it appears that the KDE comes to a decision faster than baseline, while the neural circuit tends to take longer.

One observation that should be made is that in the anomaly detection test we found that while the SSP was a reasonable fit to the probability distribution, it was not as good a fit as the KDE, yet their performance on the decision making task are effectively equivalent. For this we should consider two things. First, in the anomaly detection task, the SSP estimator had errors estimating small probabilities. In the SPRT task unlikely events are unlikely to be observed, thus favouring one of the two distributions in a forced choice task. Second, in the SPRT algorithm all approaches used the “safe” log function, where probabilities were limited to be no less than 10^{-3} . This mitigates the SSP estimator’s worse performance estimating small probabilities. If we were to make this lower-limit smaller in the current circuit performance is likely to decrease, but it could be recovered through the use of additional neural resources.

Conclusions

We have demonstrated that it is possible to use representations that we anticipate to exist in the brain to capture probability distributions, and use them to detect novelty and integrate evidence for decision making. We have demonstrated that neural encodings of continuous spaces

can support neural implementations of information theoretic algorithms. We have also constructed a neural circuit that uses these representations to compute the SPRT criterion. This circuit provides parameters that can be manipulated to provide hypotheses about behaviour.

We have found that neural approximations do deviate from the analytic model, although this is to be expected. We also note that using LIF neurons instead of ReLU neurons should cause further deviation from the exact expressions of novelty and evidence integration. Exploring these differences remains an area of ongoing work. However, because the tuning curve of the LIF neuron is a monotonically increasing function, it should be possible to make meaningful statements about probability.

In this work we assumed that all the training data for the synaptic weights of the single neuron distribution models were available instantaneously. This assumption does not hold for agents that are embedded in time. Indeed, an open question of future work is how online learning may be integrated into the circuit. To learn the synaptic weights that encode the distributions, which is fundamental to this approach, one could employ a simple learning rule, like the vectorized version of Oja's rule (Voelker et al., 2014). However, this will then have implications for how the ordering and recency of data effect decision-making. Online learning will also change the kernel approximated by the dot product from an atemporal kernel to a temporal one. Further, the SPRT task is criticized as a decision-making framework because of the fixed decision thresholds. The choice of decision threshold, and their modification, could be formulated as an RL task, allowing agents to have context-dependent criteria for confidence in a decision.

One may ask the question - why use these representations to model probability when exact probabilistic models already exist? This is a fair question, but we are asking the question: How far into neurology can we push probability as a model of cognition, given representations that unify high-level cognition and low-level implementation, and what implications do neural implementations have for probabilistic models? While more investigation is warranted, the present results constitute a modest first effort in these directions.

Online Resources

All code necessary to reproduce these results are hosted at: <https://gitlab.com/furlong/neural-anomaly-detection>.

Acknowledgments

This project was supported in part by collaborative research funding from the National Research Council of Canada's Artificial Intelligence for Logistics program (AI4L-116), as well as by CFI (52479-10006) and OIT

(35768) infrastructure funding, the Canada Research Chairs program, and NSERC Discovery grant 261453.

References

- Anastasio, T. J., Patton, P. E., & Belkacem-Boussaid, K. (2000). Using bayes' rule to model multisensory enhancement in the superior colliculus. *Neural Computation*, 12(5), 1165–1187.
- Anderson, C. H., & Van Essen, D. C. (1994). Neurobiological computational systems. *Computational intelligence imitating life*, 213222.
- Barber, M. J., Clark, J. W., & Anderson, C. H. (2003). Neural representation of probabilistic information. *Neural Computation*, 15(8), 1843–1864.
- Beck, J., Pouget, A., & Heller, K. A. (2012). Complex inference in neural circuits with probabilistic population codes and topic models. *Advances in neural information processing systems*, 25.
- Beck, J. M., Latham, P. E., & Pouget, A. (2011). Marginalization in neural circuits with divisive normalization. *Journal of Neuroscience*, 31(43), 15310–15319.
- Bekolay, T., Bergstra, J., Hunsberger, E., DeWolf, T., Stewart, T. C., Rasmussen, D., ... Eliasmith, C. (2014). Nengo: a python tool for building large-scale functional brain models. *Frontiers in neuroinformatics*, 7, 48.
- Bogacz, R. (2015). Optimal decision making in the cortico-basal-ganglia circuit. In *An introduction to model-based cognitive neuroscience* (pp. 291–302). Springer.
- Bogacz, R., & Gurney, K. (2007). The basal ganglia and cortex implement optimal decision making between alternative actions. *Neural computation*, 19(2), 442–477.
- Bogacz, R., & Larsen, T. (2011). Integration of reinforcement learning and optimal decision-making theories of the basal ganglia. *Neural computation*, 23(4), 817–851.
- Buesing, L., Bill, J., Nessler, B., & Maass, W. (2011). Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons. *PLoS computational biology*, 7(11), e1002211.
- Dasgupta, S., Sheehan, T. C., Stevens, C. F., & Navlakha, S. (2018). A neural data structure for novelty detection. *Proceedings of the National Academy of Sciences*, 115(51), 13093–13098.
- Doya, K., Ishii, S., Pouget, A., & Rao, R. P. (2007). *Bayesian brain: Probabilistic approaches to neural coding*. MIT press.
- Drugowitsch, J., & Pouget, A. (2012). Probabilistic vs. non-probabilistic approaches to the neurobiology of

- perceptual decision-making. *Current opinion in neurobiology*, 22(6), 963–969.
- Dumont, N. S.-Y., & Eliasmith, C. (2020). Accurate representation for spatial cognition using grid cells. In *Cogsci*.
- Dumont, N. S.-Y., Stöckel, A., Furlong, P. M., Bartlett, M., Eliasmith, C., & Stewart, T. C. (2023). Biologically-based computation: How neural details and dynamics are suited for implementing a variety of algorithms. *Brain Sciences*, 13(2), 245.
- Eliasmith, C. (2013). *How to build a brain: A neural architecture for biological cognition*. Oxford University Press.
- Eliasmith, C., & Anderson, C. H. (2003). *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press.
- Frady, E. P., Kleyko, D., Kymn, C. J., Olshausen, B. A., & Sommer, F. T. (2021). Computing on functions using randomized vector representations. *arXiv preprint arXiv:2109.03429*.
- Furlong, P. M., & Eliasmith, C. (2022). Fractional binding in vector symbolic architectures as quasi-probability statements. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 44).
- Glad, I. K., Hjort, N. L., & Ushakov, N. G. (2003). Correction of density estimators that are not densities. *Scandinavian Journal of Statistics*, 30(2), 415–427.
- Hou, H., Zheng, Q., Zhao, Y., Pouget, A., & Gu, Y. (2019). Neural correlates of optimal multisensory decision making under time-varying reliabilities with an invariant linear probabilistic population code. *Neuron*, 104(5), 1010–1021.
- Hoyer, P., & Hyvärinen, A. (2002). Interpreting neural response variability as monte carlo sampling of the posterior. *Advances in neural information processing systems*, 15.
- Huang, Y., & Rao, R. P. (2014). Neurons as monte carlo samplers: Bayesian inference and learning in spiking networks. *Advances in neural information processing systems*, 27.
- Joshi, A., Halseth, J. T., & Kanerva, P. (2017). Language geometry using random indexing. In *Quantum interaction: 10th international conference, qi 2016, san francisco, ca, usa, july 20-22, 2016, revised selected papers 10* (pp. 265–274).
- Kappel, D., Habenschuss, S., Legenstein, R., & Maass, W. (2015). Network plasticity as bayesian inference. *PLoS computational biology*, 11(11), e1004485.
- Komer, B. (2020). *Biologically inspired spatial representation*. Unpublished doctoral dissertation, University of Waterloo.
- Komer, B., Stewart, T. C., Voelker, A. R., & Eliasmith, C. (2019). A neural representation of continuous space using fractional binding. In *41st annual meeting of the cognitive science society*. Montreal, QC: Cognitive Science Society.
- Ma, W. J., Beck, J. M., Latham, P. E., & Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nature neuroscience*, 9(11), 1432–1438.
- Ma, W. J., Beck, J. M., & Pouget, A. (2008). Spiking networks for bayesian inference and choice. *Current opinion in neurobiology*, 18(2), 217–222.
- Ma, W. J., Navalpakkam, V., Beck, J. M., Berg, R. v. d., & Pouget, A. (2011). Behavior and neural basis of near-optimal visual search. *Nature neuroscience*, 14(6), 783–790.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Rahimi, A., Recht, B., et al. (2007). Random features for large-scale kernel machines. In *Nips* (Vol. 3, p. 5).
- Ratcliff, R., Smith, P. L., Brown, S. D., & McKoon, G. (2016). Diffusion decision model: Current issues and history. *Trends in cognitive sciences*, 20(4), 260–281.
- Sharma, S., Voelker, A., & Eliasmith, C. (2017). A spiking neural bayesian model of life span inference. In *Cogsci*.
- Silverman, B. W. (1986). *Density estimation for statistics and data analysis* (Vol. 26). CRC press.
- Tsybakov, A. B. (2009). *Introduction to nonparametric estimation*. Springer.
- Voelker, A. R. (2020). A short letter on the dot product between rotated fourier transforms. *arXiv preprint arXiv:2007.13462*.
- Voelker, A. R., Blouw, P., Choo, X., Dumont, N. S.-Y., Stewart, T. C., & Eliasmith, C. (2021). Simulating and predicting dynamical systems with spatial semantic pointers. *Neural Computation*, 33(8), 2033–2067.
- Voelker, A. R., Crawford, E., & Eliasmith, C. (2014). Learning large-scale heteroassociative memories in spiking neurons. *Unconventional Computation and Natural Computation*, 7(2014).
- Wald, A. (1945). Sequential tests of statistical hypotheses. *Annals of Mathematical Statistics*, 16(4), 117–186.
- Walker, E. Y., Cotton, R. J., Ma, W. J., & Tolia, A. S. (2020). A neural basis of probabilistic computation in visual cortex. *Nature Neuroscience*, 23(1), 122–129.
- Zemel, R., Dayan, P., & Pouget, A. (1996). Probabilistic interpretation of population codes. *Advances in Neural Information Processing Systems*, 9.

Illuminating Individual Learning Dynamics Within a Task: A Computational Model Analysis

Theodros Haile (theodros@uw.edu)

Chantel Prat (csprat@uw.edu)

Andrea Stocco (stocco@uw.edu)

Department of Psychology, University of Washington, Seattle

Abstract

Individual learners rely on different strategies (e.g., different combinations of declarative and reinforcement learning) to acquire new skills, but little is known about how these strategies change throughout the duration of learning. In this study, we fit four idiographic ACT-R models the first and second halves of a stimulus-response learning task (Collins, 2018) to identify learning strategy dynamics within an individual. We found that a majority learners were best described by a declarative memory (LTM) model in both halves of the task (86%). Of the minority of learners who were best described by a reinforcement learning strategy (RL) or combined RL-LTM strategy in the first half, most were more successful in the second half if they fit an LTM only strategy.

Keywords: individual differences, learning, learning strategies, reinforcement learning, declarative memory, ACT-R.

Introduction

Individual differences in learning strategies exist (e.g., Haile et al., 2020). The match between an appropriate learning strategy and task demand has been related to successful learning (e.g., DeCaro et al., 2008). But it is of considerable interest to investigate individual learning *dynamics*; in other words, do learning strategies change in an individual learner, for a particular task, during learning? Capturing these changes and identifying if a mismatch occurs between strategy and task demand, might facilitate better learning.

Several studies have successfully demonstrated that learning strategies can be captured using computational models (e.g., Collins, 2018; Haile et al., 2020). Learning strategies are defined as the combination of available learning mechanisms, like declarative long-term memory (LTM), reinforcement learning (RL), or even brief task completion using working memory (WM), that learners use to acquire new associations or skills. Individual learning strategies might arise due to differences in cognitive capacities like WM capacity (Just et al., 1992; DeCaro et al., 2008), and LTM decay rate (Haile et al., in review). But they are also affected by meta-cognitive evaluation of success with learning (Winne, 1996; Shute, 1991), previous knowledge (e.g., Heitzman et al., 2023; Cetron et al., 2020), and other factors that affect task engagement and motivation like fatigue (e.g., Krinsky et al., 2017). These might trigger a change in learning strategy during task progression. Some of these strategy changes might result in improvement of learning outcomes, while others might detract from successful learning.

In this study, we aim to use ACT-R (Anderson, 2007) based idiographic computational models, fit to two halves of a long learning task (RLWM task, Collins, 2018), separately, to detect if a strategy change occurs.

The RLWM task (Collins, 2018), is the task of choice for this study because its design makes it amenable to being completed using multiple strategies, as previous studies have established (Haile et al., 2020). But we also hope to exploit the fact that this task has a long learning phase (approximately 40 minutes) with two symmetric sets of learning blocks. Briefly, the RLWM task involves learning associations between images and letters in two learning conditions, easy short blocks, and difficult long blocks. Participants are then given a surprise test after a 10-minute break, which comprises images from across the 40-minute learning task. We hope to capture strategy changes, if they occur, by comparing model fits from the first 20 minutes of learning to the last 20 minutes, along with test accuracy for images from those time epochs. We further aim to ascertain whether a change in strategy is reflected in improvements to learning.

Materials and Methods

Participants. 83 undergraduate students from the University of Washington participated in this experiment. All participants were monolingual English speakers recruited through the UW Psychology subject pool (47 females, aged 18-35 years) who received extra credit for their participation. Data were collected after receiving informed consent in one 2-hour session.

Behavioral Task. The Reinforcement Learning Working Memory task (RLWM, Collins, 2018) involves learning stimulus-response associations through a series of 14 blocks. Participants are instructed to respond with a keypress of either 'C', 'V' or 'B' to the displayed images. In 8 of the 14 blocks, participants learn to associate keypresses with three unique images, presented 12 times in random order. In the remaining 6 of the 14 blocks, participants learn to associate 6 unique images each presented 12 times within the block with the key presses stated above. The stimulus-response associations are deterministic, and participants learn through reward (+1 point for correct responses and 0 points for incorrect responses). The same sequence of set-size 3 and 6 blocks was presented to all participants. Following this learning phase, a 10-minute distractor task is administered

before a surprise 206-trial test block. Participants make responses without feedback to items taken from both 3- and 6-set learning blocks. Stimulus presentations and data collection were done in MATLAB (mathworks.com) and Psychophysics Toolbox (Brainard, 1997).

Models

We built a series of four models in the ACT-R cognitive architecture to capture different learning strategies in the RLWM task (Anderson, 2007). We hypothesized that learners might use a single-component strategy, based on either the declarative long-term (LTM) or Reinforcement Learning (RL) memory systems or a multi-system approach with specific designs on integration. ACT-R was the optimal choice because of its integrated and flexible architecture for knowledge representation. ACT-R represents declarative memories as static records of information in its declarative module, and stimulus-response associations learned through reinforcement learning as conditional IF-THEN rules in its procedural module. These two modules interact with each other as well as with other perceptual and motor modules, capturing multiple aspects of cognition in a single framework. The use and acquisition of declarative and procedural representations are governed by a formal system of equations that capture the hallmarks of declarative and procedural memories, like memory decay over time, learning rate in response to feedback and, for explicit memories, the role of attention and working memory resources.

Declarative Learning Model. This single-system model stores memories of specific task events, like stimulus images, responses, and outcomes, for later recall and use. If it has never encountered a particular stimulus before, it executes a random response, the outcome of which is stored for later recall. If it does have memory of a previous encounter, it attempts to retrieve a response that led to a correct outcome, it makes a random response otherwise. All attempts and outcomes are memorized.

In ACT-R, declarative memories consist of multiple identical *traces*, each of which decay over time according to a power function (Anderson, 2007; Anderson 2000). The availability of a memory m depends on its activation $A(m)$, which is the log function of the sum of its decaying traces. Activation can be momentarily increased through spreading activation, an attentional mechanism that can be used to maintain information for a brief amount of time and reflects the weights W given to any existing association between a contextual cues q and m . Formally:

$$A(m, t) = \sum_i (t - t_i)^{-d} + \sum_q W S_{q, m} \quad (1)$$

We rely on three parameters that affect memory retrieval to capture individual differences: (a) activation noise s , which captures random fluctuations in a memory's activations and are associated with the probability of retrieval, (b) decay rate d , which captures the rate at which memories fade away and are forgotten (Sense et al., 2016);

and (c) spreading activation weight W , which captures the attentional resources allocated, and has been shown to capture individual differences in working memory capacity (Lovett, et al., 2000; Daily et al, 2001).

Reinforcement Learning Model. This second single-system model uses production rules to represent all the possible stimulus-response associations in the RLWM task. The model initially responds randomly, until the correct rule accrues sufficient rewards to overcome the competitors as the task progresses, and the interface provides feedback.

ACT-R's procedural module relies on reinforcement learning where the value or *utility* of a specific production, which contains a rule for a specific response, given a stimulus, is determined gradually through feedback.

We rely on two parameters that affect the utility of productions, learning rate (α), and selection noise (soft-max temperature (τ)). Specifically, each production rule p has an associated *utility* value, $U(p)$, that reflects its expected rewards and is learned through a temporal difference rule.

$$U_t(p) = U_{t-1}(p) + \alpha [R_t - U_{t-1}(p)] \quad (2)$$

in which α is the learning rate and R_t is the reward given at time t . In our experiment, R_t is binary and corresponds to the feedback ("Correct", $R_t = 1$, and "Incorrect", $R_t = -1$) given by the task interface. Competing responses are selected on the bases of their respective utilities, using a soft-max rule controlled by a noise parameter τ .

Integrated RL-LTM Model: Biased. The simpler of our two integrated, multi-system models utilizes a bias parameter (β), in addition to the two RL and three LTM parameters. This parameter explicitly biases the model to use its LTM or RL sub-system to deploy to learn and respond to task trials. The bias is set in proportions of RL-use from mostly LTM at 20% to mostly RL at 80% in twenty percent increments.

This model was designed with the expectation that learners might, somewhat rigidly, utilize a strategy that favors either RL or LTM or both, consistently throughout a learning task. The next model uses a more dynamic approach to address how systems might be integrated.

Integrated RL-LTM Model: Meta. This more complex version of our integrated model does not have additional parameters but includes meta-learning productions (prefer-RL and prefer-LTM) that are deployed dynamically throughout the task. They compete for task control through reinforcement learning, and the best subsystem, RL or LTM, is selected depending on how many rewards each was able to accumulate throughout the task. We measure what percentage of RL was used at the end of a simulation run.

This model assumes that individuals are adaptive learners and can optimally choose strategies based on their relative success over a short time. For example, if the long-term memory strategy proves too difficult (as in the case of too many stimuli), the model would switch to a RL-based learning strategy. RL learned associations are shared with the

LTM system by inserting explicit information into the memory module.

Experiment

For this study, we executed a two-pronged approach: (1) obtain model fitting reliability by performing parameter recovery (e.g., Wilson and Collins, 2019), and (2) split the learning data in half and fit models separately to assess if a change in individual strategy occurs. The parameter recovery step was necessary to establish the sensitivity and reliability of our models and model fitting procedure to identify a learning strategy. Here, we can generate simulated learning data from known learning strategies like Declarative long-term memory, Reinforcement learning and a mix of the two and see how well those ground truths are recovered.

Modelling Procedure. Each of the above models was run across a discretized range of its parameter space. Each model interacted with the same interface that displayed a stimulus, received response, and provided feedback. One simulation run contains 1 block of the short set-size 3 and 1 block of the longer set-size 6 condition. To obtain stable estimates, each model was run 100 times for each possible combination of parameters. In discretizing the range of each parameter, values were chosen to form an interval that surrounds the recommended value in the ACT-R documentation.

Model Fitting Procedure. Models were fit to each participant’s data by selecting parameters that maximized each model’s fit while penalizing the models’ complexity. To this end, the Bayesian Information Criterion (Schwartz, 1978) was used (see Results, below).

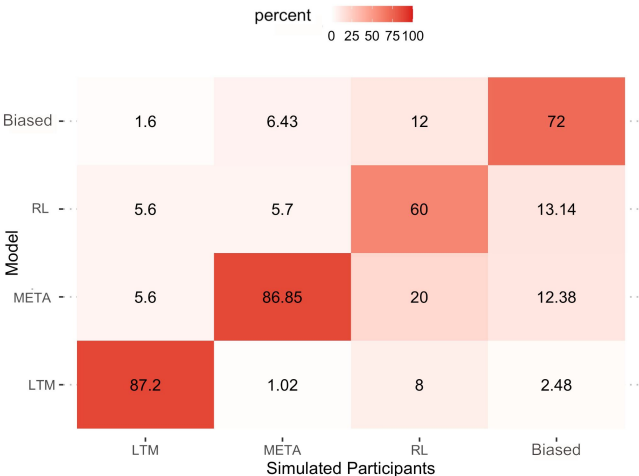


Figure 1: Proportions of simulated data (x-axis, 8 simulations for set-size 3 and 6 simulations for set-size 6) correctly identified by the model (100 simulations per model).

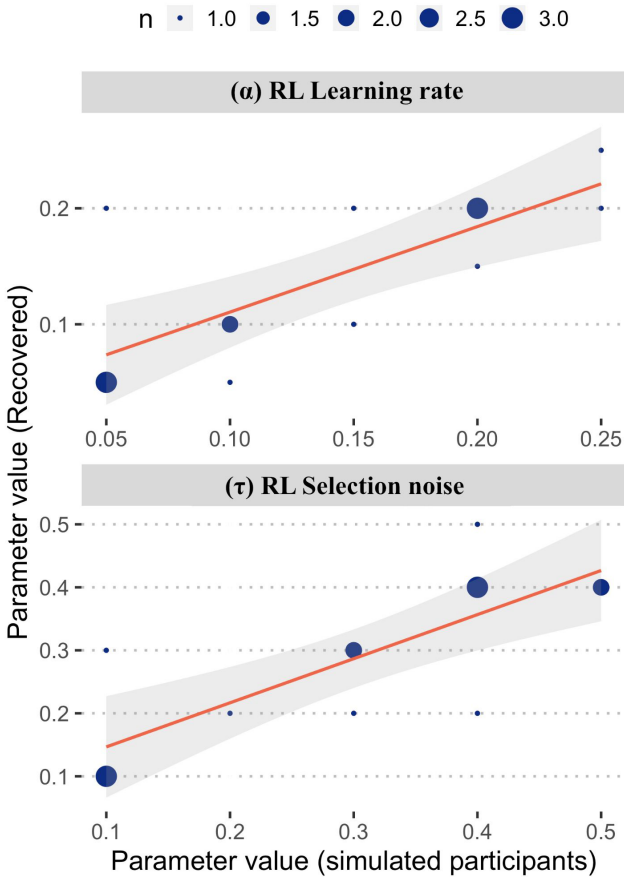


Figure 2: Correlation of recovered RL parameters (y-axis) with true parameter values from participant simulations (x-axis). Shown are the 15 unique parameter-sets, out of 25, that were correctly identified as RL.

Parameter recovery. To perform parameter recovery analysis a new set of model simulations were generated with the intent to mimic a level of experience with the task similar to that of human participants. Recall that human participants encountered 8 set-size 3 blocks and 6 set-size 6 blocks. One run of a simulation is equivalent to just one block of learning on the two set-size conditions, so final simulated data were averages of 8 (set-size 3) and 6 (set-size 6) simulations. This produced considerably noisier simulations that closely resembled our human participants. These simulated “participants” were then fit with the original set of 100-run simulations.

Split-half analysis. In this analysis, we split the 14-block learning data in half (labeled Half-1 and Half-2). Each half contained equal numbers of set-size 3 and 6 blocks. The test phase of the task contained one large block of 210 images sampled from all learning blocks, so the images were filtered by their occurrence in Half-1 and Half-2 to separately measure test accuracy for each half. Finally, the halves were separately fit to models to identify learning strategies. Figure 3: Correlation of recovered LTM parameters with true, simulation generating parameters. Size of the markers shows the count of overlapping points. Shown are the 109 unique

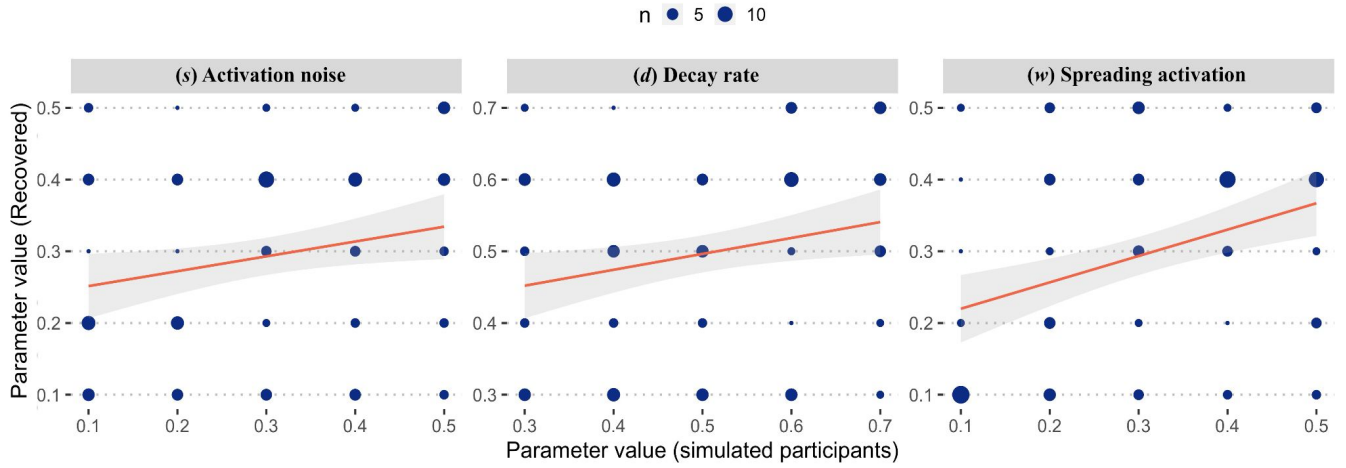


Figure 3: Correlation of recovered LTM parameters with true, simulation generating parameters. Size of the markers shows the count of overlapping points. Shown are the 109 unique parameter-set simulations (out of 125) that were correctly identified as coming from the LTM model.

parameter-set simulations (out of 125) that were correctly identified as coming from the LTM model.

and spreading activation (w), were poorly correlated with simulation parameter values ($d: r = 0.22$; $s: r = 0.21$; $w: r = 0.35$).

Parameter recovery for the two mixed models showed similar patterns to that of the two single-system models but correlations were considerably lower for the Biased model, even for the RL portions of the models ($\alpha: r = 0.22$, $\tau: r = 0.38$, $w: r = 0.15$, $d: r = 0.18$, $s: r = 0.03$), but well recovered

Results

Parameter recovery

In this two-pronged approach, we first performed the parameter recovery analysis. Here, as described above, simulated participants were generated using our ACT-R models and fit to 100-run simulations. All simulated subjects were fit to all four models and the best fit model was identified by selecting the fit that had the lowest BIC value produced by equation 3.

$$\text{BIC} = n + n \log(2\pi) + n \log(\text{RSS}/n) + \log(n)(k+1) \quad (3)$$

We performed two sets of analyses that answered the questions: (1) Does the procedure correctly identify which model produced the data? And (2) Does the procedure identify which sets of parameters, for a given model, produced the data?

Regarding the first question, we found that our models correctly identified where the simulated data came from 76.51% of the time. This percentage was highest for the LTM model at 87.2%, and lowest for the RL model at 60% (Meta: 86.85%; Biased 72%). Interestingly, 20% of the RL simulated data were identified as Meta (Figure 1).

Next, we tested how well parameters were recovered. Here, our success rate was different for the types of models. We achieved high levels of parameter recovery for RL parameters but not for LTM parameters (Figures 2 and 3).

Parameters for RL simulations were recovered well at $r = 0.76$ for α (learning rate) and $r = 0.79$ for τ (soft-max selection noise). For the LTM model however, recovered parameters for the LTM decay rate (d), selection noise (s),

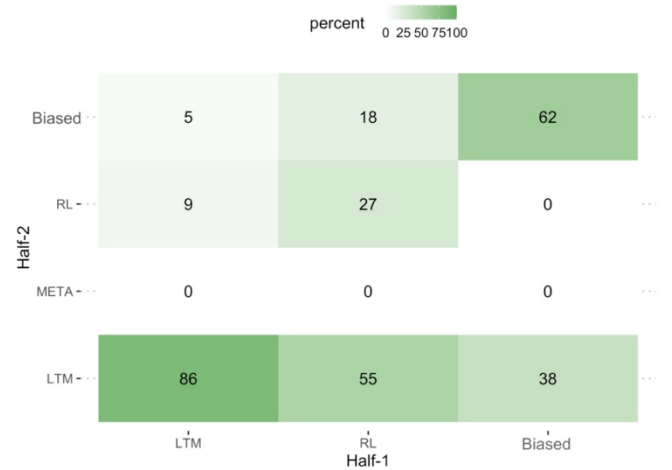


Figure 4: Matrix showing distributions of model fits across the two halves based on where participants landed in Half-1 (columns add up to 100%).

for most parameters in the Meta model ($\alpha: r = 0.51$, $\tau: r = 0.69$, $w: r = 0.53$, $d: r = 0.44$, $s: r = 0.10$). However, the bias parameter (β) was better recovered for the Biased model ($r = 0.58$) than the estimated bias in the Meta model ($r = 0.40$). This comparison takes advantage of the single-system models by providing a frame of reference, but it should be stressed that the parameters affect model performance in unison. For instance, when looking at the Meta model alone, preference for RL or LTM sub-systems (estimated post-hoc) was influenced by all the constituent parameters together

Split-half analysis

In the second part of this analysis, we sought to test if participants relied upon the same strategies to learn the associations throughout the task. So, the learning data were

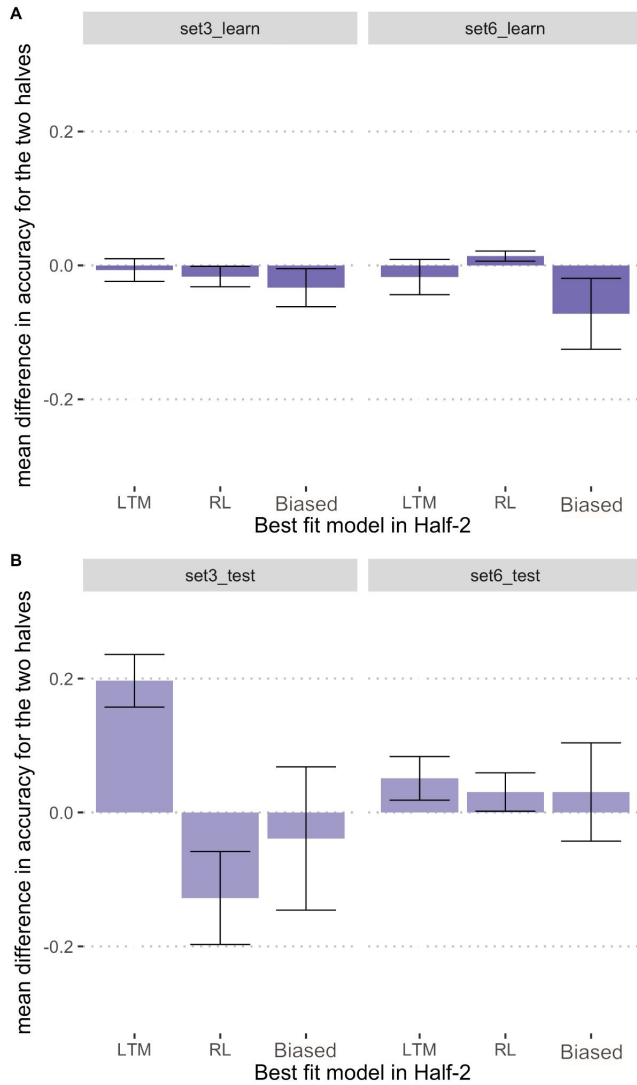


Figure 5: A) Shows the mean difference in accuracy between Half-1 and Half-2 for learning. B) Shows the mean difference in accuracy between Half-1 and Half-2 for Test. The columns are grouped by the best fitting model in Half-2 (x-axis). Of the 22 participants, 12 fit LTM, 5 fit RL and the remaining 5 fit the Biased model.

split into the first and last 7 blocks to compare learning outcomes and fit to models separately to identify strategies that might have led to those learning outcomes.

We first correlated behavioral learning outcomes - learning accuracy and test accuracy- for the two set-size conditions, from Half-1 to Half-2. This analysis was done to ensure that participants behaved similarly across the two halves of the task. Correlation was highest for learning accuracy in set-size

6 condition ($r = 0.67, p < 0.01$), and lowest for learning accuracy in the set-size 3 condition ($r = 0.34, p < 0.01$). Correlations between the two halves for the testing conditions were high and significant (set-size 3: $r = 0.57$; set-size 6: $r = 0.65$; $p < 0.01$). Additionally, there is not a significant main

effect of blocks in the 2 halves ($F(1, 656) = 0.96, p = 0.33$). This suggests that performance in the two halves, for most participants, is stable, and perhaps likely reliant on the same learning strategies, but we sought to answer this next, more robustly, by fitting models to each half and comparing identified learning strategies.

After fitting the models to each half separately, model predictions were compared. We found that 73.4% ($n = 61$) of participants fit the same models in the two halves, suggesting that strategy use was stable throughout the learning task. This percentage is higher (86%) for participants who fit the LTM model in Half-1 (Figure 4). It is important to note here that none of our participants fit the Meta model in either half and that most participants fit the LTM model most ($n = 56$, in Half-1), and the RL model least ($n = 11$; Biased: $n = 16$, in Half-1).

Let us assume for a moment that we were able to identify a true switch in strategy for the minority of participants who fit different models in the two halves ($n = 22$), is there a measurable benefit in learning outcomes for this group? Our results indicate that there were no statistically significant differences between the two halves, in-terms of accuracy, during learning and test phases within the two set-size conditions, when comparisons were agnostic to best-fit model types. However, when split up by best-fit model type, “switching” to a LTM strategy was associated with a higher increase in accuracy from learning to test, by almost 20%, but *only* for set-size 3, and *only* during test (Figure 5B). Similar benefits were not observed for the set-size 6 testing phase, and both set-sizes for accuracy at the end of learning. Accuracy, however, tends to be lower in the second half for the learning phase (Figure 5A).

Discussion

How learners use their available memory resources, what we call a learning strategy, affects how well they acquire new associations or skills. We hypothesized that some learners might rely mostly on single memory mechanisms like declarative long-term memory or use a mixture when learning a task. But it is not clear if strategy selection at the individual level is stable. Meaning, once a learner lands on a strategy, do they tend to alter their learning approach, perhaps based on meta-cognitive evaluation of learning success, or differing task demands or changes in motivation? In this study, we attempted to address this question by breaking up a long stimulus-response learning task into two identical halves, and test if different models explain learning in the two halves.

First, we demonstrated that our four ACT-R models can capture which memory system likely led to specific patterns

of behavior, by performing a parameter recovery procedure. This involves generating learning data on the task using known models and parameters, albeit with fewer ‘learning blocks’ or simulations than the testing models, to mimic noisy participant data. Our testing models, generated from 100-run simulations (equivalent to learning in 100 blocks) were then fit to the participant simulations, and the recovered parameters and models were compared to our ground truth values. This resulted in as high a congruence as 87% for the declarative only model (LTM) and as low as 60% for the Reinforcement Learning only model (RL). However, success in recovery of the generating parameters was mixed. We were able to obtain high correlations between true and recovered parameters for the RL model but not for the LTM model. This is perhaps owing to the fact that we only have 2 parameters for the RL model - learning rate (α) and selection noise (τ), where pairs of values resulted in unique patterns of behavior. For instance, low performance is only evident in situations where there is high noise, low learning rate or the combination of the two. But the LTM model has three parameters, memory decay rate (d), retrieval noise (s), and spreading activation (w), which leads to more ambiguous instances. In other words, many more combinations of parameter values could lead to similar results. In future studies we aim to use independent tasks to estimate these parameters separately to improve our predictive or measurement accuracy in-terms of estimating reliable individual parameters that describe and explain the learners’ strategy choices.

In the next analysis, we fit the models to each half of the data separately and compared model fits. The goal here was to identify if learners stably use a learning strategy throughout the task or switch. Strategy use dynamics might be one of the determining factors of successful learning outcomes. A learner might adjust strategies after meta-cognitive evaluation of outcomes or task requirements, due to fatigue, or change in motivation to complete the task.

We found that many participants (72%) fit the same models in the two halves. This was considerably higher for the participants who fit the LTM model in the first half. It also appears that most participants fit the LTM model in the second half, regardless of which model fit them best in the first half.

The RLWM task (Collins, 2018) can potentially be successfully completed (i.e., achieving high accuracy in learning with little or no decay during test) using either an RL, LTM or combined RL-LTM strategies as evidenced by our modeling efforts. Meaning, the major contributing factor to low learning accuracy, or large decay, seems to be the specific parameter values; all model instances that had favorable parameter values, regardless of model type, learned well. But a majority of our learners exhibit behaviors that are most similar to those generated by the LTM model. It would seem that, if they landed on an LTM strategy, they saw little reason to switch, hence the large percentage of participants who fit the same (LTM) model for both halves. We cannot identify why a strategy change occurred, but it appears that if

a learner did not start out with a strategy that resembled LTM, they made a switch. Surprisingly, this switch was associated with large gains in learning, at least for test accuracy in the set-size 3 condition, and minimally in the set-size 6 condition. While insignificant, there were fewer losses to learning accuracy in set-size 3, compared to the other strategies.

There are limitations to our modeling effort that deserve mention. It can be argued that an explicit, declarative strategy is popular with our participants, perhaps because they are university students where a lot of learning is instructed, explicit and declarative. It is also not too far a leap to suggest that most students would rely on semantic links between stimuli in the same block for the aforementioned reason. The stimuli in this task also are rich in detail and amenable to forming idiosyncratic semantic categories, which are supported by declarative memory. But our model fitting procedure might be biased towards the LTM model as it is also a relatively simple model with a small number of parameters. The BIC function that we minimize during model fitting penalizes larger models, so we tend to see fewer numbers of those models explaining learning behavior.

While we can label learning behavior with the model that likely produced it, we cannot confidently estimate the parameters that led to that behavior. This limits what we can explain about our learners and their learning outcomes. Lastly, we explored a coarse-grained, narrow set of parameters for our models. This might have led to some participants being shifted to a different model because a specific learning pattern did not exist in the hypothetical true model because of the limited range in parameter values. We hope to address these limitations in future studies by providing validation from independent tasks and exploring more fine grained parameter sets.

References

- Anderson, J. R. (2007). How can the human mind occur in the physical universe? *Oxford University Press*.
- Brainard, D. H., & Vision, S. (1997). The psychophysics toolbox. *Spatial vision*, 10(4), 433-436.
- Cetron, J. S., Connolly, A. C., Diamond, S. G., May, V. V., Haxby, J. V., & Kraemer, D. J. (2020). Using the force: STEM knowledge and experience construct shared neural representations of engineering concepts. *NPJ science of learning*, 5(1), 1-10.
- Collins, A. G. (2018). The tortoise and the hare: Interactions between reinforcement learning and working memory. *Journal of cognitive neuroscience*, 30(10), 1422-1432.
- Daily, L. Z., Lovett, M. C., & Reder, L. M. (2001). Modeling individual differences in working memory performance: A source activation account. *Cognitive Science*, 25 (3), 315-353.
- DeCaro, M. S., Thomas, R. D., & Beilock, S. L. (2008). Individual differences in category learning: Sometimes less working memory capacity is better than more. *Cognition*, 107(1), 284-294.

- Haile, T., Prat, C. S., & Stocco, A. (2020). One size doesn't fit all: Idiographic computational models reveal individual differences in learning and meta-learning strategies. In *Proceedings of the 18th International Conference on Cognitive Modeling*.
- Heitzmann, N., Stadler, M., Richters, C., Radkowitsch, A., Schmidmaier, R., Weidenbusch, M., & Fischer, M. R. (2023). Learners' adjustment strategies following impasses in simulations-Effects of prior knowledge. *Learning and Instruction*, 83, 101632.
- Just, M. A., & Carpenter, P. A. (1992). A capacity theory of comprehension: individual differences in working memory. *Psychological review*, 99(1), 122.
- Krimsky, M., Forster, D. E., Llabre, M. M., & Jha, A. P. (2017). The influence of time on task on mind wandering and visual working memory. *Cognition*, 169, 84-90.
- Lovett, M. C., Daily, L. Z., & Reder, L. M. (2000). A source activation theory of working memory: Cross-task prediction of performance in ACT-R. *Cognitive Systems Research*, 1(2), 99-118.
- Schwartz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6, 461–464.
- Shute, V. J. (1991). Who is Likely to Acquire Programming Skills? *Journal of Educational Computing Research*, 7(1), 1–24
- Stocco A., Lebiere, C., & Anderson, J. R. (2010). Conditional routing of information to the cortex: A model of the basal ganglia's role in cognitive coordination. *Psychological review*, 117(2), 541
- Wilson, R. C., & Collins, A. G. (2019). Ten simple rules for the computational modeling of behavioral data. *ELife*, 8, e49547.
- Winne, P. H. (1996). A metacognitive view of individual differences in self-regulated learning. *Learning and individual differences*, 8(4), 327-353.

An ACT-R Observer Model for Anticipatory Assistive Robots

Chenxu Hao (chenxu.hao@fau.de)

Chair of Autonomous Systems and Mechatronics,
Friedrich–Alexander Universität Erlangen–Nürnberg

Colin Halupczok (colin.halupczok@student.uni-tuebingen.de)

Hertie Institute for Clinical Brain Research and Center for Integrative Neuroscience,
University of Tübingen

Winfried Ilg (winfried.ilg@uni-tuebingen.de)

Hertie Institute for Clinical Brain Research and Center for Integrative Neuroscience,
University of Tübingen

Daniel Haeufle (daniel.haeufle@uni-tuebingen.de)

Hertie Institute for Clinical Brain Research and Center for Integrative Neuroscience,
University of Tübingen

Philipp Beckerle (philipp.beckerle@fau.de)

Chair of Autonomous Systems and Mechatronics,
Department of Artificial Intelligence in Biomedical Engineering,
Friedrich–Alexander Universität Erlangen–Nürnberg

Nele Russwinkel (nele.russwinkel@uni-luebeck.de)

Institute of Information Systems,
Universität zu Lübeck

Abstract

Interactions between human users and assistive robotic systems in real life often involve both cognitive and physical interactions. In order to support humans well in their daily life, a robotic agent needs to be aware of the situation, anticipate the human agent, and generate human-like behaviors. In this work, we present an ACT-R observer model as a possible implementation on the robotic agent's cognitive level. The model anticipates the human agent's behaviors in an application example: a tea-making task. We discuss how such a model provides us the possibility to connect cognitive and physical human-robot interactions, and the advantages of such a model compared with common state-of-the-art approaches for human intention and behavior predictions. We also discuss how such an individual ACT-R model provides potential for an anticipatory, situation-aware robotic agent in real life applications, allowing us to solve ambiguities from acquiring input via various sensors and gain time for proactive support.

Keywords: human-robot interaction; anticipatory thinking; ACT-R

Introduction

Assistive robots are becoming increasingly common, providing people with social support as well as physical assistance. We are particularly interested in improving robotic assistive devices that aim to help and support patients with motor control impairments in their everyday life such as a tremor (Castrillo-Fraile et al., 2019; Fromme, Camenzind, Riener, & Rossi, 2019). Such an assistive device can either be a robotic arm or a wearable assistive device.

Real-life interactions between the human user and assistive robotic systems often involve both cognitive interactions and physical interactions (Bartneck et al., 2020). On the cogni-

tive level, the robot infers human intentions and predicts human behaviors; on the physical level, the robotic agent provides the human with required physical assistance based on information on the cognitive level. In the daily life, assistive robotic devices for patients are often expected to also react proactively during the interaction in order to prevent injuries in potentially dangerous situations. This requires the robotic agent to have a cognitive understanding of the task situation, be able to anticipate and adapt to the human agent's actions, and generate goal-directed human-like behaviors (Hao, Russwinkel, Haeufle, & Beckerle, under review; Klein, Snowden, & Pin, 2011).

In this paper, we present an ACT-R based observer model that takes a step towards implementing a cognitive architecture for the assistive robot's cognitive layer. Our model aims to provide the assistive robot with the ability to be aware of the state of the task and the environment, and anticipate necessary future task states. We construct our model for a specific use case: a tea-making task. The model observes the human user's tea making actions, predicts the human user's next step, and provides an alert in anticipation of a potentially dangerous situation (e.g., spilling hot water).

In the remaining sections of the paper, we first briefly discuss models of situation understanding and anticipation in general, as well as requirements for our model. Then, we present our use case and our observer model. Lastly, we discuss our future steps.

Models of Anticipation

User anticipation for robots that assist human users in specific tasks requires the robot to 1) predict the user’s mental states based on the understanding of the user’s goals and intentions, i.e., to have the ability of Theory of Mind (ToM), and 2) be aware of the situation and have a shared representation of the task with the human user (Borst, Bulling, Gonzalez, & Russwinkel, 2022).

Various forward generative models and inverse models of human planning and decision making can already be used to predict the human user’s intentions and behaviors to the current state-of-the-art level (Ho & Griffiths, 2022), including computational models directly based on ToM (Berke & Jara-Ettinger, 2021; Jara-Ettinger, 2019; Rabinowitz et al., 2018). Such models can be very helpful for the robotic agent to achieve anticipatory thinking (Klein et al., 2011).

In addition, anticipation could be realized through constructed mental representations of tasks and specific situations to build up expectations about the human user’s intentions, goals, and mental states (Borst et al., 2022). Past work has shown this possibility with various models, including models based on visual attention (Wickens, 2015), mental models for human reasoning (Johnson-Laird, 2010), predictive coding (Friston & Kiebel, 2009), instance-based learning (Gonzalez & Dutt, 2011), models for situation awareness in decision making (Endsley, 2015), and so on. These approaches indicate that it is also helpful for the assistive robotic agent to have a structured representation of the situation.

Realizing anticipation with a structured representation of the situation and providing the robotic system with situation awareness can bring several advantages. Particularly, combined with external data input, a structured cognitive layer can make the robotic system more flexible to emerging situations, leading to the potential to generate real-time behaviors proactively. In addition, structured models can be theory driven (e.g., unified theory of cognition, Newell, 1994) and provide a transparent illustration of how the mind (in our case, the artificial mind of the robotic agent) observes data, keeps awareness of the situation, and anticipates further.

Cognitive architectures such as ACT-R (Anderson, 2009) provide us with a possibility to construct such structured systems that can take data input from the human user and provide real-time support or feedback in the interaction between the human user and the assistive robotic system (Fu et al., 2006). Besides, ACT-R’s structure also provides us a possibility to interface the robotic agent’s cognitive layer with models of the human’s motor system such as a neuromusculoskeletal model or interaction primitives (Amor, Neumann, Kamthe, Kroemer, & Peters, 2014) on the robotic agent’s physical layer, enabling the robot to produce physical assistance based on the anticipated task states from its cognitive layer.

Our work takes a first step towards such a structured ACT-R model by creating a simple observer model for an application example (or a use case) of real-life human-robot interaction—a tea-making task. However, we believe that a

structured anticipatory model have the potential for applications of assistive robotic systems for other tasks as well.

A Use Case for Assistive Robotic Systems

Consider an example where a human user is making tea with the help of a assistive robot. Tea-making is a simple task that happen in the daily life. However, patients with motor control impairments such as a tremor may find some aspects challenging—from picking up a spoon precisely to keeping the kettle or the mug stable while pouring hot water. When the patients experience a tremor episode while trying to make tea or drink hot tea, they may accidentally spill hot water, leading to potential injuries such as burning.

Besides the real-life implications, a tea-making task represents a good use case because the task has a general structure while allowing individual flexibility (see Figure 1 for an analysis of the task). Specifically, the usual ordering suggests that water and tea leaves should be readily prepared before the tea is made, but each human user may have individual habit or preference over whether boiled water is poured into cup first or if tea leaves are put into cup first. This task feature particularly challenges the model to attend to the interaction environment and be aware of the changes in the task state regardless of the user’s individual preferences.

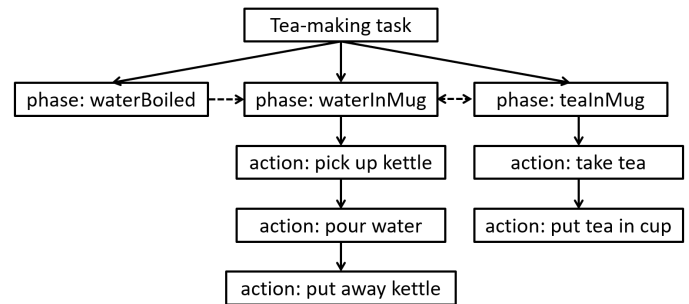


Figure 1: An analysis of the tea-making task: we assume that the general task structure contains three main phases: boiling water, putting water in mug, and putting tea in mug, with the order where boiling water precedes putting water in mug and the order of the other two phases interchangeable. The actions within each phase follow a specific order.

In order to achieve this, the anticipatory model for the robotic agent needs to have a general task-specific knowledge about tea making sequences. The model also needs to have a shared situation understanding or representation as the human user. As the human agent performs sequences of actions (e.g., pour water in the kettle), the task situation changes. By processing information about the interaction environment such as whether the water in the kettle is hot, the model integrates observed interactions and keeps an awareness of the state of the task. With these captured essential aspects for anticipation, the model also makes a prediction about the subsequent task states and the human user’s next action. If there is po-

tential danger (e.g., being burnt), the model can alert the user that there is a possibility of spilling hot water.

Tea-Making Task Data Set

Data showing action sequences of 8 healthy participants (N=8) making tea in a real-life setting were collected in the motion laboratory of the Crona Clinics Tübingen.

The experimental setup of the tea-making task is shown in Figure 2. In front of the participant, there is a kettle, a box with tea bags, and a cup for making and drinking tea. For time purpose the water in the kettle is cold but is assumed to be boiled already. In addition, participants can also choose to put sugar in the tea from a shaker or use a spoon if they choose to.

The participants were instructed to make tea in any order they wish to and the task ends with them drinking the tea. Each participant completed the task ten times and the positions of the objects on the table were changed between the repetitions.

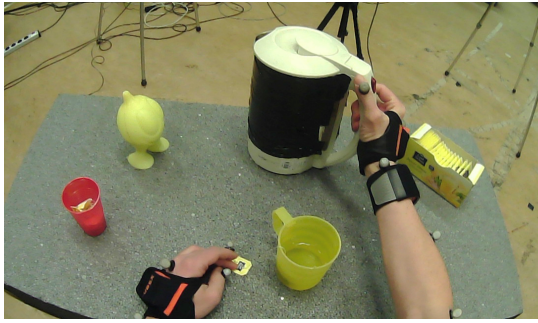


Figure 2: Tea-making task setup shown from the egocentric camera perspective.

Although the full tea-making process was recorded in video format for each trial, for our simple observer model in this paper, we only used the transcribed data, i.e., action sequences, of one trial as a guide to construct our observer model for the tea-making task. For the basic version of the model, we also simplified the action sequences by keeping only the basic actions by ignoring processes such as adding sugar into the tea. We also categorize each action in the sequence into its corresponding tea-making phase (see Figure 1 for our specified phases). One example of such an action sequence is shown in Table 1.

An ACT-R Based Observer Model for the Tea-Making Task

We take the first step towards a fully situation aware and anticipatory model by constructing an ACT-R based simple observer model to achieve a shared representation between the robotic agent (whose cognitive layer is represented by our model) and the human user.

In this model, we assume three critical phases: boiling water, putting water in the mug, and putting tea in the mug.

Step	Action	Phase
1	step near	start
2	pick up kettle	waterInMug
3	pour water in cup	waterInMug
4	put away kettle	waterInMug
5	take out tea bag	teaInMug
6	put tea bag in cup	teaInMug
7	pick up spoon	mixTea
8	mix tea	mixTea
9	put away spoon	mixTea
10	pick up cup	not categorized
11	drink tea	complete

Table 1: One transcribed and simplified trial of a participant completing the tea-making task. Each action corresponds to a pre-defined phase in the tea-making task and some actions (e.g., mixing tea) do not belong to any of the "critical" phases.

When all three critical phases are achieved, the task is completed. To match our data, we also assume that the boiling water phase is already completed at the beginning of the task.

Our observer model has several main components in its *declarative memory*: *task knowledge* includes chunks that represent action sequences in their corresponding phases. Each action is also combined with the object that affords the action. In addition, *input data* representing the action sequences are stored as chunks that link each action and object pair with the subsequent action object pair—this structure of input data provides us with a method to iterate through the action sequence in a trial, allowing the model to update task states at each action, achieving a certain level of situation awareness. Lastly, the ability of *situation awareness* is represented as a chunk that keeps track of the current task phase, the state of each critical phase, as well as the model's internal state indicating whether task states have been updated or not.

The simple observer model has seven productions and uses the declarative memory module, the goal module, the imaginal module and their corresponding buffers (retrieval buffer, goal buffer, and imaginal buffer).

When the model starts with the *step near* action, the chunk keeping information directly related to *situation awareness* is created in the imaginal buffer—and gets updated through out one task trial.

After starting, the model iterates through the actions from the input data. For each step of the data, *situation awareness* chunk is updated in the imaginal buffer. Given the current state suggested by the chunk in the imaginal buffer, the model retrieves a chunk that represents the current action-object pair, with information on the next action-object pair in the current phase, making a prediction (of the next action-object pair).

With such simple declarative memory and only seven productions, our model is able to observe one example of the tea-making process, keep its awareness of the situation through out the process, update the state of the task situation at each

step, and make predictions of the upcoming action. We will discuss this result further and our future steps in the next section.

Discussion

In this paper, we present an ACT-R based observer model for a tea-making task as a use case for real-life human-robot interaction.

Although our model is simple at the moment, it already achieves a certain level of situation awareness and is able to maintain an constant update of the state of the task with only basic knowledge of the task. The model's main component, a chunk representing its ability of situation awareness, keeps track of whether critical phases of the task are completed or not. When a non-essential action such as picking up a spoon happens, the model recognizes the corresponding *mixTea* phase but does not update the critical phase.

However, there are several exceptions that this model yet needs to handle. One example is that when an action does not belong to any existing phase (e.g., picking up a cup), the model stops due to its failure to retrieve a chunk containing information about this new action. Such actions may be handled with additional context/data-driven bottom-up productions. Another example is that the situation awareness chunk in the imaginal buffer lacks a corresponding slot indicating that danger may appear. Therefore, while our model can identify the correct tea-making phase and predict the next corresponding action, it does not yet provide an alert as an output.

Based on our current result and the identified model exceptions, we have a few immediate future goals for our model development. First, we will make the observer model more flexible for handling the situations stated above and be able to produce alert to the human user. Second, we plan to test the model systematically with the full tea-making data set in order to achieve a result where the model can predict most actions correctly while maintaining an accurate understanding of the task state. This would indicate our model's ability of keeping aware of the situation and adapt to individual users flexibly.

Going forward, we also aim to explore the advantages of the ACT-R model further for human-robot interaction. Particularly, its structure brings potentials for interfacing the ACT-R with other models representing the human's motor control system, unifying cognitive and physical human-robot interactions. In addition, such a structured model provides us with a possibility for situation awareness to be online-updated with multi-modal sensor data from the human user, e.g., eye-tracking, motion capture data, etc., leading to more precise support for the human user. (Hao et al., under review)

By starting with such a simple model, we believe that we show the potential of using structured representation of the collaborative task for achieving situation awareness and anticipatory proactive behaviors for assistive robotic systems that aid human users in the daily life.

Data & Material Availability

The current ACT-R model containing one tea-making process based on real-data is available at https://osf.io/tv7da/?view_only=c313235ae6314ecfbe8f6d3327aa153f.

Acknowledgments

This research was supported by the Volkswagen-Foundation (Az. 9B 007). Sponsors played no role in the conduct of the research or the preparation of the article or the decision to submit the article for publication.

References

- Amor, H. B., Neumann, G., Kamthe, S., Kroemer, O., & Peters, J. (2014). Interaction primitives for human-robot co-operation tasks. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2831–2837).
- Anderson, J. R. (2009). *How can the human mind occur in the physical universe?* Oxford University Press.
- Bartneck, C., Belpaeme, T., Eyssel, F., Kanda, T., Keijsers, M., & Šabanović, S. (2020). *Human-robot interaction: An introduction*. Cambridge University Press.
- Berke, M., & Jara-Ettinger, J. (2021). Thinking about thinking through inverse reasoning.
- Borst, J., Bulling, A., Gonzalez, C., & Russwinkel, N. (2022). Anticipatory Human-Machine Interaction (Dagstuhl Seminar 22202). *Dagstuhl Reports*, 12(5), 131–169. doi: 10.4230/DagRep.12.5.131
- Castrillo-Fraile, V., Peña, E. C., y Galán, J. M. T. G., Delgado-López, P. D., Collazo, C., & Cubo, E. (2019, dec). Tremor control devices for essential tremor: A systematic literature review. *Tremor and Other Hyperkinetic Movements*, 9(0). doi: 10.5334/tohm.511
- Endsley, M. R. (2015). Situation awareness misconceptions and misunderstandings. *Journal of Cognitive Engineering and Decision Making*, 9(1), 4–32.
- Friston, K., & Kiebel, S. (2009). Predictive coding under the free-energy principle. *Philosophical transactions of the Royal Society B: Biological sciences*, 364(1521), 1211–1221.
- Fromme, N. P., Camenzind, M., Riener, R., & Rossi, R. (2019). Need for mechanically and ergonomically enhanced tremor-suppression orthoses for the upper limb: a systematic review. *Journal of NeuroEngineering and Rehabilitation*, 16(1), 93. doi: 10.1186/s12984-019-0543-7
- Fu, W.-T., Bothell, D., Douglass, S., Haimson, C., Sohn, M.-H., & Anderson, J. (2006). Toward a real-time model-based training system. *Interacting with Computers*, 18(6), 1215–1241.
- Gonzalez, C., & Dutt, V. (2011). Instance-based learning: integrating sampling and repeated decisions from experience. *Psychological review*, 118(4), 523.
- Hao, C., Russwinkel, N., Haeufle, F., Daniel, & Beckerle, P. (under review). Modelling the individual by integrat-

- ing cognitive and physical aspects for human-robot interaction.
- Ho, M. K., & Griffiths, T. L. (2022). Cognitive science as a source of forward and inverse models of human decisions for robotics and control. *Annual Review of Control, Robotics, and Autonomous Systems*, 5, 33–53.
- Jara-Ettinger, J. (2019). Theory of mind as inverse reinforcement learning. *Current Opinion in Behavioral Sciences*, 29. doi: 10.1016/j.cobeha.2019.04.010
- Johnson-Laird, P. N. (2010). Mental models and human reasoning. *Proceedings of the National Academy of Sciences*, 107(43), 18243–18250.
- Klein, G., Snowden, D., & Pin, C. L. (2011). Anticipatory thinking. In *Informed by knowledge* (pp. 249–260). Psychology Press.
- Newell, A. (1994). *Unified theories of cognition*. Harvard University Press.
- Rabinowitz, N., Perbet, F., Song, F., Zhang, C., Eslami, S. A., & Botvinick, M. (2018). Machine theory of mind. In *International conference on machine learning* (pp. 4218–4227).
- Wickens, C. D. (2015). Noticing events in the visual workplace: The seev and nseev models.

GPT-Jass : A Text-to-model Pipeline for ACT-R Models

Anthony M Harrison (anthony.harrison@nrl.navy.mil)

Laura M. Hiatt (laura.hiatt@nrl.navy.mil)

Greg Trafton (greg.trafton@nrl.navy.mil)

U.S. Naval Research Laboratory
4555 Overlook Ave., SW
Washington, DC 20375

Abstract

The GPT-family of Large Language Models has garnered significant attention in the past year. Its ability to digest natural language has opened up previously unsolvable natural language problem domains. We tasked GPT-3 with generating complex cognitive models from plain text instructions. The quality of the generated models is dependent upon the quality and quantity of fine-tuning samples, but is otherwise quite promising, producing executable and correct models in four of six task areas.

Introduction

Since their initial development, transformers (Vaswani et al., 2017) have shown incredible promise in natural language domains, producing a new class of model, large language models (LLM). The most visible of these LLMs, OpenAI's ChatGPT (Brown et al., 2020) has garnered incredible interest from business and researchers alike. These LLMs are able to generate novel, grammatically and semantically correct prose from limited inputs. This enables diverse tasks from summarization and translation to debugging, and code generation (Zong & Krishnamachari, 2022).

Code generation, the process of creating functional code from textual descriptions, is a special case of translation where the target is a written programming language instead of a spoken language. Given a large enough corpus of quality and well-commented code (say from GitHub.com), transformers can be trained to output novel code snippets, functions, or entire programs. The correctness and quality of the code is often an issue (Azerbayev, Ni, Schoelkopf, & Radev, 2023), but it still represents an incredible achievement. Given the input requirements it should come as no surprise that such pipelines are limited to the most popular programming languages (e.g. Python, JavaScript, etc.).

One common feature of the LLMs is that they are pre-trained on a corpus and then fine-tuned to their application domain. This fine-tuning process is faster and cheaper than full-training, enabling the LLMs to be customized from their base, pre-trained forms. This fine-tuning process has enabled LLMs trained in one programming language to generate code in a different, but related language (Azerbayev et al., 2023). We applied this fine-tuning process in order to produce executable cognitive models from plain text instructions.

Implementation Considerations

GPT-3 Fine-tuning At the time of this research, only the general GPT-3 model was available for fine-tuning, however, it too is capable of generating code. The fine-tuning data should consist of tens of thousands, if not hundreds of thousands, of comment-code pairs. This data requirement, while cheaper and more accessible than training from scratch, still represents a significant challenge.

Model Availability Given the volume of data required for fine-tuning, even the totality of cognitive models published by the ACT-R community (Kotseruba & Tsotsos, 2020) would be insufficient by a few orders of magnitude.

These considerations would seem to exclude the possibility of a LLM supporting ACT-R, or any cognitive modeling language for that matter. However, these issues only arise if we rely upon the *existing* code-base of ACT-R models; we could instead choose to *generate* an entirely new code-base that meets our training requirements.

Implementation

Jass Instead of generating full Lisp ACT-R models, we chose to generate models using a higher-level modeling language, Jass (Harrison, 2020). Jass aims to simplify cognitive modeling by providing an imperative programming interface that compiles directly to ACT-R productions. The Jass models are significantly smaller and simpler than the production sequences that they represent¹.

Jass's language toolkit includes not only a parser and compiler, but also code generation capabilities. This allows us to programmatically create, edit, and generate Jass models. This textual manipulation enables us to parameterize the creation of new models based on existing templates.

Model Templates Thirty different models were written in Jass. These models were all of simple UI tasks (from clicking the mouse to conditionally selecting a button) situated within an abstract computer task (i.e., all actions were of GUI interactions). The simple UI tasks fit within six different, increasingly difficult, categories. The categories and a sample of each are listed in figure 1. For each model, three to ten different but consistent comments were written. These comments were then variablized by replacing subjects, objects,

¹ See (Harrison, 2020) for sample code

Table 1: UI Task Coverage

UI Task	Example
Keyboard	Press the ENTER key.
Search and Select	Select the blue button.
Simple Conditional	If you see a yellow sign, abort.
Search Between	Select the yellow sign between the car and the parking space.
While Loop	While monitoring is true, look for an empty parking spot and right click it.
Composite	While the car is not black, look for a yellow sign between the blue car and the green parking spot. If there is one, click it.

verbs, and adjectives within each description, effectively creating Madlibs™ for fine-tuning. By varying the label sets, we were able to generate twenty thousand different model-comment pairs for fine-tuning.

Fine-tuning The generated Jass dataset was passed through OpenAI’s Davinci GPT-3 model for four epochs of fine-tuning (the recommended number of epochs for fine-tuning).

Evaluation

We evaluated the performance of the system by giving it novel task descriptions for simple UI tasks embodied within a computer-based task environment. Six task comments were written by three different individuals (other than the primary author) for a total of eighteen comments to generate models from. Each was examined to ensure that the training set included nothing identical. For each comment, a model was generated and it was evaluated based on three incrementally harder conditions. First, is the code syntactically correct? Second, is the code actually executable? Finally, does it have the intended consequences when run?

Results

The results of the evaluation are listed in table 2. Each cell represents the percentage of the instructions for that task that passed the associated test. As can be seen, GPT-3 does well at generating syntactically correct code. This code not only compiles but conforms to the stylistic patterns (e.g. consistent use of case) that it was trained upon. Of the models that ran, all of them produced the correct behavior in the RESCHU task environment. However, for the last two task categories (*while-loop* & *composite*), GPT-3 was unable to produce executable code, which was absolute gibberish.

Discussion

Generally speaking, GPT’s ability to adapt to relatively little data through fine-tuning shows great potential and opens the door to many natural language processing problems that just

Table 2: Results

UI Task	Compile	Run	Correct
Keyboard	100%	100%	100%
Search and Select	100%	100%	100%
Simple Conditional	100%	100%	100%
Search Between	100%	100%	100%
While Loop	100%	0%	0%
Composite	100%	0%	0%

couldn’t be solved using traditional methods. GPT’s competence on any one of the UI tasks was directly the result of its coverage in the training data. Specifically, the simpler tasks (*keyboard manipulation*, *search and select*, and *simple conditionals*) were over-represented since they are also present in the more complex tasks (*while-loop* & *composite*). Those same complex tasks were under-represented; there simply weren’t enough examples to extrapolate any form of composition in the model.

It is worth iterating that this was the default LLM from OpenAI, not the source-code specific version, Codex. Should OpenAI open Codex to fine-tuning, we expect that the system would be better able to handle composition and other high-level constructs.

The volume of data required for training LLMs or even fine-tuning them, presents a significant barrier of entry, preventing us from applying the LLMs to more niche problems. Parametric generation of textual inputs seems to be a viable reconciliation of this problem, at least for fine-tuning.

Acknowledgments

This work was supported by ONR under funding document N0001420WX00496 awarded to Dr. Laura Hiatt. The views and conclusions contained in this document should not be interpreted as necessarily representing the official policies of the U.S. Navy.

References

- Azerbayev, Z., Ni, A., Schoelkopf, H., & Radev, D. (2023). *Explicit knowledge transfer for weakly-supervised code generation*.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodeli, D. (2020). Language models are few-shot learners. *CoRR*.
- Harrison, A. M. (2020). An imperative alternative to productions for act-r. In *Proceedings of the international conference on cognitive modelling, ICCM 2020*.
- Kotseruba, I., & Tsotsos, J. K. (2020). 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*, 53(1), 17–94.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). *Attention is all you need*.
- Zong, M., & Krishnamachari, B. (2022). *a survey on gpt-3*.

Cognitive Modelling of Intention Recognition in Cocktail Mixing

Linda Heimisch¹
Janice Jansen¹
Nele Russwinkel²

¹ Technische Universität Berlin, Department of Psychology and Ergonomics, Berlin, Germany

² Universität zu Lübeck, Institute of Information Systems, Lübeck, Germany

Abstract

Recognising the intention of a human partner is a key challenge for collaborative systems in human-robot interaction. However, existing studies of intention recognition abilities in AI system mostly focus on data-driven approaches and the recognition of direct action intentions (low-level intentions). We propose an artificial intention recognition approach that is implemented as a cognitive model in the theory-based ACT-R architecture and that infers superordinate action goals (high-level goals). We tested our approach for the recognition of cocktails from mixing sequences performed by human participants in an experimental study. Intention recognition speed of the model was evaluated and compared to human intention recognition performance. Our results indicate that the implemented model successfully recognises high-level intentions and tends to be substantially faster than humans.

Keywords: Human-robot interaction; intention recognition; cognitive modelling; ACT-R

The Challenge of Intention Recognition in Human-Robot Interaction

For AI systems that are designed to interact with a human partner in real-world scenarios it is of essential importance that the system can anticipate actions of the human to adapt to them accordingly. This requirement has been described under the term *intention recognition* in an increasing branch of human-robot interaction (HRI) research. The ability to infer a partner's intention has been identified to be a key factor for the performance of collaborative AI systems (Smith, Belle, & Petrick, 2022). Collaborative system behaviour, which is defined by both human and AI partner working on one task at the same time and in the same place (as opposed to cooperative behaviour where both work on different (sub-)tasks) (Bauer et al., 2016), poses high demands on a system's capability to recognise a human's next action fast and reliably.

However, in HRI settings, the robot's behaviour is still often pre-programmed (Angleraud et al., 2021). Attempts have been made to enhance the flexibility of AI systems with intention recognition abilities, using unsupervised machine learning (Vinanzi, Cangelosi, & Goerick, 2021), probabilistic models (Luo & Mai, 2019), or Gaussian Mixture Models (Duarte et al., 2018). Such approaches showed promising anticipative performances, while are still lacking important criteria for interactive AI systems. According to Kambhampati (2020), it is crucial that such systems are

explicable, meaning that a human partner can comprehend them. Further, data-driven approaches pose the challenge that they always need pre-existing large data sets they can be trained with, which often also bears the problem that they can't be transferred to other situations.

For this reason, we propose an artificial intention recognition approach based on a cognitive architecture. Cognitive architectures offer a cognitive plausible framework and can make explicit predictions from abstract cognitive-psychological theories (Brasoveanu & Dotlačil, 2020). Thereby, cognitive models that are developed within a cognitive architecture provide verbalisable, explicable process descriptions, which are furthermore based on a cognitive theory instead of data sets and thereby enable transferability to similar, but new situations.

Another important aspect is the level of intentions that shall be recognised by an AI system. In previous work on intention recognition in HRI contexts, distinctions were made between different levels of intentions, where intentions on a lower level serve to realise intentions on a higher level (Howard & Cambria, 2013). Gomez Cubero & Rehm (2021) define low-level intentions as direct action intentions and high-level intentions as the superordinate goals of actions. For example, a person grabs a key (low-level intention) to unlock a door (high-level intention). Fulfilling a high-level intention can involve the necessity of several low-level intentions to be carried out. For our work, we adopt the distinction by Gomez Cubero & Rehm (2021) and distinguish between low-level and high-level intentions. Previous studies exist, e.g. those by Gomez Cubero & Rehm (2021) and (Duarte et al., 2018), that have successfully fed eye-tracking data into artificial systems to enable them to infer the low-level intentions of humans.

However, more work is needed on artificial high-level intention recognition capabilities. Further, we believe that the recognition of high-level intentions is of especial importance for a collaborative AI system since it will typically have more time to adapt and react to high-level than to low-level actions.

We therefore propose a high-level intention recognition approach in a cognitive architecture, with the aim to improve human-robot collaboration in variable real-world scenarios.

Cognitive Modelling with ACT-R

ACT-R (Anderson et al., 2004) is a well-established cognitive architecture that allows for executable models, that is, direct simulations of human task solving behaviour. Its

framework is set by rules and constraints that are based on the current state of cognitive psychological research. ACT-R model runs result in predictions about behavioural markers, e.g. reaction times or error rates.

A core characteristic of ACT-R is its modular structure, where modules interact through their corresponding buffers and different modules represent specific brain areas. Related to this is a sharp distinction between declarative and procedural knowledge, each implemented into a different module. Declarative knowledge units, either in the form of pre-known knowledge that a model is assumed to be equipped with or in the form of new facts learned during the task, are represented as so-called chunks. Chunks consist of a unique chunk name, a chunk type and a certain number of slots which can but need not contain specific values. Number and names of slots that a chunk has are defined by its chunk type. Procedural knowledge is represented as so-called productions. Productions in ACT-R are divided into a condition part and an action part, where the condition part describes the states of certain buffers or chunks that they contain, and the action part describes requests or changes to certain buffers or to their content. A model run in ACT-R is a chain of *match-select-apply cycles*, where in each cycle one production whose condition part *matches* the current state of the model is *selected* and subsequently, its action side is *applied*.

Another important feature of ACT-R is its inherent combination of symbolic and sub-symbolic processing, which defines it as a hybrid architecture (Kotseruba & Tsotsos, 2016). The symbolic processing of an ACT-R model happens on the level of chunks and productions. Sub-symbolic processing, which can optionally be enabled, refers to additional calculations involving adjustable parameters that influence symbolic processing. For instance, activation-based sub-symbolic processing mechanisms can influence the probabilities of certain chunks to be retrieved from declarative memory, while utility-based sub-symbolic processing mechanisms can influence the probabilities of certain productions to be selected.

The combination of symbolic processing on the level of chunks and productions, which can at any time during task solving be described in an explicit verbal manner, and sub-symbolic processing, which allows for the modelling of complex learning and context effects, make the ACT-R architecture a promising candidate for modelling an artificial intention recogniser that fulfils the above-mentioned criteria of flexibility, explicability, and transferability.

Research Goals and Requirements for the Experimental Study

In the present work, we address the question whether high-level intention recognition can successfully be modelled in the cognitive architecture ACT-R. As a secondary research question, we investigate whether such a model of artificial intention recognition resembles human intention recognition, that is, whether the ACT-R model identifies intentions with similar speed as humans.

Since we imbed our research into the context of collaborative human-robot interaction, our research goal makes several demands on the exemplary task that the artificial intention recogniser is tested with. The primary criterion for the exemplary task shall be that it resembles a real-world scenario, at best a daily activity that can be a realistic test case for assistive robots. This criterion holds for preparation tasks like food or drink preparation. These tasks typically require humans to fulfil a number of low-level actions, e.g. selections of ingredients and their combinations, in order to reach a high-level goal, e.g. a dish or a drink. Further, the context of our research goal demands that the high-level intention to be recognised is chosen by a human, with the possible options being known to the artificial system. Also, a realistic test case demands that high-level intention recognition cannot solely rely on the stepwise comparison of executed low-level actions with pre-known unambiguous action instructions. Rather, realistic real-world preparation tasks typically involve some degree of overlap in the low-level actions to take. Also, realistically, the preparations should involve both sequences of fixed sequential steps (*fixed sequences*) as well as sequences of freely ordered steps (*free sequences*).

These criteria hold for the task of cocktail mixing. The preparation of cocktails by recipe resembles many daily preparation tasks and initially requires the choice of a certain cocktail among a given selection. Moreover, ingredients for different cocktails typically overlap to a certain degree. Further, cocktail recipes involve both fixed sequences and free sequences. For instance, mixing any cocktail requires to take a vessel before any alcohol or juice can be inserted, but whether alcohol or juice is added first usually does not matter.

We therefore chose to test artificial intention recognition for the task of cocktail mixing. The task for the artificial intention recogniser, implemented as an ACT-R model, is to infer from the incremental combination of ingredients (low-level actions) which cocktail is being mixed (high-level intention).

To evaluate the performance of the artificial intention recogniser, human cocktail mixing actions are needed. We gathered this data in a lab study with a virtual “cocktailbar” where participants were tasked with preparing cocktails from a given selection. To compare artificial with human intention recognition speed, a follow-up task for participants in the lab study was to recognise which cocktail was being prepared by another person shown in a pre-recorded video sequence.

Experimental Study

We conducted an experimental study based on a touch-sensitive smartboard. Participants were guided through a virtual “cocktailbar” and were instructed to “mix” five cocktails of their choice out of a selection of eleven cocktails. The cocktails consisted of between five and ten ingredients, where each recipe involved fixed sequences (e.g. a glass has to be taken before ice cubes can be inserted), and free sequences (e.g. a straw can be added before a decorative lime, or vice versa). The cocktails shared differing numbers of

ingredients, resulting in varying degrees of overlap between the recipes.

In each of the five iterations, participants first selected a cocktail, then saw the corresponding recipe, including all necessary ingredients and the information which parts of the preparation were in fixed or free order, which they were instructed to memorise. Subsequently, the screen turned into the “cocktailbar mode” where participants saw all ingredients of all possible cocktails. Their instruction was to “mix” the cocktail of their choice according to the memorised recipe, which meant to select the ingredients in the correct order via touch, where the order of ingredients in the free sequences was up to the participant’s decision. Ingredients were presented as graphical drawings. Whenever a correct ingredient was touched, participants saw a graphical drawing version of the cocktail proceed. Since we wanted to limit our intention recognition paradigm to correct sequences without errors, incorrect ingredients (whether completely incorrect for the chosen cocktail, or just in the wrong order) could not be selected, meaning that participants could not make errors. After the last correct ingredient was added, participants saw their ready cocktail as a complete graphical drawing. Figure 1 shows the experimental screen in the “cocktailbar mode”.



Figure 1: Experimental screen in the “cocktailbar mode”. Participants “mixed” a cocktail based on the memorised recipe by touching the ingredients in order.

For each participant, the order of selected ingredients for each chosen cocktail was recorded. In addition, eyetracking data was recorded but not further processed in this study. The eyetracking required participants to stand in front of the smartboard screen with approximately one meter distance, which made it necessary for participants to touch the smartboard using a long stick.

The recorded behavioural data was used to later test the ACT-R model’s ability to recognise high-level intentions, i.e. which cocktail a participant is about to mix in a given trial. To test human intention-recognition ability for the same task, participants went through a second experimental phase after their fifth cocktail was completed. They were shown pre-recorded video sequences of another person mixing those five cocktails they had just created, but in randomised order. For each cocktail, one video was created, with the ingredients in the free sequences being selected in a random order.

Participants were instructed to say aloud which cocktail was being created in the video as soon as they were confident to recognise it. The step where participants correctly recognised the cocktail was manually noted.

20 participants were acquired among students at Technische Universität Berlin and via advertisements (12 female), with a mean age of 25.65 years ($SD = 1.74$), ranging from 23 to 29 years. All participants signed an informed consent. The eye gaze recording required to exclude wearers of glasses. Where applicable, participants were compensated with course credit.

ACT-R Model

For building the ACT-R model, two major criteria had to be met: (i) The model must be able to “observe” the human cocktail mixing, that is, the steps that are taken. (ii) It must have pre-knowledge about the cocktail recipes.

Regarding the challenges for high-level intention recognition in our cocktail mixing paradigm, the model needed to have the following abilities: (1) It must be able to recognise which cocktail is being built out of a selection of eleven cocktails, starting with zero prior information. (2) It must be tolerant to the ambiguity resulting from the overlapping ingredients between cocktails. (3) It must be able to include information from both fixed and free sequences into its recognition process.

Our implementations of these criteria and requirements are outlined in the following. Importantly, since our primary research goal was to investigate the implementation of high-level intention recognition in ACT-R and the post-hoc comparison with human data was only secondary, psychological plausibility was not our focus for the usage of ACT-R mechanisms.

Criterion (i): Information about the human cocktail mixing sequences was provided to the model in the form of pre-known knowledge in declarative memory. Each of these chunks, in the following referred to as *trial chunks*, represented either one experimental trial, i.e. one cocktail mixing sequence by a human participant, or one cocktail mixing sequence as shown in the pre-recorded video sequences. The performed action sequence was encoded in the form of values, standing for the ingredients, assigned to numbered slots. At the beginning of each model run, one trial chunk was placed into the goal buffer.

Criterion (ii): Information about the eleven cocktail recipes was equally provided to the model in the form of pre-known knowledge in declarative memory. Each of these chunks, in the following referred to as *recipe chunks*, represented one cocktail recipe. Analogously to the trial chunks, the recipe chunks contained numbered slots filled with values representing the required ingredients in the correct order. Since slots must be listed sequentially, free sequences in the recipes were treated as fixed sequences listed in a random order.

Ability (1): For a realistic modelling of high-level intention recognition, the model must start at the beginning of each run without any prior information about the high-

level goal, but it must incrementally be provided with information about which action was taken, i.e. which ingredient was added, by the human participant. Therefore, the imaginal buffer was chosen for the model’s information maintenance and information updating. At the beginning of each model run, the imaginal buffer was empty. The first production to fire requested the imaginal buffer to create and hold one chunk of type memory, in the following referred to as *memory chunk*. The memory chunks had the same structure as the trial chunks, but with the numbered slots, representing the performed actions, being empty. Then, iterative cycles followed, each represented by one production. In each of these productions, the next action in the human’s mixing sequence, that is, the ingredient value in the next slot of the trial chunk in the goal buffer, was placed into the next slot in the memory chunk in the imaginal buffer. Thereby, the model’s information base was built up incrementally. The model could use this permanently extending representation of its “observation” for stepwise comparisons with its knowledge about the eleven cocktail recipes.

Ability (2) and ability (3): Stepwise comparisons between the information held in the imaginal buffer and the recipe chunks in declarative memory were made using specific retrieval requests, that is, requests to the retrieval buffer to search for a recipe chunk in declarative memory whose slot values matched with those currently contained in the memory chunk. When such a recipe chunk was found, the model output its guess about which cocktail was being mixed. However, the overlapping cocktail recipes and the free sequences in the recipes, resulting in several possible action sequences for one cocktail, caused ambiguity for the model that did not allow for simple stepwise comparisons. Therefore, the sub-symbolic mechanism of spreading activation was added to the model. This mechanism influences retrieval requests such that activation affects which chunk is retrieved, rather than specific slot value matches between the retrieval request and the chunks in declarative memory. We set the imaginal buffer as source for spreading activation, whereby at every retrieval request, activation spread to those chunks in declarative memory that contained the values that were hold in the chunk in the imaginal buffer at that point of time. Importantly, spreading activation is unaffected by the specific slot assignments. Therefore, retrieval requests without any slot assignments can be made and it is irrelevant which slot of a chunk in declarative memory contains a relevant value. Consequently, the model’s retrieval requests were completely blind to fixed or free sequences since the order of ingredients did not matter in the stepwise comparisons between the memory chunk and the recipe chunks. Hence, the holistic instead of order-sensitive nature of spreading activation-based retrieval requests ensured that the model had ability (3). Regarding ability (2), it was the cumulative character of spreading activation that made the model robust to slot value overlapping between different recipe chunks: The activation that a chunk in declarative memory receives increases with

the number of values it shares with the chunk in the source buffer, which means that the chunk in declarative memory that shares the most values with the chunk in the source buffer receives the most activation and, as a consequence, will be retrieved.

One model run can be summarised as follows: A model run started with the placement of one trial chunk into the goal buffer and the creation of one memory chunk in the imaginal buffer. In the following iterations, i.e. production firings, continuous retrieval requests to declarative memory were made that were based on the iteratively growing information in the memory chunk in the retrieval buffer, mediated through the spreading activation mechanism. In each iterative cycle, the recipe chunk with the highest activation was retrieved and additionally output by the model, representing its current guess about the high-level intention. A model run stopped when all information from the trial chunk had been provided to the memory chunk, i.e. the number of iterations equalled the number of actions taken by the human participant in the respective trial, which in turn equalled the number of ingredients of the cocktail that was being mixed.

The structure and function of the ACT-R model is depicted in Figure 2.

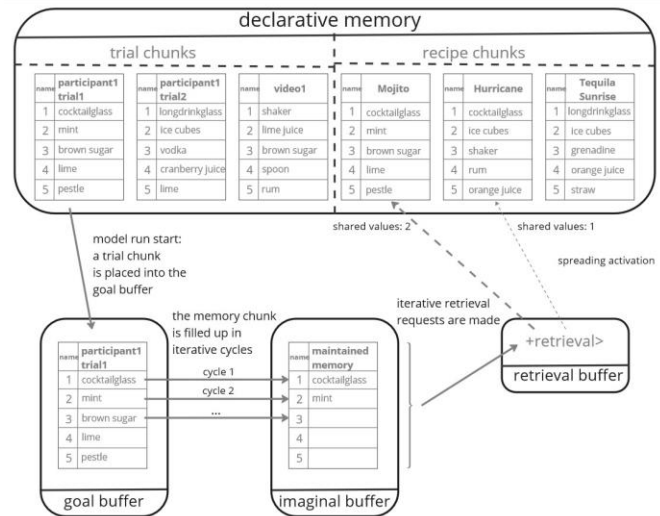


Figure 2: Structure and function of the ACT-R model. Arrows show actions carried out in the production firings. Declarative memory content and chunks are shortened for illustrative purposes. In declarative memory, the two different kinds of trial chunks are depicted as examples.

The model’s intention recognition performance was measured in two tasks: First, the model was tasked with recognising the human participants’ intentions based on their cocktail mixing sequences. For this, one model run was simulated for each of the trial chunks representing participant cocktail mixing sequences. Second, since the human participants inferred intentions not from other participants’ action sequences but from the pre-recorded videos, the model

was also tasked with recognising intentions based on the cocktail mixing sequences shown in the eleven pre-recorded video sequences. For this, each video was “shown” to the model the total number of times the according cocktail had been chosen by the human participants, meaning that the according number of model runs was simulated for each of the trial chunks representing video mixing sequences.

Results

Speed of intention recognition was operationalised as the step number in the mixing sequence where the correct cocktail was identified by a human participant or by the ACT-R

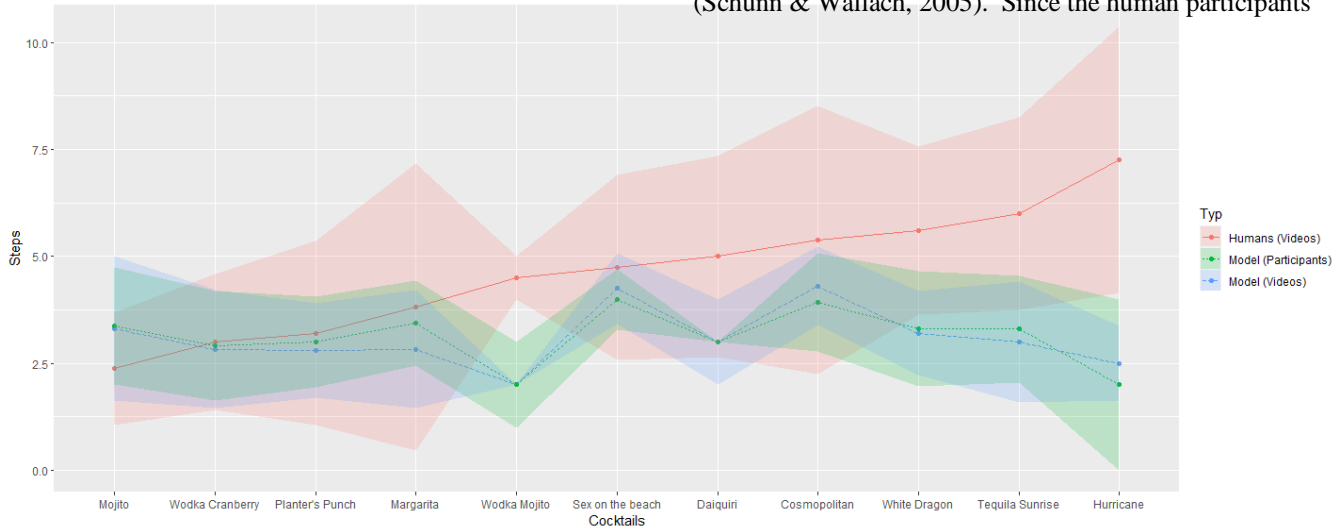


Figure 3: Mean intention recognition speeds and standard deviations for *Human(Videos)*, *Model(Participants)* and *Model(Videos)*.

model, respectively. Since our primary research goal was to analyse whether high-level intention recognition can successfully be modelled in ACT-R, we were mainly interested in an investigation of the model’s performance and its variation between different high-level goals, i.e. between the different cocktails. In a second step, we investigated the comparison with the human performance.

Figure 3 shows a descriptive comparison between the averaged intention recognition speeds of human participants watching the videos (*HumanVideos*), the ACT-R “watching” the mixing sequences performed by the human participants (*ModelParticipants*), and the ACT-R model “watching” the mixing sequences in the videos (*ModelVideos*). The mean speeds were calculated separately for each cocktail to allow for the descriptive identification of differences between cocktails. Since the total number of times a cocktail was mixed by the human participants differed substantially between cocktails, with a minimum of two total choices for the Wodka Mojito and a maximum of 16 total choices for the Mojito and the Margarita, respectively, we decided against inferential statistics and statistical model estimation.

Except for the Mojito, the ACT-R model showed a faster mean intention recognition speed than the human participants for all cocktails, both when anticipating the cocktail from the mixing sequences performed by the human participants and when anticipating them from the videos. Apparently, for all cocktails, the mean intention recognition speed of the ACT-R model is very similar for both kinds of anticipations.

As a measure for the deviation of the model’s mean intention recognition speed from those of the human participants, we applied the root mean square deviation (RMSD) measure which is a common measure to evaluate model fit when models are compared to empirical data (Schunn & Wallach, 2005). Since the human participants

inferred intentions only from the video sequences and therefore there was no “human equivalent” to the data from *Model(Participants)*, RMSD was calculated only for the comparison between *Model(Videos)* and *Human(Videos)*. RMSD ranged between .17 for the Wodka Cranberry and 4.75 for the Hurricane, underpinning the huge variance between different cocktails.

Discussion

The artificial intention recognition approach implemented as a cognitive model in the ACT-R architecture is able to infer high-level intentions in the form of cocktails from a sequence of mixing actions taken by humans, that is, ingredients that are sequentially chosen. This result illustrates how symbolic and sub-symbolic processing mechanisms in ACT-R can be combined to achieve a system with incremental information growth and context-dependent, robust memory retrievals. Our approach extends previous research which has mainly focused on data-driven approaches and low-level intention recognition based on eye tracking data.

The ACT-R model proves to be robust against ambiguity resulting from overlapping cocktail recipes and from free sequences within recipes. The latter is underpinned by the model's similar intention recognition speeds for mixing sequences performed by the human participants and mixing sequences performed in pre-recorded videos, which is consistent for all cocktails. This indicates that our artificial intention recognition approach fulfils the criterion of flexibility in terms of robustness against varying input.

Regarding Kambhampati (2020)'s criterion of explicability, the function of the developed ACT-R model can explicitly be linked to its structure, which in turn can be described transparently. Thus, ACT-R models prove to be promising candidates for implementing high-level intention recognition into AI systems that are designed to interact with humans.

Concerning the criterion of transferability, we argue that the implemented ACT-R mechanisms, in particular the combination of symbolic processing mechanisms in the form of structured pre-knowledge and iterative production cycles, and sub-symbolic processing mechanisms in the form of spreading activation, are general enough to be transferable to other tasks or experimental paradigms. Most importantly, models in the theory-driven ACT-R architecture do not need to be trained with pre-existing data, which poses a practical advantage over data-driven approaches.

The model's intention recognition speed varies substantially between different cocktails. Future developments of our study will have to analytically investigate how the number of free sequences, and the number of ingredients within free sequences, in a cocktail recipe influences how fast the cocktail can be recognised by the model. Also, the degree of overlap between cocktail recipes will have to be considered as a factor for intention recognition speed.

Our results point to substantial differences between the model's and the human participants' intention recognition speed for most, albeit not all cocktails, strongly suggesting that the modelled intention recognition approach differs essentially from human high-level intention recognition. Since except for the Mojito, human participants were, on average, slower in correctly recognising the high-level intention, it can be assumed that the efficient use of memory mechanisms in the ACT-R models deviates from average human cognitive processing. For instance, humans can be assumed to have a much more error-prone maintenance of the observed information than the model which perfectly updates and maintains information in its imaginal buffer. Moreover, the model has equally complete memory for all eleven cocktail recipes, whereas human participants had been confronted with five cocktail recipes for a short time only and supposedly remember some recipes better than others, where relevant factors can be assumed to be extent of the recipe and previous experience with the cocktail, or cocktails in general.

Future studies will have to analyse the influence of factors such as structural differences between the possible high-level goals (here: different cocktails with their differing recipes) on

the intention recognition speed of an artificial system. Methodically, statistical model estimation, for instance in the form of linear mixed models, could be suitable to systematically analyse the influence of item differences on the model's performance. Likewise, the influence of item as well as participant differences on human intention recognition performance could be analysed using such methods.

Critically, our experimental paradigm prevented the possibility of making mistakes during cocktail mixing. However, robustness to human errors and violations of rules and instructions will be an important property of future assistive AI systems in real-world scenarios. Thus, future developments of our study will have to take errors into consideration and carefully evaluate the applicability of the demonstrated ACT-R mechanisms for high-level intention recognition in imperfect task executions.

References

- Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111(4), 1036–1060.
- Angleraud, A., Mehman Sefat, A., Netzev, M., & Pieters, R. (2021). Coordinating Shared Tasks in Human-Robot Collaboration by Commands. *Frontiers in Robotics and AI*, 8, 734548.
- Bauer, W., Bender, M., Braun, M., Rally, P., & Scholtz, O. (2016). Lightweight robots in manual assembly—best to start simply. *Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, Stuttgart*, 1.
- Brasoveanu, A., & Dotlačil, J. (2020). *Computational cognitive modeling and linguistic theory*. Springer Nature.
- Gomez Cubero, C., & Rehm, M. (2021). Intention Recognition in Human Robot Interaction Based on Eye Tracking. In *Human-Computer Interaction—INTERACT 2021: 18th IFIP TC 13 International Conference, Bari, Italy, August 30–September 3, 2021, Proceedings, Part III* (pp. 428–437).
- Duarte, N. F., Raković, M., Tasevski, J., Coco, M. I., Billard, A., & Santos-Victor, J. (2018). Action anticipation: Reading the intentions of humans and robots. *IEEE Robotics and Automation Letters*, 3(4), 4132–4139.
- Howard, N., & Cambria, E. (2013). Intention awareness: improving upon situation awareness in human-centric environments. *Human-centric Computing and Information Sciences*, 3(1), 1–17.
- Kambhampati, S. (2020). Challenges of human-aware ai systems: Aai presidential address. *AI Magazine*, 41(3), 3–17.
- Kotseruba, I., & Tsotsos, J. K. (2016). A review of 40 years of cognitive architecture research: Core cognitive abilities and practical applications. *arXiv preprint arXiv: 1610.08602*.
- Luo, R. C., & Mai, L. (2019). Human intention inference and on-line human hand motion prediction for human-robot collaboration. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp.

- 5958-5964). IEEE.
- Schunn, C. D., & Wallach, D. (2005). Evaluating goodness of-fit in comparison of models to data. *Psychologie der Kognition: Reden and Vorträge anlässlich der Emeritierung von Werner Tack*, 115-154.
- Smith, G. B., Belle, V., & Petrick, R. (2022). Intention Recognition With ProbLog. *Frontiers in Artificial Intelligence*, 75.
- Vinanzi, S., Cangelosi, A., & Goerick, C. (2021). The collaborative mind: intention reading and trust in human-robot interaction. *Isience*, 24(2), 102130.

A Straightforward Implementation of Sensorimotor Abstraction in a Two-Layer Architecture for Dynamic Decision-Making

Nils Heinrich^{1,3,*}, Annika Österdiekhoff², Stefan Kopp², Nele Russwinkel¹

¹ Human-Aware AI, Universität zu Lübeck, Lübeck, Germany

² Kognitive Systeme und Soziale Interaktion, Universität Bielefeld, Bielefeld, Germany

³ Kognitionspsychologie und Kognitive Ergonomie, Technische Universität Berlin, Berlin, Germany

* E-mail: heinrich@tu-berlin.de

Abstract

Cognitive and sensorimotor functions are usually assessed separately and therefore also modeled individually although they are strongly intertwined. One way to link these two levels conceptually is sensorimotor abstraction. It is the simplification of complex sensorimotor experiences, and it might enable goal-directed planning in situations with high uncertainties. We propose a computational model for dynamic decision-making that employs two distinct layers, a (lower) sensorimotor control layer holding sub-symbolic information, and a (higher) cognitive control layer holding abstracted information as symbols. In this two-layer architecture information about action control is passed upwards in the hierarchy, abstracted, and used to generate explicit action intentions which are passed downwards again. The hierarchization of model components is intended to represent the different levels of regulatory control (automated vs. fully conscious). We also use different forms of modeling for the individual layers. We employ predictive coding for sensorimotor and ACT-R for cognitive control. An agent equipped with the two-layer architecture is situated in a grid world and tasked to reach a finish line. However, the environment poses challenges on motor control by causing perturbations in the action execution of traversal reflecting varying uncertainty encountered in the real world. Here we describe a straightforward approach to the multi-layer architecture and relate it to the embodied cognition perspective. We also discuss possible extensions that we plan to introduce which depict fundamental cognitive functions such as representing the visual environment in varying granularity.

Keywords: Embodied cognition, dynamic decision-making, ACT-R, Sensorimotor abstraction

Introduction

The field of embodied cognition supports the assumption that higher cognitive functions are grounded in low-level sensorimotor experiences (Shapiro, 2019). Explicit computational models of cognition are less concerned with how sensorimotor grounding is intrinsic to cognitive processes (Pezzulo et al., 2011). The challenge in this is to implement theoretical descriptions using mathematical mechanisms. We present here a computational cognitive model for dynamic decision-making (Gonzalez et al., 2017)

appropriate for representing sensorimotor grounding, and specifically relates decision-making to the motor experience of control in addition to what is visually perceived. For this purpose, we define 2 different layers, which are embedded in a cognitive architecture, henceforth called two-layer architecture.

Humans show efficient problem-solving capabilities in highly dynamic environments. This may be based on the ability to reduce complex environments to simple representations of various modalities which are then used for goal-oriented behavior (Turner et al., 2019). This implies that humans do not act alone on what is perceived at that very moment but rather they represent the environment in terms of action possibilities between which a decision is made. In these representations, many things are combined. Not only environment-related properties themselves, but also properties that relate to the action within the environment are included. This could aid decision-making by having the decision process only consider represented options, in turn reducing the number of possibilities and thus reducing mental load. Where several possibilities that are similar to each other and share rough features are combined into one represented option. The shared features refer on the one hand to the changes in the environment brought about by this option, and on the other hand to the motor executions required to bring about these changes. The result is a simplification of the complex environment represented by its possibilities to be acted upon. However, implementing this in a computational model poses the difficulty of how to simulate spatial associations of abstract concepts (Pezzulo et al., 2011). We are currently in the process of developing the computational model and simulating behavior while situating it in a custom environment. Human behavioral data for a later comparison are already available.

Rationale for Hierarchical Structure

Representing the environment from the perspective of embodied cognition can be achieved through the process of *sensorimotor abstraction*. For this, we adopt the perspective of Eppe and colleagues (2022) to distinguish between action and state abstraction. Action abstractions refer to the summarization of many individual consecutive motor commands into a comprehensive movement primitive (Flash & Hochner, 2005; Schaal, 2006). The other relevant type of abstraction, state abstraction, refers to a simplification of the

environment into a representation that focuses on task-relevant features while giving less weight to task-irrelevant features. The two-layer architecture described here is intended to cover both forms of abstraction. In our model, these representations are depicted as explicit symbols on a high level of abstraction while sensorimotor experiences are depicted sub-symbolically. We put particular emphasis on modeling the construction of mental representations at the cognitive level from sub-symbolic information passed up by the sensorimotor level. We thereby postulate a hybrid model which combines symbols with mathematical mechanisms (Wang, 2017).

To test the assumptions that went into our two-layer architecture and to enable simulations, we situate the computational model in a 2-dimensional grid world by means of an agent. The grid world resembles a complex physical world which is the condition for *situated action* (Vera & Simon, 1993). It is a theoretical framework that emphasizes the importance of considering the direct (social and) physical contexts in which human behavior occurs. Here, in our architecture, the physical features of the world are experienced through sensorimotor functions and transformed into symbols.

We narrow down the sensorimotor experiences the two-layer architecture draws upon to two essential ones. It is able to visually perceive its surroundings. In this respect, the sensorimotor level has information about the exact position in pixels of objects in the world. The world, however, is not visually perceived as a whole, but only the immediate environment, which is determined by the size of the observation window. Furthermore, the two-layer architecture has access to information expressing the correctness of the realization of its motor commands. This information is extracted from the deviation between expected consequences in the environment by motor executions and the actual perceived consequences. It reflects the situated action control, termed *Sense of Control* (SoC; Pacherie, 2007), and plays a fundamental role within the two-layer architecture. The two layers within the architecture are a sensorimotor control layer implemented in Python (Van Rossum & Drake, 2009) and a cognitive control layer implemented in ACT-R (Anderson & Lebiere, 1998). Each layer contains different levels of control processes. The multilevel structure is intended to emphasize the abstraction from sub-symbols at the sensorimotor level to symbols at the cognitive level as information is passed up in the hierarchy.

Multi-layered Computational Model

Our two-layer architecture highlights the intersection between low-level sensorimotor and higher-level cognitive functions. Both the defined layers have to be distinct from another holding their individual functions for explicitness while nevertheless being strongly interconnected through active exchange of information. The result is a complex structure of predictive and postdictive processes with action planning and motor control components. Different levels of

regulatory control are deployed in the individual model layers. The sensorimotor layer applies automated regulatory control which is happening unconsciously. When applied, the agent can usually infer that something went wrong, but not exactly what. On the other hand, the cognitive control layer applies fully conscious regulatory control. In this the agent is completely aware of what went wrong. This is implemented through a fundamental aspect of the two-layer architecture, the SoC. It is the subjective feeling of whether the agent is in control of a current action execution (Pacherie, 2007). We defined two individual SoCs, one at the level of sensorimotor control and one at the level of cognitive control. The low-level Sense of Control (LL SoC) and the high-level Sense of Control (HL SoC) respectively. Both range from 0 (total loss of control) to 1 (being in full control). They are interconnected in the way that the HL SoC is only affected if LL SoC reaches a threshold given by the free parameter *Cognitive Control Layer (CCL) threshold*. This means that only large changes or several consecutive changes in the same direction and without recovery in LL SoC cause changes in HL SoC. This is supposed to reflect becoming aware of previously subconscious information. The transition from LL SoC to HL SoC is to depict state abstraction (Epe et al., 2022), where dynamics on control at the sensorimotor level are represented in a simplified way (summed up in our case) at the cognitive level. We argue the existence of two distinct SoCs embedded at different levels of the model structure with the hierarchization of intentions by Pacherie (2007). Originally the author postulates the existence of three levels of intentions (distal vs. proximal vs. motor), where each individual level refers to a different temporal scale and rigor in which aspects of the intention are specified. The aspects of the intention thereby comprise of the goal state of the intention. How strict this goal state is defined depends on the level of the intention. With increasing level of the intention in the hierarchy, the more distant in the future its realization is and the more abstract the intention is specified.

On the highest level of abstraction and time scale, Pacherie (2007) denotes *distal intentions*. They reflect general future objectives, that can be directed only a few moments or far into the future. Due to distal intentions being completely detached from the current situation and therefore from situated action, they are not of interest for this specific model structure. We therefore omit this level of intentions within the two-layer architecture.

Once distal intentions are passed downstream in the hierarchy, they become situated in the agent's current spatial and temporal environment. Now termed *proximal intentions*, at this level the temporal scale is defined less flexible compared to distal intentions, in the way in which one plans only a few moments into the future (Pacherie, 2007). We model these intentions at the level of cognitive control (CCL) as general *action goals* (Kahl et al., 2021). An action goal is a symbol of the consequences anticipated in the environment. An example would be the agent's action goal of having walked around his desk to the refrigerator in order to get the leftover donut from the day before. It is situated by taking

into account the agent's immediate environment (refrigerator at the other end of the office, with desk in the way) as well as the agent's motor repertoire (walking around the desk). The symbol of the action goal contains the final consequence (to be in the immediate vicinity of the refrigerator hopefully with a donut in hand), as well as a specification of the path to the final outcome (*around* the desk).

But the action goal is not the only possible way to act at that exact moment. In one fell swoop, we can think of several ways to reach the objective, of several action possibilities. Inspired by *affordances* which denote all possible relations of agent and direct environment (Gibson, 1966), we define the *action field*. We use this term to refer only to perceived affordances (Norman, 1988) that can be used to directly solve the objective at hand. For the specific objective of getting the leftover donut, the action field may contain possible paths around the desk to get to the refrigerator. The possible path to climb *over* the desk for example is not perceived by the agent, perhaps due to the motor repertoire not allowing to climb, preventing the agent to even be aware of that possibility. It will therefore not be contained in the action field. Action abstraction is applied to the perceived paths, summarizing the consecutive motor commands to a representation, a general motor primitive which might simply be going clockwise vs. counterclockwise around the desk. Finally, in order to solve the objective at hand, a decision has to be made between the action possibilities contained in the action field. We assume that when an agent engages in a new self-selected action there is a boost in self-belief regarding the ability to realize it (Kahl et al., 2021). We model this by an increase in HL SoC by a fixed factor. The cognitive function of generating an action field and selecting an action goal is implemented in the ACT-R architecture (Anderson & Lebiere, 1998) in which we base the decision process on the HL SoC.

As soon as an action goal is selected, it is passed on to the next level down in the hierarchy, the sensorimotor control layer (SCL; Kahl et al., 2021). It is tasked to realize the action goal in the environment by interpreting and executing the required individual consecutive motor commands. In our example the action goal is to go clockwise around the desk towards the refrigerator. Now the SCL lays out all the necessary steps to get to the destination. However, these are not the literal steps, but rather the individual muscle movements it takes to walk. These *motor intentions* are at the lowest level of the hierarchy (Pacherie, 2007). They refer to the shortest temporal scale and their implications are specified in great detail. The SCL is also tasked to assess the extent of control over the realization of the action goal. We apply predictive coding implemented in Python (Van Rossum & Drake, 2009) for the evaluation of the LL SoC. For this during each of these motor commands, the SCL generates a sensorimotor prediction (predicted position of the agent in pixel), anticipating the changes in the environment right after the execution. Sensorimotor feedback (also given in pixel position) after the motor command is then compared with the prediction. In the two-layer architecture, the

feedback solely involves visual information of the environment, but it can just as well be haptic or proprioceptive, depending on the modality in which aspects of the intention were formalized. Deviations between predictions and sensory feedback given by the absolute distance in pixel that exceed a certain magnitude are referred to as *prediction errors* and lead to a reduction in LL SoC. Deviations below the magnitude are referred to as matches and lead to an increase in LL SoC. The change in LL SoC as well as the magnitude are again based on individual fixed factors.

Ultimately both layers of our computational model generate a loop in which bottom-up evidence about the execution of motor commands is fed to the CCL which in turn passes top-down intentions, action goals, down the hierarchy to the SCL for it to implement and evaluate action control (Figure 1). We aimed for the model to be capable of performing dynamic decision-making in environments in which each decision (the execution of it) directly influences the situation (Gonzalez et al., 2017). For this we programmed the *Dodge Asteroids* environment. A custom environment that is inspired by the standard domain of Atari games for implementing computational agents (Mnih et al., 2013). It is a video game setting meant to especially challenge action control.

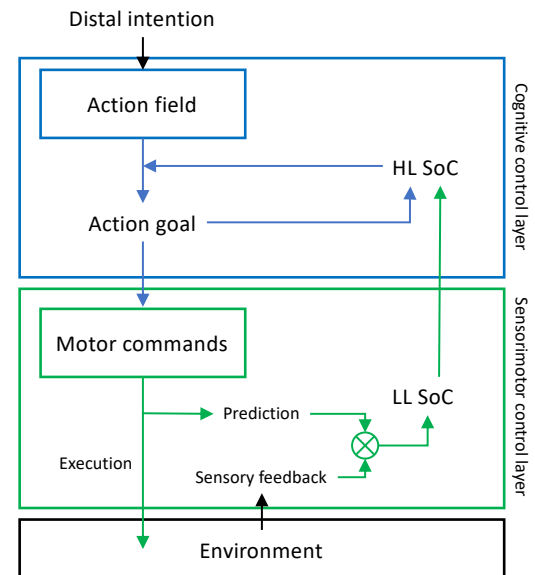


Figure 1. Schematic of the two-layer architecture. The arrows depict the information exchange within the two-layer architecture. In the CCL, an action goal is selected from the action field. The HL SoC is integrated in the decision process. In the SCL the action goal is broken down into individual motor commands. Their execution is accompanied by a prediction. Sensory feedback of the changes in the environment from the motor execution is compared with the prediction. Depending on prediction error or match, the LL SoC is adjusted down or up.

Environment

We have implemented the *Dodge Asteroids* environment in Python (Van Rossum & Drake, 2009). Based on the concept of situated action, the environment must offer the possibility to interact with it. A simple form of interaction is traversing through the environment. Here, each action of the agent has direct influence on its position within the world. Also, for situated action to take place, temporal demand is required (Vera & Simon, 1993). Therefore, we introduce “free fall”, automatic traversal downwards, which if not controlled, will result in a crash or losing the game. The game runs in 60 FPS meaning that every time step is 1/60 of a second. An agent will be situated within the environment having access to the visual information within a restricted area, the observation space, the size of which can be freely varied as well as the height and width of the whole world in pixels. The starting position of the agent will be at the horizontal center of the vertical top of the world. In every time step the agent will automatically traverse 6 pixels vertically down. There are three possible actions the agent can execute at every time step: left vs. stay vs. right. Steering left or right will shift the agent's horizontal position by 6 pixels in the respective direction whereas when executing stay the agent keeps its horizontal position. The objective of the game is to avoid crashing into walls or incoming obstacles and reach the finish line at the bottom of the world. The walls surround the world on both sides. Obstacles are randomly scattered in the world, with the exact position in height and length drawn from a uniform distribution. We control for non-overlapping obstacles. The number of obstacles within the world can be freely varied and reflects the degree of necessary action control, as more obstacles more often demand execution to the left or right. We introduce *drift* to the environment, a manipulation specifically to affect action control. Drift zones cover the entire width of the world and feature variable vertical size (Figure 2). In this example drift is kept constant in its size of 270 pixels in height. Drift can either be visible or invisible, directly influencing whether it is able to be anticipated prior to onset or not. As soon as the agent enters a drift zone, its horizontal position is affected displacing the agent by a specific number of pixels with every time step, whereby drift can be directional (either left or right) or randomized. In randomized drift zones, the drift direction is randomly sampled from left and right in each time step. We kept drift magnitude at 3 pixels, making it able to be countered by steering in the opposite direction of the drift direction. We intended for the agents' actions to still have tangible effects, so that they engage in goal-directed planning.

If the agent is now controlled by our computational model, at the CCL the perceived observation space is searched for collision threats. If one is detected, an action field is generated containing routes that pass the obstacle on the left or right. In the following step, an action goal is selected, with the decision process also considering visible drift. The decision process focuses on the horizontal distance to the

potential action goal, preferring shorter distances. In case of drift, however, action goals are additionally selected on the basis of free space, which offers more possibilities to counteract drift. The action goal is passed to the SCL which tries to reduce horizontal distance to the intended position, thereby steering left or right. At every time step, the SCL generates a prediction of the future position at the next time step. LL SoC is reduced if the prediction is violated due to drift. In case LL SoC falls below the CCL threshold, HL SoC is adjusted downwards. This signals the activation of CCL functions within the model, the fully conscious regulatory control. Now the model tries to infer the directionality of the drift at the CCL, generating a new action field that more accurately incorporates the drift properties. Subsequently, a decision is made again, which is passed on to the SCL.

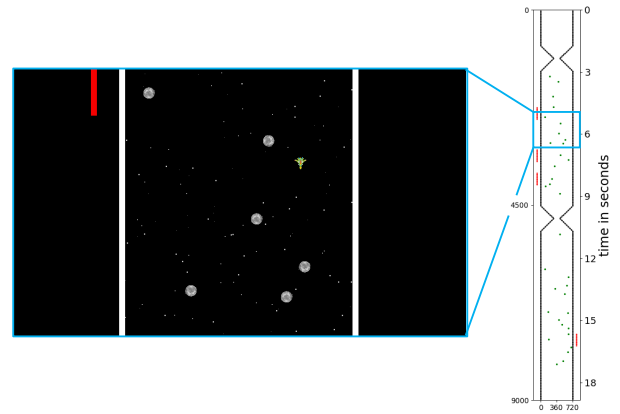


Figure 2. Visualization of an instance within a run in the *Dodge Asteroids* environment. The complete environment for this run is shown on the right. For scale purposes, a world height of 9,000 was chosen. The right axis label shows the time in seconds it takes the agent to get to that location. The enlarged section shows the agent's observation space, the space that can be visually perceived. Obstacles are scattered between the two walls on the left and right. The red bar signals a, in this case, visible drift that is directed to the right.

Methods

To test the dynamic decision-making capability of our model, we will run simulations in which world height will be set to 18,000 and world width to 720. The observation space will be 720x720. These numbers correspond to the visualization of the *Dodge Asteroids* environment on a computer screen for an experiment with humans. The CCL threshold parameter will be varied while the threshold for prediction errors as well as the respective increase and decrease in LL SoC will be kept constant. We will specifically explore the behavior elicited by the computational model focusing on the statistical features of the imminent environment when it crashes. These environmental properties tell us what other capabilities the model needs to solve challenging decision moments. On top of that, we will compare the model behavior with that of human participants. This comparison will include crash situations. However, the focus will be on decision situations based on the paths taken between obstacles. In this way, we

can investigate, for example, the assumption that participants steer towards areas of more free space in case of visible drift. Presumably, however, human behavior in our environment will be based on more than this one policy. Adding more decision criteria to the CCL and simulating behavior again and again could incrementally lead to the accurate modeling of human dynamic decision-making.

Ultimately, we would like to keep a low number of free parameters with unambiguous interpretability in our two-layer architecture to allow for inference of parameter values from real human behavior at a later stage. We therefore differentiate between fixed factors and free parameters, such as the CCL threshold, that we plan to explore.

Conclusion & Prospect

We have explicitly defined the operations within the computational model. At present, we are in the process of implementing our two-layer architecture and simulating behavior within our custom *Dodge Asteroids* environment. This allows us to closely examine the hypotheses behind these operations based on the resulting behavior. Furthermore, we can compare the model behavior with already collected behavior of human participants and, if necessary, adjust the inner operations continuously to better match human dynamic decision-making.

The two-layer architecture described here is intended to be the starting point for a more sophisticated computational model capable of making decisions in more complex environments. For this, we will add a function at the CCL that abstracts and thus simplifies the visual environment of the agent. We will introduce a convolution of the visual environment, with the pooling factor being a free parameter that reflects the granularity of the mental representation of what is visually perceived. The current state abstraction of the two-layer architecture, which is an abstraction of the control state, is thus extended by a state abstraction of the direct physical environment of the agent. This could lead to generating action goals that lie behind an obstacle. Therefore, we need to adjust the SCL as well because simply reducing the horizontal distance will not do the trick anymore.

We will employ model-based deep reinforcement learning (Sutton & Barto, 1998) for the SCL. A forward model will generate sensorimotor predictions. Furthermore, during the training, the SCL will elaborate policies that will make it possible to realize even complex paths around obstacles. While this will remove explicitness from the 2-layer architecture, it also allows us to focus on the functions in the CCL and examine them in more detail. Overall, the model thus becomes more potent in making and executing dynamic decisions. These increased capabilities make it possible for us to introduce even more dynamics to the *Dodge Asteroids* environment and add more uncertainties, giving us a more variable test bed to explore the limits of human dynamic decision-making.

References

- Anderson, J. R., & Lebiere, C. J. (1998). *Hybrid modeling of cognition: Review of the atomic components of thought*. Erlbaum.[AGB].
- Eppe, M., Gumbsch, C., Kerzel, M., Nguyen, P. D., Butz, M. V., & Wermter, S. (2022). Intelligent problem-solving as integrated hierarchical reinforcement learning. *Nature Machine Intelligence*, 4(1), 11–20.
- Flash, T., & Hochner, B. (2005). Motor primitives in vertebrates and invertebrates. *Current opinion in neurobiology*, 15(6), 660–666.
- Gibson, J. J. (1966). *The senses considered as perceptual systems*.
- Gonzalez, C., Fakhari, P., & Busemeyer, J. (2017). Dynamic decision making: Learning processes and new research directions. *Human factors*, 59(5), 713–721.
- Kahl, S., Wiese, S., Russwinkel, N., & Kopp, S. (2022). Towards autonomous artificial agents with an active self: modeling sense of control in situated action. *Cognitive Systems Research*, 72, 50–62.
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- Norman, D. A. (1988). *The psychology of everyday things*. Basic books.
- Pacherie, E. (2007). The Sense of Control and the Sense of Agency. *Psyche*, 13(1), 1.
- Pezzulo, G., Barsalou, L. W., Cangelosi, A., Fischer, M. H., McRae, K., & Spivey, M. J. (2011). The mechanics of embodiment: A dialog on embodiment and computational modeling. *Frontiers in psychology*, 2, 5.
- Schaal, S. (2006). Dynamic movement primitives—a framework for motor control in humans and humanoid robotics. *Adaptive motion of animals and machines*, 261–280.
- Shapiro, L. (2019). *Embodied cognition*. Routledge.
- Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Bradford. The MIT Press, London, England.
- Turner, M. H., Sanchez Giraldo, L. G., Schwartz, O., & Rieke, F. (2019). Stimulus-and goal-oriented frameworks for understanding natural vision. *Nature neuroscience*, 22(1), 15–24.
- Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace.
- Vera, A. H., & Simon, H. A. (1993). Situated action: A symbolic interpretation. *Cognitive science*, 17(1), 7–48.
- Wang, J. (2017). Symbolism vs. Connectionism: A Closing Gap in Artificial Intelligence. *Jieshu's Blog*. Dec.

Using Cognitive Models to Test Interventions Against Mind-Wandering During Driving

Moritz Held^{1,2} (moritz.held@uol.de), Andreea Minculescu² (a.minculescu@student.rug.nl), Jochem Rieger¹ (jochem.rieger@uol.de), Jelmer Borst² (j.p.borst@rug.nl)

¹Dept. of Applied Psychology, University of Oldenburg, Germany

²Bernoulli Institute, Dept. of Artificial Intelligence, University of Groningen, the Netherlands

Keywords: ACT-R; driving; mind-wandering

Introduction

Workload while driving has been investigated in numerous forms. While most researchers focus on cognitive *overload*, increasing automation draws attention to the effects of low cognitive workload, which has shown to be harmful as well (e.g., Nijboer et al., 2016). The authors suggested that low driving performance in low workload scenarios may be due to drivers starting to mind-wander, which has been found to have a harmful effect on driving performance (Yanko & Spalek, 2014).

An optimal level of workload – neither under- nor overload – can be reached by using an adaptive system that increases or decreases workload. However, artificially increasing workload during driving to prevent cognitive underload may be dangerous as it can quickly have negative effects if the system is miscalibrated.

In this work, we combined three different models in the cognitive architecture ACT-R (Anderson, 2007) to test different interventions intended to prevent harmful mind-wandering during driving.

Methods

We combined the seminal driving model by Salvucci (2006) with a mind-wandering model by Van Vugt et al. (2015) to simulate the dangerous effects of mind-wandering while driving.

The driving model used in this work is identical to the Salvucci driving model (2006) and consists of a driving loop

that utilizes a two-point steering model to maintain a steady lane position.

When the model starts mind-wandering, it starts retrieving random chunks from the declarative memory of ACT-R. According to the model by Van Vugt et al. (2015), we defined 31 chunks in declarative memory. The chunks are divided in 30 memories and a single chunk that encodes the spontaneous refocusing on the driving task (“remembering to drive”). Upon a successful retrieval of a memory, the model issues another retrieval and continues until it hits one of two criteria: a) the model retrieves the single chunk that encodes the refocusing on the driving task or b) the model refocuses to the driving task upon exceeding a lateral threshold as this was assumed to be so disruptive that it would stop mind-wandering.

During mind-wandering, the model does not initiate any driving productions.

Next, we simulated varying levels of effort needed to process an auditory stimulus based on a listening model by Borst et al. (2010, exp. 3). We defined a “mild load” stimulus that needed only superficial processing and an “intermediate load” stimulus that recruited the full listening stream of the model by Borst et al. (2010) to process. When the mild-load stimulus appears in the environment (step 2 in Figure 1), it is stuffed into the aural-location buffer of ACT-R (step 3) and after attending the stimulus (step 4), the model refocuses to driving. However, the “intermediate load” stimulus needs additional processing as the audio signal needs to be decoded (step 5) and the meaning of the words needs to be accessed (step 6-7) before driving can be resumed.

Using this procedure, we essentially simulated adaptive systems that may display auditory stimuli that require

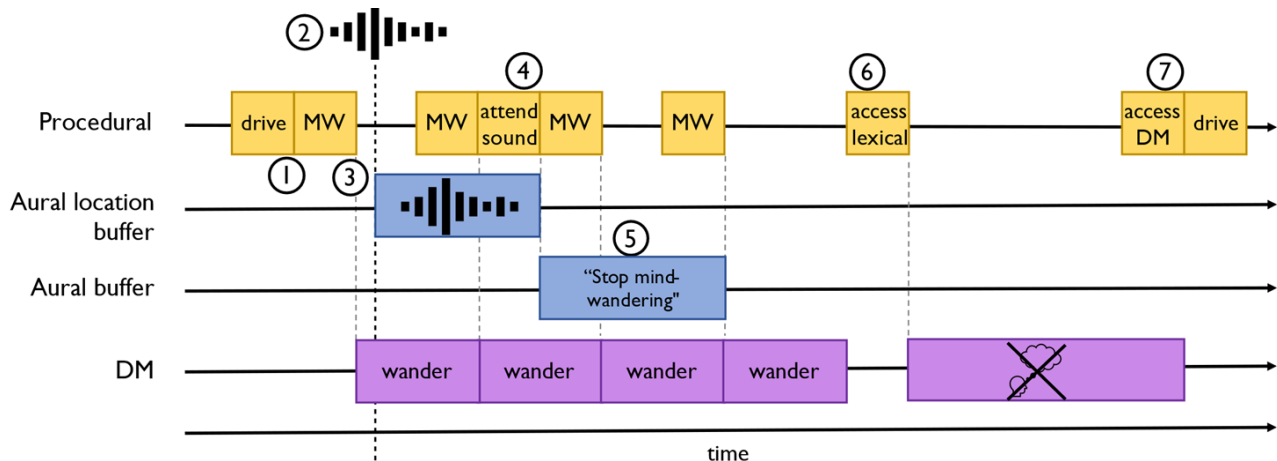


Figure 1: Sketch of the full model.

different amounts of cognitive workload to process. We devised four interventions intended to reduce mind-wandering in four different models.

We tested these interventions in a simple driving environment, where the model had to follow a lead car that was driving at a consistent speed with no brakes. The driving environment consisted of a highway with no turns or bends in the road. Each model run lasted for five minutes.

Intervention models

1. The *mild load model* simulates a system that injects minor load continuously.
2. The *intermediate load model* simulates a system that injects intermediate load to ensure the human is not underloaded.
3. The *warning model* simulates the effects of a single disruptive signal that is activated when the human is underloaded and starts mind-wandering. The warning is processed using the full listening stream.
4. The *mild load + warning model* simulates a system that continuously injects minor cognitive load and, in addition, plays a warning signal when the human starts mind-wandering.

Results

To evaluate the models, we calculated the number of mind-wandering productions in each model run, which indicates the effectiveness of the respective intervention.

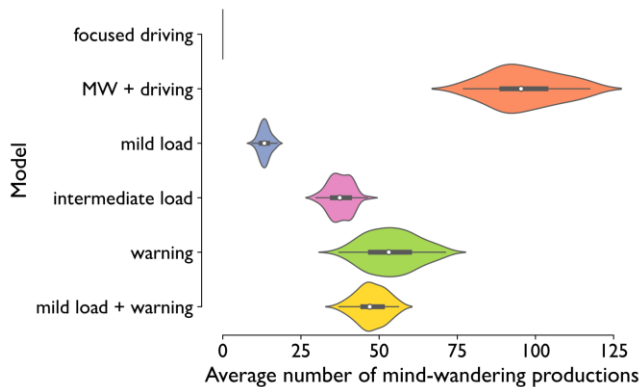


Figure 2: Average number of mind-wandering productions per model run.

The results show that the amount of mind-wandering can be reduced by all interventions (Figure 2). However, the mild-load model showed the lowest amount of mind-wandering. In addition, the results indicate that inducing intermediate workload does not interrupt mind-wandering as quickly and therefore more productions may fire before an episode gets interrupted.

Interestingly, adapting the intervention to the current mental state of the model (mind-wandering vs. driving) does not reduce mind-wandering more effectively than continuously inducing cognitive load while driving. Both the warning and the mild load + warning model show an

increased amount of mind-wandering compared to the models that continuously induce a fixed load on the driver.

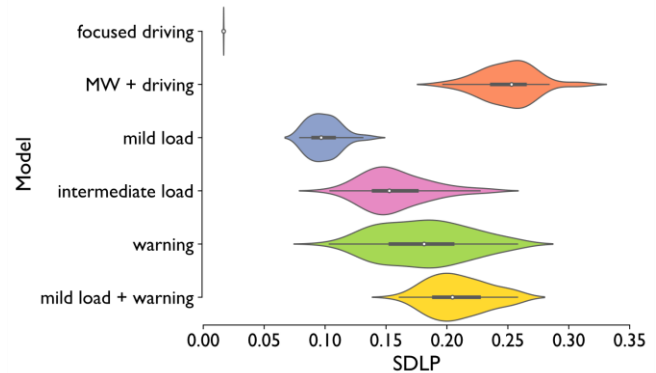


Figure 3: Standard deviation of lateral position in meters shown for all models.

Similarly, all interventions lead to a lower standard deviation of lateral position (SDLP) compared to the MW + driving model (Figure 3). However, the additional processing cost in the intermediate load model lead to a higher SDLP compared to the mild load model. In addition, even though the mild load + warning model shows fewer mind-wandering productions than the warning model, it displays the highest SDLP, indicating the worst driving performance.

Discussion

In this work, we compared different interventions intended to prevent mind-wandering during driving. We found that interventions imposing little workload showed the most effect. Furthermore, we found that interventions that adapt to the cognitive state of the model perform worse than interventions that continuously impose workload.

In the continuous models the auditory processing is constantly ongoing, which may lead to a very fast interruption of the mind-wandering process if the auditory processing of a stimulus is almost complete. However, in the adaptive models, the auditory processing always starts after mind-wandering has begun as the warning signal is only displayed when the model starts mind-wandering. This acts as a switch-cost and results in a longer time until the mind-wandering process is interrupted.

In conclusion, our models indicate that with increases in cognitive load, there comes an additional processing cost that affects driving performance negatively. In addition, interventions designed to prevent mind-wandering in driving, need to account for the cost of switching to a new stimulus.

References

- Anderson, J. R. (2007). How Can the Human Mind Occur in the Physical Universe? In *How Can the Human Mind Occur in the Physical Universe?* (Vol. 148).
- Borst, J. P., Taatgen, N. A., & van Rijn, H. (2010). The Problem State: A Cognitive Bottleneck in Multitasking. *Journal of Experimental Psychology: Learning Memory and Cognition*, 36(2), 363–382.

- Salvucci, D. (2006). Modeling driver behavior in a cognitive architecture. *Human Factors*, 48(2), 362–380
- van Vugt, M., Taatgen, N., Sackur, J., & Bastian, M. (2015). Modeling mind-wandering: A tool to better understand distraction. *Proceedings of the 13th International Conference on Cognitive Modeling*, 252.
- Yanko, M. R., & Spalek, T. M. (2014). Driving With the Wandering Mind: The Effect That Mind-Wandering Has on Driving Performance. *Human Factors*, 56(2), 260–269. <https://doi.org/10.1177/0018720813495280>

An Initial Cognitive Model of a Radar Detection Task

Alexander R. Hough (alexander.hough.1@us.af.mil)
 Christopher Stevens (christopher.stevens.28@us.af.mil)
 Elizabeth Fox (elizabeth.fox.9@us.af.mil)
 Christopher Myers (christopher.myers.29@us.af.mil)

Air Force Research Laboratory
 Wright-Patterson AFB

Abstract

In adversarial operational environments like radar monitoring, humans have to monitor large amounts of information, multi-task, and manage threats. They may also face electronic disruption or attacks aimed at degrading radar monitor effectiveness (a.k.a electronic warfare or EW). In these settings, it is unclear how frequent changes in personnel, training, and updates to visual displays affect an operator's readiness. A recent experiment used an analogous radar monitoring task to investigate effects of display density and electronic warfare on an operator's threat detection performance. Here, we present a cognitive model capable of completing a scaled-down version of that task to better understand the experimental results and underlying cognitive processes. Similar to the human experiment, our cognitive model completed conditions comprised of changes to the nature of the task(s), the number of targets to track, and the presence or absence of distractors, deemed 'friendlies'. Although this initial cognitive model uses primarily default ACT-R parameters, it was able to capture patterns in human performance across conditions. We present the results and discuss limitations to address in future work.

Keywords: Cognitive model; ACT-R; Multiple object tracking; Multitasking

Introduction

Many tasks in modern environments require individuals to maintain awareness of complex, evolving situations. For example, air-traffic-control, lifeguarding, and childcare are all situations in which an overlooked detail or event can lead to serious consequences. It is important to understand the way in which people maintain awareness in these complex situations, and the circumstances under which that awareness will be impaired. Here we consider a particular variety of radar monitoring that occurs in naval operational environments. In these settings, operators must monitor visual displays with many entities (tracks) and identify certain tracks for follow on action. These situations often involve large amounts information, multitasking, and continuous decision making. Operators rely on systems like the Aegis Combat System (i.e., ACS) that continuously update visual displays with information from multiple sources (Bath, 2020). Appropriate representation of entity types (e.g., hostiles and friendlies) in such systems is important for correctly identifying threats to avoid accidents (Pogue, 2016) and it is not clear how frequent changes to the system and training affects monitor's readiness (Fisher & Kingma, 2001).

Visual search literature provides constraints (Treisman & Gelade, 1980; Glavan, Haggitt, & Houpt, 2020) and models (Wolfe, 2021; Nyamsuren & Taatgen, 2013; Fleetwood

& Byrne, 2006) for representing human-like visual search. However, dynamic stimuli are out of scope for most models. Here, we present a cognitive model capable of completing a laboratory radar monitoring task with dynamic stimuli. We show how well it captures human performance and discuss its limitations to address in future work.

MOT-EW Task

The MOT-EW task (Fox et al., 2023) served as an AEGIS analog to investigate human performance in a laboratory setting. The multiple object tracking (MOT) task (Figure 1) involved four quadrants with moving hostiles and friendlies. Hostiles (i.e., targets) were presented as red circles and friendlies (i.e., distractors) were comprised of octagons and diamonds that were pink or magenta. Objects had protruding black track lines indicating the direction they were moving. Hostiles were slightly larger and had longer track lines.

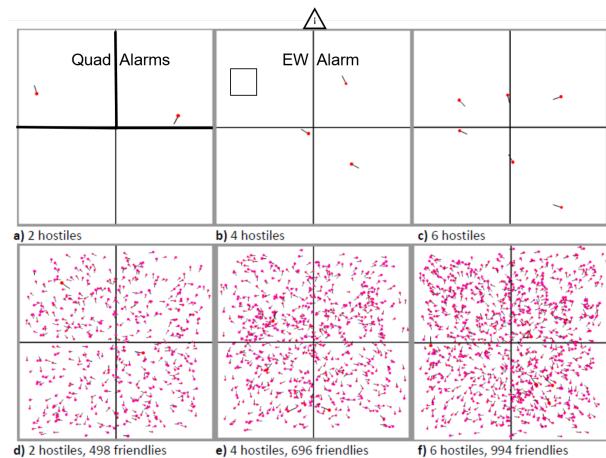


Figure 1: Depiction of the MOT-EW task, alarm states for quadrants, and EW attacks from Fox et al. (2023).

The task involved turning on quadrant alarms when at least one hostile was present and turning off alarms when hostiles were no longer present in that quadrant (See Figure 1a). The electronic warfare (EW) task used the same stimuli, but involved a hostile disappearing for 1-4 seconds once during 30 second windows. An EW alarm was turned on after a hostile had disappeared and off when it returned (see Figure 1b). In Fox et al. (2023), participants completed 18 conditions with

three independent variables: number of hostiles (i.e., 2, 4, and 6), presence of friendlies (i.e., yes or no), and task (MOT, EW, or MOT-EW). Each condition was completed in a separate session that lasted approximately 12 minutes. Our goal was to develop a cognitive model capable of capturing human behavior in all of these 18 conditions.

MOT-EW Model

We implemented a cognitive model, we simply call the MOT-EW model, in the ACT-R cognitive architecture (Anderson, 2007). ACT-R includes both symbolic and sub-symbolic structures, and modules that represent systems of the mind. The MOT-EW model uses the goal, vision, motor, and procedural modules. The goal module serves as the models focus and stores goal relevant information. The vision module allows the model to perceive visual stimuli and direct attention. The motor module allows the model to turn on/off alarms using a keyboard. The procedural module uses condition-action rules (i.e., productions) to represent knowledge about how to do things and to drive the behavior of the model.

Our approach was to construct the simplest model without modifying parameters to test the "out-of-the-box" capability of ACT-R to complete the MOT-EW task. However, we had to make three modifications so the model could reasonably perform the task: 1) We modified the experimental task, 2) changed one parameter, and 3) deviated from typical visual perception methods allowing the model to reach and maintain human-like performance. We start by explaining the MOT-EW model task and then describe the model.

Model Task

Our overall goal for designing the model was to remain as faithful as possible to the experimental task, and to default ACT-R assumptions and design patterns. However, due to several constraints we had to make changes to both the virtual version of the task and the model. The first change we made to the task was reducing the number of friendlies on the screen. We reduced the number of friendlies for two reasons (Figure 2). First, if we included several hundred objects, the ACT-R visicon (i.e., collection of information available to the visual module "what" system) would get bogged down updating positions of moving objects. Interestingly, we did learn that the model is capable of handling at least 100 objects, but time and computation costs become unacceptable. Second, ACT-R has limited visual search capabilities. By default, it is possible to build a model that searches for more than one feature simultaneously (e.g. find a red circle). However, this will result in flat response time curves with respect to set size, contrary to what has been well established in the visual search literature for decades (Treisman & Gelade, 1980).

It would also result in the inability to account for effects of the presence of friendlies, which were observed in the original experiment (Fox et al., 2023). Alternatively, one could search for one feature at a time (e.g. find a red object) and search through the available objects linearly until an object is found that matches both desired criteria (Fleetwood & Byrne,

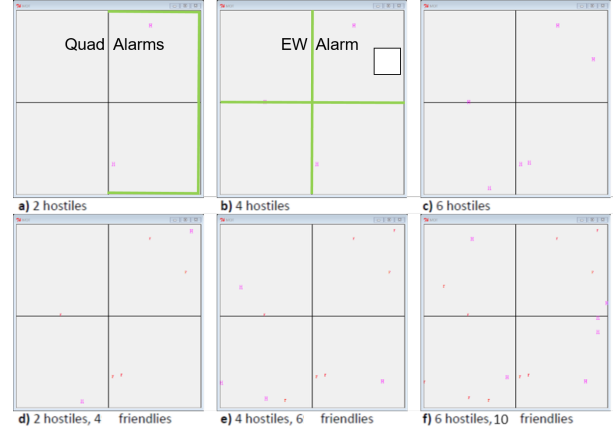


Figure 2: Depiction of model MOT-EW task and alarm states for quadrants and EW attacks.

2006). This produces set size effects, but still becomes prohibitively slow for a display with hundreds of objects. These limitations have motivated previous extensions to the vision module, such as PAAV (Nyamsuren & Taatgen, 2013) and JSegMan (Tehranchi & Ritter, 2018). However, these capabilities do not currently exist in a form that is compatible with the version of ACT-R we used for the model (7.14). We also made two minor changes to the task representation. The model task uses colored letters instead of colored shapes and the location of alarms were modified. Letter locations can be modified, which ensured moving letters were still considered the same object. The model task is therefore, a scaled-down version of the experimental task.

Model Description

The model uses only four modules (i.e., goal, vision, motor, and procedural memory) and there is no semantic or procedural learning. There are several important parameters related to the task: visual attention latency, the speed of productions (i.e., processes), and visual finsts (i.e., number and span). All but one parameter, number of visual finsts, are left at default values. The number of visual finsts controls how many visual objects can be marked as attended and the span controls how long they remain marked. We have changed the number of visual finsts to 16 (default is 4). In addition, we deviated from the standard visual find-attend-encode loop. Typically, an object is found, the model shifts its attention to that object, and the object is then encoded (i.e., identified). In two instances, we allow the model to skip the attend step to simulate human's ability to extract information peripherally without directly fixating on it. We further explain why we adjusted the number of visual finsts and deviated from the standard visual processing loop in the following sections to provide context.

The MOT and EW tasks have their own set of processes, but share several general productions. They are completed separately in single task conditions or serially interleaved during dual task conditions. The model was provided informa-

tion about the task (i.e., MOT, EW, or DUAL) at the start of each session (i.e., condition), but was not provided with information about the amount of hostiles or friendlies. We start by describing the MOT processes, followed by EW, and then the full model that interleaves both.

MOT Processes. For the MOT task, the model has to differentiate between quadrants, determine their unique alarm state, and identify which do and do not have a hostile. To accomplish this, the model searches through quadrants one at a time in a clockwise direction and makes alarm decisions (Figure 3). Once a quadrant is selected (e.g., search NW), the model orients to that quadrant’s coordinates and checks the alarm state (i.e., check-alarm). Rather than attending and encoding the alarm state, the model finds the quadrant and encodes alarm state information or color peripherally without shifting attention directly. Alarm state information for the current quadrant is held in the goal buffer. After checking the alarm state, the model finds objects in the quadrant (i.e., find-object). If a friendly is found, it is marked as attended without shifting attention (i.e., friendly-no-attend), like the quadrant alarm state. The model continues to find objects until there are no new objects to find (i.e., all objects marked) or a hostile is found. In rare cases, the model shifts attention to a hostile, but there is no object at that location. This occurs because the object can potentially move beyond the focal area of the model. A reorient production (i.e., reorient) handles these rare cases and reorients the model to find the nearest object to the attended location, which should be the intended target. This represents a fixation that was slightly off and corrected.

Once all objects are searched or a hostile is found, the model makes an alarm decision. If the quadrant was completely searched and no hostile was found, the model turns off the alarm if it is currently on (i.e., turn-off-alarm), or moves to the next quadrant if it is already off (i.e., alarm-off-ok). If a hostile is found, the model shifts attention to the location of the hostile (i.e., hostile-attend). The subsequent production encodes the hostile and either turns on the alarm if currently off (i.e., turn-on-alarm), or moves to the next quadrant if already on (i.e., alarm-on-ok). The model turns on and off alarms by pressing a keyboard key corresponding to the quadrant using the punch command that assumes fingers are resting on the home keys. After making a decision, the quadrant is marked as searched in a goal buffer slot and visual finsts are cleared. The model leverages visual finsts to ensure that quadrants are searched and setting the number of finsts to 16 ensures that the model does not get stuck in an endless loop within a quadrant (e.g., number of objects exceeds the parameter). Furthermore, clearing visual finsts ensures that the model does not skip over a marked object that moves into another quadrant within the default finst span time of 3 seconds. This could result in missing a hostile and either not turning on a quadrant alarm or incorrectly turning one off. The model completes the same process for each quadrant until all quadrants are searched and then model starts another quadrant search cycle (i.e., all-quads-searched).

Now that the MOT processes have been described, we provide an explanation for why we chose to skip the attend step for quadrant alarms and friendlies. The standard time for each production is 50ms, shifting attention takes 85ms, and punching (i.e. pressing a key) takes 210ms. The standard find-attend-encode loop would take 235ms to encode a single object or alarm state and changing an alarm state takes 260ms. Therefore, it would take 730ms just to encode a quadrant alarm state, encode a single object, and change an alarm state. Additionally, it would take 965ms to encode two objects, 1200ms for three, and 1435 for four. For comparison, the average human response time for changing quadrant alarm states is 900ms for the single MOT task and that includes conditions ranging from 2-6 hostiles and 498-994 friendlies. Reasonable response times are important, because they are related to alarm state change accuracy. By eliminating the need for the model to attend to alarm states and friendlies, we were able to achieve an average 1020ms response time across varied hostiles and friendlies in the scaled-down MOT model task. This decision also aligns with the human ability to extract information without direct fixation (Wolfe, 2021) and change alarm states quickly. These decisions are more important for EW processes, where the standard find-attend-encode loop would result in unreasonable response times, even in this scaled-down version of the experiment.

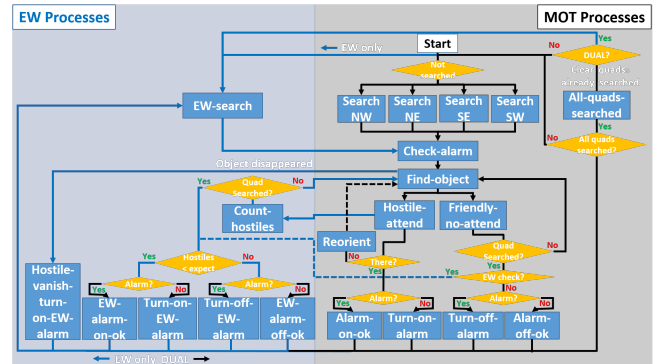


Figure 3: Diagram of processes to complete both MOT and EW tasks.

EW Processes. To complete the EW task, the model leverages MOT productions, which are considered general visual search processes (i.e., check-alarm and find-object). The model continuously checks whether an EW attack is occurring (i.e., a hostile has disappeared), and the EW alarm is turned on when an EW attack is ongoing and off when it ends (i.e., hostile reappears). The model starts the EW task by orienting to the entire screen (i.e., ew-search) and treating all four quadrants as the search area. Next, it checks the alarm state the same way as it does for MOT (i.e., check-alarm) and stores the alarm state in a slot in the goal buffer. The model then searches for objects (i.e. find-object), but contrary to the MOT, it is an exhaustive search. There are a maximum of 16 objects in one of the conditions (i.e., 6 hostiles and 10

friendlies) and this is why we set the number of visual finsts to 16. Although exhaustive, the model completes EW search similar to MOT. It perceives friendlies without attending and marks them as attended (i.e., friendly-no-attend). Using a find-attend-encode loop for all objects would be more of an issue for EW as there are more objects to search. It would take between 1200ms (i.e., 2 hostiles without friendlies) and 4255ms (i.e., 6 hostiles and 10 friendlies) to do a complete search and change the alarm state. For reference, the average human reaction time across number of objects was 734ms (1006ms for our model). Just like during MOT, if the model finds a hostile it shifts attention to its location (i.e., hostile-attend). However, rather than moving right to an alarm decision as in the MOT, the model encodes the hostile and keeps a count of how many hostiles have been attended (i.e., count-hostiles). Similar to participants, the model does not know how many hostiles to expect at the start of the session. For generality across conditions, the model updates a slot in the goal buffer that stores the amount of hostiles to expect. This slot is set to the highest number of hostiles counted during a session. For instance, if there are four hostiles and no EW attacks occur during the first five seconds, the model will count and set expected hostiles to four. If there was an EW attack at the start of the session, the model would count three and miss the EW attack. Once the model has performed its exhaustive search, it compares how many hostiles were counted with the amount expected (i.e., conditions for alarm decisions). If it found the amount expected it turns off the EW alarm (i.e., turn-off-EW-alarm) if already on or it does nothing if the alarm is already off (i.e., EW-alarm-off-ok). If it found less hostiles than expected, it turns on the EW alarm if currently off (i.e., turn-on-EW-alarm) or does nothing if the alarm is already on (i.e., EW-alarm-on-ok). There is one additional alarm decision production (i.e., hostile-vanish-turn-on-EW-alarm) that is analogous to a person seeing a hostile disappear. A hostile could disappear after the model finds a hostile and starts shifting attention to that hostile. This hostile-vanish production handles this by turning on the EW alarm. Once a decision is made, the finsts are cleared and the model starts another EW check.

MOT-EW Processes. The complete model (Figure 3) is used for single and dual task conditions. During the dual task conditions, the model has to change quadrant alarm states based on hostile presence and change the EW alarm depending on whether an EW attack is occurring. The current model treats these as separate tasks and interleaves them. MOT processes are given priority and EW checks are initiated after all four quadrants have been searched. Therefore, a full quadrant search and EW check can be considered a complete cycle in the dual task conditions. After all quadrants are searched, the all-quads-search production fires, followed by the EW-search production that begins the EW check. One exception to this cycle is the production that turns on the EW alarm if a hostile disappears while shifting attention to a hostile (i.e., hostile-vanish-turn-on-alarm), which can supersede the cycle.

Results

We assess model performance and show how well it fits collected human data from the MOT-EW experiment (Fox et al., 2023). The experiment included 28 participants and a within-subjects 3 (amount of hostiles: 2, 4, and 6) x 3 (task: MOT, EW, and MOT-EW) x 2 (friendlies: present and not present) design. We simulated 25 participants for all 18 conditions. As the model does not possess individual differences, the model essentially simulates one participant completing the experiment 25 times. To assess how well the model captured the human data, we compared single and dual task performance for MOT and EW separately. We included both accuracy and response time for correct responses (i.e., time for correct alarm changes) as the dependent measures. For each comparison, we assess behavior patterns in dependent measures across variations in the number of objects (i.e., amount of hostiles and presence of friendlies). We used correlations to assess the ability for the model to capture patterns across conditions and use root mean squared error (RMSE) to assess the average difference between the model and human data.

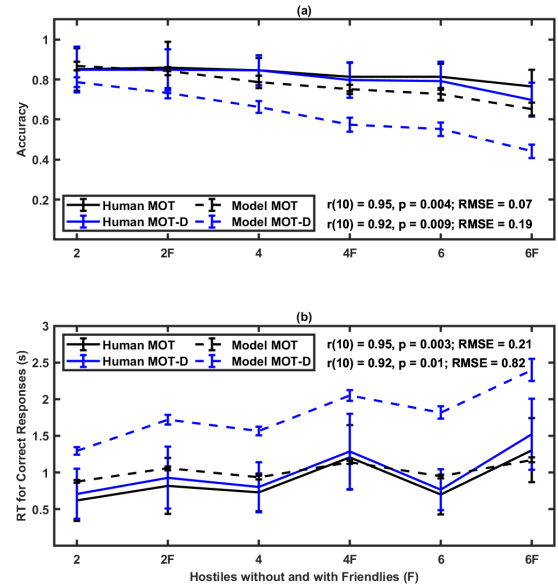


Figure 4: Model fit to human accuracy (a) and RT for correct responses (b) across all MOT conditions.

MOT Task

For the MOT (Figure 4), the model captured accuracy and response time patterns for single and dual task conditions ($p < .05$). However, the average difference for dual task accuracy ($RMSE = .19$) and response time ($RMSE = 820ms$) was higher than single task accuracy ($RMSE = .07$) and response time ($RMSE = 210ms$). The model had a larger difference between single and dual task accuracy (.15) and response time (787ms) than human data (.02 and 106ms, respectively). Therefore,

the model was better able to capture human performance in the single task MOT, then dual task. We also assessed the overall relationship between accuracy and response time in single and dual conditions for the human data and model. For the single task MOT, there were non significant negative relationships between accuracy and reaction time for both human ($r(10) = -.78, p = .07$) and model data ($r(10) = -.63, p = .177$). However, there were significant negative relationships in the dual task MOT for both human ($r(10) = -.82, p = .044$) and model ($r(10) = -.91, p = .010$).

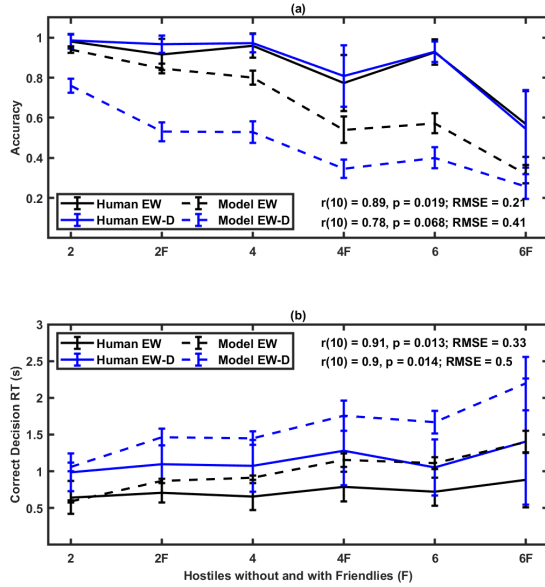


Figure 5: Model fit to human accuracy (a) and RT for correct responses (b) across all EW conditions.

EW Task

For the EW task, there were clearer average differences between human and model performance. The model captured the pattern for the EW single task accuracy, but not dual task accuracy ($p > .05$). The model had a higher average difference for EW accuracy in single ($RMSE = .21$) and dual task ($RMSE = .41$) compared to the MOT ($RMSE = .07$ and $RMSE = .19$, respectively). Similar to the MOT, the model had a greater difference in EW accuracy between single and dual task (.2) compared to humans (.01). The model was able to capture EW response time patterns for both single and dual task ($p < .05$) and similar to the MOT, the average difference was higher for the dual task ($RMSE = .5$) compared to the single ($RMSE = .33$). However, the average difference for EW dual task was lower than the MOT dual task (.5 compared to .82), suggesting EW processes had a stronger negative effect on MOT in the dual task conditions than vice versa. Again, the human data demonstrated a lesser difference between single (.01) and dual task (415ms) compared to the model (.2 and 594ms, respectively). We also assessed the relationship

between accuracy and response time for the human data and model. In the single task EW, there were significant negative relationships for both human ($r(10) = -.98, p = .001$) and model ($r(10) = -.98, p = .001$). There were also significant negative relationships in the dual task EW for human ($r(10) = -.94, p = .004$) and model ($r(10) = -.97, p = .001$).

Table 1: Relationships between accuracy and RT.

	Condition	df	r	p
Human	Single MOT	10	-.78	.066
	Dual MOT	10	-.82	.044*
	Single EW	10	-.98	.001*
	Dual EW	10	-.94	.005*
Model	Single MOT	10	-.63	.177
	Dual MOT	10	-.91	.010*
	Single EW	10	-.98	.001*
	Dual EW	10	-.97	.001*

Discussion

The model was able to capture behavior patterns for both accuracy and response time, via significant correlations, for all but one task and condition (i.e., EW dual task accuracy). Although, the average difference in performance as measured by RMSE varied and in some cases, was rather high. MOT performance was captured better than EW, particularly with the single task. There was more deviation in MOT accuracy and response time for the dual task compared to the single task, which was not found in the human data. Interestingly, the model dual task average difference for MOT response time was greater than EW. This suggested that completing EW had a stronger effect on MOT performance in the dual task conditions than number of hostiles and presence of friendlies.

The model was not able to capture EW behavior as well. In contrast to MOT performance, model performance for EW is consistently lower for both single and dual. This suggests the EW model processes are not as well aligned with humans and is likely contributing to the performance decrement seen in the MOT during dual task. The model takes longer than humans to change EW alarm states, suggesting humans are doing EW checks differently or potentially adopting different EW strategies across conditions. For example, the model EW checks involve exhaustive search regardless of condition, and tasks are treated as separate and are interleaved serially instead of processed in parallel (e.g., multitasking). Given the nature of the model EW checks, there should be a rather linear decrease in EW performance as the number objects to search increases and this is evident in the EW figures (Figure 5). We see a similar pattern with the stronger negative trend in the MOT dual task accuracy. The trend for the MOT dual task reaction time is consistent with that of single task, but responses took an average of 787ms longer across conditions.

As stated, this model was intended to test out of the box capabilities of ACT-R and serve as a baseline. We believe we have achieved close to the best performance possible using canonical ACT-R without adjusting more than one parameter. There are of course, limitations with the existing model. Next we discuss these limitations and ideas to improve the model.

Limitations and Future Work

The model has several limitations: 1) There is no variation in task execution for single and dual task conditions (i.e., no strategies or individual differences), 2) the speed of processing and visual attention is notably slower than humans, and 3) EW performance is notably worse than MOT and is the likely cause for the decrease in dual task MOT performance.

The model has a rigid approach to completing the MOT, EW, and interleaving them in the dual task condition. There are no strategies and the model does not learn. In the human data, there is more variation within and between participants compared to the model. However, this is not surprising given the model could be considered one individual completing the experiment repeatedly. Using current and future human data could identify strategies, condition specific strategies, and perhaps clusters of individuals or types that have similar patterns of behavior. This would inform the ability of the model to capture individual differences and adding learning mechanisms enables strategy shifts across conditions. In addition, we could consider threaded cognition (Salvucci & Taatgen, 2008) as a method to enable multitasking rather than treating tasks as separately and interleaving them.

As mentioned, ACT-R does not currently have visual search capabilities beyond deterministic or featureless strategy-based search. To facilitate visual search, we deviated from the typical find-attend-encode loop and allowed the model to attend some stimuli peripherally. This decision was guided by the visual search literature (Wolfe, 2021) and was a plausible way to speed up visual search processes. Despite our efforts, it was clear that the model took longer to change alarm states than humans, which also relates to accuracy. Furthermore, the model interacted with a scaled-down version of the task presented to human participants. To address visual search capabilities and improve model performance, we plan to revive or implement features from the PAAV module (Nyamsuren & Taatgen, 2013). PAAV has both bottom up (e.g., color and shape salience) and top-down (e.g., strategy based) features that enable more directed visual search observed in humans. Furthermore, this should eliminate the need for exhaustive serial search. After making progress with visual search capabilities, we plan to scale up the model task to better align with the experimental task.

The model performed worse for the EW task, which appeared to reduce MOT performance in the dual task conditions. We believe this resulted from: 1) flawed EW processes, not as well aligned with human behavior and 2) limited visual search capabilities that encouraged exhaustive serial search. We plan to address these points with the future work outlined above: 1) better understand and implement strategies and 2)

make some improvements to visual search capabilities.

Conclusions

We successfully implemented a simplified model in ACT-R capable of performing a complex radar detection task and testing hypotheses about the underlying cognitive processes. The model provided a reasonable fit to human data across 18 conditions and serves as a solid baseline by demonstrating the out of the box capabilities of ACT-R. Future work will extend this model to address identified limitations of the architecture, model performance, and the scaled-down model task.

Acknowledgments

This research was supported by the U. S. Air Force Research Laboratory's 711th Human Performance Wing, Cognitive Models and Sensory Systems Branches. The contents have been reviewed and deemed Distribution A. Approved for public release. Case number: AFRL-2023-1409. The views expressed are those of the authors and do not reflect the official guidance or position of the United States Government, the Department of Defense, or the United States Air Force.

References

- Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press.
- Bath, W. G. (2020). Overview of platforms and combat systems. *Johns Hopkins APL Technical Digest*, 35(2), 90–98.
- Fisher, C. W., & Kingma, B. R. (2001). Criticality of data quality as exemplified in two disasters. *Information & Management*, 39(2), 109–116.
- Fleetwood, M. D., & Byrne, M. D. (2006). Modeling the visual search of displays: a revised act-r model of icon search based on eye-tracking data. *Human-Computer Interaction*, 21(2), 153–197.
- Fox, E. L., Stephenson, A., Stevens, C. A., & Bowers, G. (2023). Predictors of human efficiency in radar detection tasks. In *Proceedings of the 18th international conference on cyber warfare and security*. Towson, Maryland, USA.
- Glavan, J. J., Haggit, J. M., & Houpt, J. W. (2020). Temporal organization of color and shape processing during visual search. *Attention, Perception, & Psychophysics*, 82, 426–456.
- Nyamsuren, E., & Taatgen, N. A. (2013). Pre-attentive and attentive vision module. *Cognitive systems research*, 24, 62–71.
- Pogue, D. (2016). 5 of the worst user-interface disasters. *Scientific American*.
- Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: an integrated theory of concurrent multitasking. *Psychological review*, 115(1), 101.
- Tehranchi, F., & Ritter, F. E. (2018). Modeling visual search in interactive graphic interfaces: Adding visual pattern matching algorithms to ACT-R. In *Proceedings of the 16th international conference on cognitive modeling* (pp. 162–167).

- Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, 12(1), 97–136.
- Wolfe, J. M. (2021). Guided search 6.0: An updated model of visual search. *Psychonomic Bulletin & Review*, 28(4), 1060–1092.

Integrated Cognitive Model Framework for Analogical Reasoning

¹Alexander R. Hough (alexander.hough.1@us.af.mil)
²Othalia Larue (othalia.larue@parallaxresearch.org)
¹Christopher Myers (christopher.myers.29@us.af.mil)
¹Olivia Leung (olivia.leung@us.af.mil)

¹Air Force Research Laboratory, Wright-Patterson AFB
²Parallax Advanced Research, Beavercreek, OH 45431 USA

Abstract

Analogical reasoning is a core cognitive process. Models have implemented features of analogical reasoning with varied success. Successful models approximate analogical mapping but are not focused on cognitive plausibility. Here, we present and demo an integrated model framework leveraging a component model of analogy (Structure Mapping Engine) to extend a cognitive architecture (ACT-R) for cognitively plausible analogical mapping to inform higher-order cognition.

Keywords: ACT-R; Structure mapping engine; Analogy; Similarity; Generalizability; Cognitive modeling

Introduction

Analogical reasoning is important for understanding (Hough & Gluck, 2019) and involves the cognitive process of mapping: finding a common or similar relational structure between systems (e.g., concepts) (Gentner, 1983). For instance, finding commonalities between an atom and a solar system (i.e., force causes one body to revolve around another). Mapping is a main component in models of analogy, but is less common in cognitive architectures (Gentner & Forbus, 2011). Here, we present a proof-of-concept leveraging strengths of a cognitive architecture and model of analogy. We drew inspiration from the path mapping model (i.e., PM) (Salvucci & Anderson, 2001) implemented in the ACT-R cognitive architecture (Anderson, 2007) and the structure mapping engine (i.e., SME) (Falkenhainer et al., 1989; Forbus et al., 2017).

PM used direct word matches or partial matching based on provided similarity values to complete a series of retrievals. Retrieved chunks are "chained" to represent a path from an object to its root (i.e., highest level) relation. The mapping process then compared a source path (e.g., electromagnetism causes electron to revolve) to one in an identified target (e.g., gravity causes planet to revolve).

SME is a computational implementation of Structure-mapping theory (i.e., SMT) (Gentner, 1983). The theory suggests mapping new (i.e., target) and existing (i.e., base) knowledge structures underlies experiential learning. While mapping, one assumes relations in the base also exist in the target. There is a preference for relations (e.g., sun is larger than planet) over attributes (e.g., sun is yellow), and interrelated or second-order relations (e.g., sun has more mass/gravity than planet *causing* planet to revolve around sun) over lower-order (e.g., sun is hotter than planet). This preference for greater coherence is referred to as the systematicity principle and it guides the mapping process. In SMT,

mappings are restricted to one-to-one correspondence (i.e., one item in base maps to only one in target) and are structurally consistent (i.e., if second-order relations map, then their first-order must too). SME incorporates these features, generates matches between objects and relations (i.e., match hypotheses), and calculates structural evaluation scores.

Proof-of-concept Cognitive Model

The proof-of-concept framework includes a model implemented in ACT-R leveraging SME as an external module to map knowledge structures and quantify similarity. We demo initial capabilities using sport representations (Figure 1).

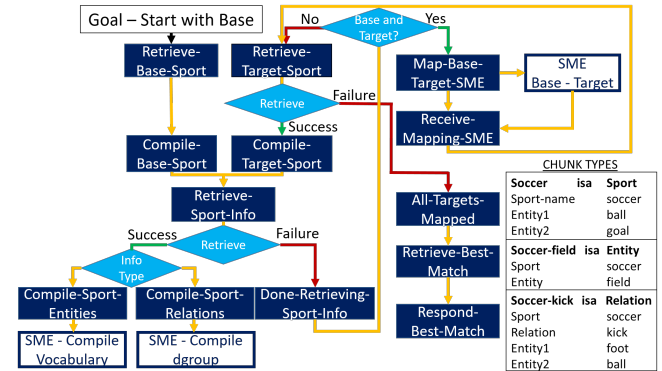


Figure 1: Proof-of-concept demo model processes and chunk types with SME interactions outlined in blue.

The demo model was provided knowledge implemented as chunks in declarative memory (i.e., DM) that represent sports. There are three chunk types: 1) Sport chunks represent the sport and contain two entities that "best" represent it, 2) entity chunks contain single entities, and 3) relation chunks contain a relation and entities ordered to represent their roles. The model is given a sport (i.e., base) and productions guide the model through a series of retrievals to compile all information (i.e., chunks) for that sport. SME compiles this information into two complete structured representations: 1) dgroup describes systems as a list of entities and predicates, and 2) a common vocabulary file. After all base sport chunks have been retrieved, the model finds another sport in memory (i.e., target), compiles it, and maps the base and target. The mapping production passes these structured representa-

tions to SME and it computes: 1) match hypotheses or pairs of items that align between base and the target, 2) structural evaluation quantifies the degree of match, and 3) candidate inferences extrapolate information from base to target. The demo model uses a fraction of structural evaluations and normalizes them to 1. Normalized values are passed to ACT-R and are multiplied by default chunk similarity values, which default to -1 due to max similarity and difference defaults. After a target sport is mapped and similarity values are set, targets are sequentially retrieved, compiled, and mapped until all have been mapped to the base sport. Then the model uses similarity values to retrieve the best matching target.

Demo Model Results

We tested the demo model with knowledge about 8 different sports: soccer, water-polo, baseball, kickball, ping-pong, tennis, volleyball, and badminton (Figure 2).

(Soccer ball goal)	(Water-polo ball goal)	(Baseball ball bat)	(Kickball ball foot)
(entity field)	(entity pool)	(entity field)	(entity field)
(entity player)	(entity player)	(entity bases)	(entity bases)
(entity goalie)	(entity goalie)	(entity fielder)	(entity fielder)
(entity foot)	(entity hand)	(entity pitcher)	(entity pitcher)
(entity hand)	(entity game)	(entity batter)	(entity batter)
(entity game)	(play game pool)	(entity hand)	(entity hand)
(play game field)	(use player hand)	(entity glove)	(entity game)
(use player foot)	(use goalie hand)	(entity game)	(play game field)
(use goalie hand)	(throw hand ball)	(play game field)	(use batter foot)
(kicks foot ball)	(score ball goal)	(use batter bat)	(use fielder hand)
(score ball goal)	(offense player goal)	(use fielder glove)	(use pitcher hand)
(offense player goal)	(defend goalie goal)	(use pitcher hand)	(run batter bases)
(defend goalie goal)		(run batter bases)	(hits foot ball)
		(hit bat ball)	(throw hand ball)
		(throw hand ball)	(catch ball hand)
		(catch ball glove)	(score batter homebase)
		(score batter homebase)	(offense batter pitcher)
		(offense batter pitcher)	(offense batter fielder)
		(offense batter fielder)	(defend pitcher batter)
		(defend pitcher batter)	(defend fielder batter)
		(defend fielder batter)	

Figure 2: Example sport representations

Each has a single sport chunk with two entities (e.g., soccer ball goal), several entities (e.g., entity field), and several relation chunks with two entities (e.g., play game ball). Relation chunks were designed to be general (e.g., play and use), the order of entities shows their role (e.g., play game field means the game is played on a field), and some relations are interdependent (e.g., use player foot and kicks foot ball).

Figure 3 shows match hypotheses between soccer and badminton. Entities are matched based on their roles within rela-

player		player
foot		racket
(USE player foot)		(USE player racket)
use		use
hand		racket
(USE goalie hand)		(USE player racket)
game		game
field		court
(PLAY game field)		(PLAY game court)
play		play
(OFFENSE player goal)		(OFFENSE player player)
offense		offense
ball		shuttlecock
goal		bounce
(SCORE ball goal)		(SCORE shuttlecock bounce)

Figure 3: Match hypotheses between soccer and badminton

tions. For instance, foot and hand matched with racket (use), field matched with court (play), ball is matched with shuttlecock (score), and goal is matched with bounce (score).

In Figure 4, we show chunk similarities in ACT-R post-SME interaction. When soccer was the base sport, water-polo was the best match (i.e., most likely to be retrieved), followed by baseball and kickball. We note match similarities are low, which we suspect is due to our normalization procedure, which we plan to improve.

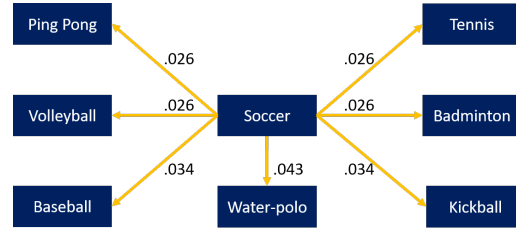


Figure 4: Similarities for soccer after integration in ACT-R.

Discussion

We presented a proof-of-concept framework: A cognitive model implemented in ACT-R that leveraged SME as an external module to map structures and set similarity. For the demo, we provided representations of sports to ACT-R DM and through a series of processes, the model successfully mapped sport representations to the base and responded with the "best" match. Through the addition of a SME module we provided ACT-R with a different way to learn by experience: abstractly comparing new knowledge to existing knowledge (i.e. analogy). Using ACT-R allows easy exploration of interactions with other cognitive phenomena like fatigue and workload. In future work, we plan to refine the framework and explore its capabilities to facilitate things like: 1) analogical transfer for realistic situations and 2) situation awareness in multi-modal decision making tasks.

Acknowledgments

This research was supported by the U. S. Air Force Research Laboratory's 711th Human Performance Wing, Cognitive Models Branch. Contents have been reviewed and approved for public release (Distribution A), case number: AFRL-2023-2153. The views expressed are those of the authors and do not reflect the official guidance or position of the United States Government, the U.S. Department of Defense, the U.S. Air Force, or any of their subsidiaries or employees.

References

- Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press.
- Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial intelligence*, 41(1), 1–63.

- Forbus, K. D., Ferguson, R. W., Lovett, A., & Gentner, D. (2017). Extending sme to handle large-scale cognitive modeling. *Cognitive Science*, 41(5), 1152–1201.
- Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2), 155–170.
- Gentner, D., & Forbus, K. D. (2011). Computational models of analogy. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2(3), 266–276.
- Hough, A. R., & Gluck, K. A. (2019). The understanding problem in cognitive science. *Advances in Cognitive Systems*, 8, 13–32.
- Salvucci, D. D., & Anderson, J. R. (2001). Integrating analogical mapping and general problem solving: the path-mapping theory. *Cognitive Science*, 25(1), 67–110.

Seeing What You Believe: Cognitive Mechanisms of Flexible Integration of Priors in Visual Decisions

Gabriela Iwama (gabriela.iwama@uni-tuebingen.de)

Hertie Institute for Clinical Brain Research
Tübingen, Germany

Randolph Helfrich (Randolph.Helfrich@med.uni-tuebingen.de)

Hertie Institute for Clinical Brain Research
Tübingen, Germany

Abstract

Individual beliefs and expectations shape how we perceive our surroundings. In a complex and ever-changing world, prior beliefs need to be flexibly integrated and updated during the decision process. The goal of this study was to dissociate the cognitive mechanisms involved in the integration of learned beliefs in visual decisions under uncertainty. Combining two well-established cognitive models – Hierarchical Gaussian Filtering and the Drift Diffusion Model – this study replicates the well-established finding that priors bias the starting point of evidence accumulation and the rate of evidence accumulation. Critically, the results also reveal a decrease in non-decision time and an increase in the amount of evidence accumulated when the belief was congruent with the outcome. Collectively, these results provide evidence for the hypothesis that prior beliefs are implemented into visual decisions through distinct cognitive mechanisms.

Keywords: Uncertainty; Prior; Belief Update; Visual Decision; Modeling

Introduction

Beliefs and expectations, or priors, shape our perception of the environment (Gold & Stocker, 2017). In an ever-changing world, priors must be flexibly and continuously integrated into sensory decision processes to guide adaptive behavior. Nonetheless, its underlying cognitive mechanisms are not well understood. The Drift Diffusion Model (DDM) is a widely used model for studying visual decision-making (Gold & Shadlen, 2007; Ratcliff & Rouder, 1998). Previous studies have shown that priors can increase the starting point of evidence accumulation and the drift rate (Dunovan & Wheeler, 2018; Dunovan, Tremel, & Wheeler, 2014; Thakur, Basso, Ditterich, & Knowlton, 2021). However, these studies often overlook the potential effects of priors on decision threshold and non-decision time parameters.

The goal of this study was to dissociate the effects of priors on multiple cognitive mechanisms in visual decisions. Specifically, I tested how the strength of prior beliefs affects: (a) the integration of momentary sensory evidence; (b) the amount of evidence required to decide; (c) pre-stimulus presentation processes; and (d) non-evidence accumulation effects.

Method

Eight participants completed a behavioral task that required tracking the cue validity across trials and using the cue information flexibly. The task combined a reversal learning and a random dot motion discrimination task (Figure 1) and involved three main decisions per trial: cue choice, confidence, and motion direction. The identity of the cue was determined by its color, and the cue direction was displayed with a predetermined but unknown validity. Each participant completed a maximum of 320 trials, which were divided into informative and non-informative blocks. The interval between blocks varied from 15-30 trials. The validity of the cue in informative blocks was set at 80% or 30%, while the validity of the cues in non-informative blocks was set at 30% for both cues. At the end of each trial, participants received rewards for their motion judgment and cue choice. The reward for the cue choice depended on the confidence reported earlier in the trial.

Results

Belief Tracks Subjective Confidence and Accuracy

To evaluate the validity of the estimated belief, we tested whether belief strength is associated with confidence and the true contingency. Belief strength was higher when participants reported high confidence in their cue choice ($t(7) = 5.31$, $p = .001$). Furthermore, when belief strength was higher, participants chose the best cue for the block more often than when belief strength was low ($t(7) = 24.52$, $p < .001$). Altogether, these findings provide evidence of validity for trial-wise measures of belief strength.

Multiple Cognitive Mechanisms of Prior Integration

The results show that the strength of belief affects various aspects of visual decision-making (Figure 2). When the cue was valid, stronger beliefs increased the drift rate (rate of evidence accumulation), increased the response bias towards the direction indicated by the cue, increased the threshold (amount of evidence needed to reach a decision), and reduced non-decision time (secondary processes involved in the decision execution). In contrast, when the cue was invalid, stronger beliefs had the opposite effects on these parameters. Overall, belief strength modulates the DDM parameters depending on the accuracy of the belief for a given trial.

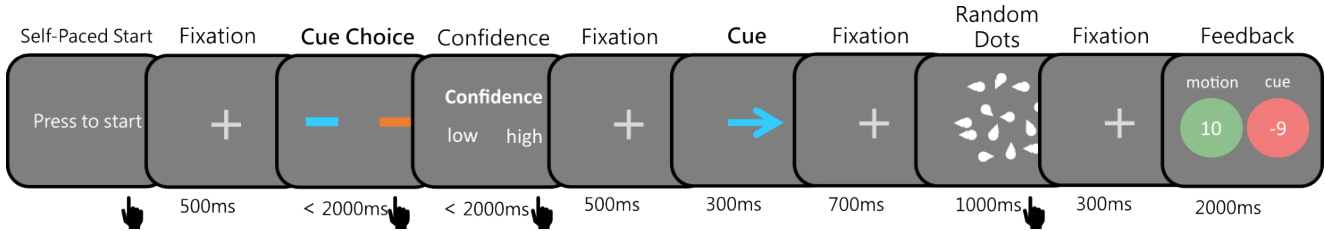


Figure 1: **Behavioral task.** During each trial, participants made three decisions: selecting a cue, indicating their level of confidence in the cue, and indicating the direction of the motion they observed. The motion stimulus consisted of a series of moving dots, with a small subset of them (up to 20%) moving in a specific direction. At the end of each trial, participants received two rewards: one for their judgment of the motion and another for their choice of cue.

Discussion

The main goal of this study was to behaviorally dissociate the effect of belief on visual decision-making using trial-wise estimates of belief strength. The effects on drift rate reflect the ramping of activity in parietal regions that scale with the strength of evidence (Hanks et al., 2015). In the present study, the effect of belief strength on the drift rate is congruent with biased evidence sampling driven by the post-decisional confidence (Rollwage et al., 2020).

The effects on the starting point are usually interpreted as a choice response bias (Dunovan et al., 2014; Dunovan & Wheeler, 2018). The origin of such biases in the starting point can be a result of a tendency to accept belief-congruent evidence, motor preparation (de Lange, Rahnev, Donner, & Lau, 2013), or even an increase in the sensitivity of low-level sensory representations before stimulus presentation (Kok, Failing, & de Lange, 2014). Although DDM does not dissociate between these subcomponents, it is possible to constrain them neurophysiologically (Harris & Hutcherson, 2022).

Effects on the evidence accumulation threshold are associated with speed-accuracy trade-offs (Bogacz, Wagenmakers, Forstmann, & Nieuwenhuis, 2010). In the present study, we observed an effect of belief on decision threshold, suggesting that belief strength increases the amount of evidence that needs to be accumulated when the belief is congruent with visual input. This effect might be caused by a compensation mechanism to maintain high accuracy when the belief is invalid for a particular trial.

The non-decision time parameter has often been neglected in the literature. Despite its marginalization, it might reflect important processes. For example, the latency of N200 potentials, which is associated with the encoding of visual stimuli, seems to track non-decision times (Nunez, Gosai, Vandekerckhove, & Srinivasan, 2019). The effect of non-decision time found in this study could emerge from the evidence-encoding onset, evidence accumulation onset, or post-decision motor execution time (Kelly, Corbett, & O'Connell, 2021). In the future, we will leverage the temporal dynamics of decision-making using neurophysiological recordings to constrain and dissociate these parameters (Harris & Hutcherson, 2022).

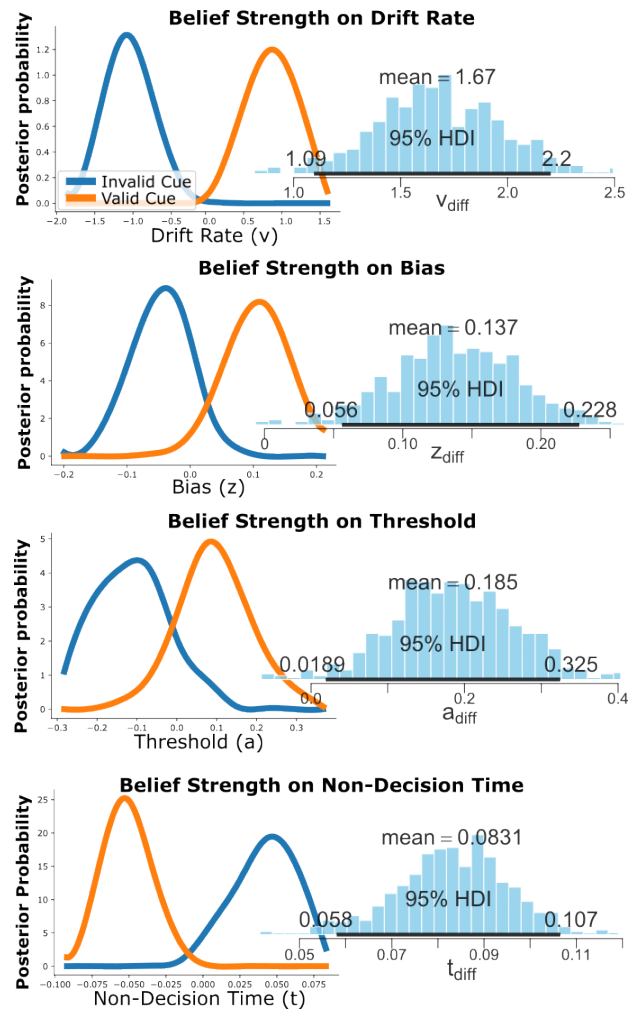


Figure 2: **Effect of belief strength on visual decision-making.** Left: Posterior distributions of within-subject effects of belief strength for each DDM group-level parameter are shown according to cue validity. Right: 95% Highest Density Interval (HDI) of the difference between posterior distributions for each parameter.

Acknowledgments

This work was supported by the International Max Planck Research School for the Mechanisms of Mental Function and Dysfunction (IMPRS-MMFD), the Hertie-Foundation (Network for Excellence in Clinical Neuroscience), the Medical Faculty of Tuebingen (JRG Plus Program) and the German Research Foundation's (DFG) Emmy Noether Programme (HE8329/2-1).

Thakur, V. N., Basso, M. A., Ditterich, J., & Knowlton, B. J. (2021, August). Implicit and explicit learning of Bayesian priors differently impacts bias during perceptual decision-making. *Scientific Reports*, 11(1), 16932.

References

- Bogacz, R., Wagenmakers, E.-J., Forstmann, B. U., & Nieuwenhuis, S. (2010). The neural basis of the speed–accuracy tradeoff. *Trends in Neurosciences*, 33(1), 10–16.
- de Lange, F. P., Rahnev, D. A., Donner, T. H., & Lau, H. (2013). Prestimulus oscillatory activity over motor cortex reflects perceptual expectations. *Journal of Neuroscience*, 33(4), 1400–1410.
- Dunovan, K., Tremel, J. J., & Wheeler, M. E. (2014, August). Prior probability and feature predictability interactively bias perceptual decisions. *Neuropsychologia*, 61, 210–221.
- Dunovan, K., & Wheeler, M. E. (2018, September). Computational and neural signatures of pre and post-sensory expectation bias in inferior temporal cortex. *Scientific Reports*, 8(1), 13256.
- Gold, J. I., & Shadlen, M. N. (2007). The Neural Basis of Decision Making. *Annual Review of Neuroscience*, 30(1), 535–574.
- Gold, J. I., & Stocker, A. A. (2017). Visual Decision-Making in an Uncertain and Dynamic World. *Annual Review of Vision Science*, 3(1), 227–250.
- Hanks, T., Kopec, C. D., Brunton, B. W., Duan, C. A., Erlich, J. C., & Brody, C. D. (2015). Distinct relationships of parietal and prefrontal cortices to evidence accumulation. *Nature*, 520(7546), 220–223.
- Harris, A., & Hutcherson, C. A. (2022). Temporal dynamics of decision making: A synthesis of computational and neurophysiological approaches. *WIREs Cognitive Science*, 13(3), e1586.
- Kelly, S. P., Corbett, E. A., & O'Connell, R. G. (2021). Neurocomputational mechanisms of prior-informed perceptual decision-making in humans. *Nature Human Behaviour*, 5(4), 467–481.
- Kok, P., Failing, M. F., & de Lange, F. P. (2014). Prior expectations evoke stimulus templates in the primary visual cortex. *Journal of Cognitive Neuroscience*, 26(7), 1546–1554.
- Nunez, M. D., Gosai, A., Vandekerckhove, J., & Srinivasan, R. (2019). The latency of a visual evoked potential tracks the onset of decision making. *NeuroImage*, 197, 93–108.
- Ratcliff, R., & Rouder, J. N. (1998, September). Modeling Response Times for Two-Choice Decisions. *Psychological Science*, 9(5), 347–356.
- Rollwage, M., Loosen, A., Hauser, T. U., Moran, R., Dolan, R. J., & Fleming, S. M. (2020). Confidence drives a neural confirmation bias. *Nature Communications*, 11(1), 2634.

Predicting Human Interleaving Time in Semi-Automated Vehicles

Christian P. Janssen (c.p.janssen@uu.nl)

Experimental psychology, Heidelberglaan 1, 3573 SH Utrecht, The Netherlands

Leonard Praetorius (leonardpraetorius@googlemail.com) and Jelmer P. Borst (j.p.borst@rug.nl)

Bernoulli Institute, Nijenborgh 9, 9747 AG Groningen, The Netherlands

Keywords: semi-automated driving; process model; interruptions; attention

The Case for Modeling Transitions of Control

Current commercial cars can assist the human driver with various driving tasks (i.e., SAE levels 1, 2; SAE Int., 2018). The next generation of semi-automated vehicles is expected to take over more driving tasks, giving the human driver more a monitoring role. Specifically, in SAE level 3, the car can drive independently in some operational domains, but can prompt the driver for help, in which case they *must* assist. This is a “transition of control” process from car to human.

Transitions of control are interesting to describe using cognitive models, as such models can help to understand the cognitive process that the human driver goes through. In addition, cognitive models can aid in the design of the vehicle and its interfaces (cf. e.g., Oulasvirta, 2019; Salvucci, 2009). For example, by quantifying how long it might take to transition control, and in what modality information can be best presented to avoid user overload. Indeed, there is already a wide set of models for human-automated vehicle interaction (Janssen, Baumann, Oulasvirta, Iqbal, Heinrich, 2022)

The current paper presents our first steps in developing formal, cognitive model-based, predictions for SAE level 3 transitions of control scenarios. Based on a theoretical model of attention interleaving, we developed a tool that breaks down transitions of control in five steps. We then use the tool in combination with empirical literature to predict response times for different steps of the interleaving process.

Theoretical Model: Interruption Handling

Transitions of control are mostly studied as relatively fast responses. For example, in a meta-review of 129 studies by Zhang et al (2019), the time between an alert and the first human action (e.g., steering movement, brake press) is reported. These studies reported fast response times ($M = 2.7s$, $SD = 1.5s$; only one study had a response time over 8.5s). This is consistent with design-oriented research that suggests having alerts 5-8 s before a critical incident (e.g., Gold et al., 2013). In effect, this describes transitions of control as a fast *task switch* in which almost all attention is dedicated to one task at a time.

As argued elsewhere in detail (Janssen et al., 2019), an alternative is to describe transitions of control as *task interruption handling*. In such scenarios, people might not immediately give up on their original (non-driving) task, but instead take time to transition from non-driving to driving. It

is motivated by a.o. user needs for automated vehicles in which performance on non-driving tasks is central (e.g., Pfleging et al., 2016) and the assumption that transitions of control will become rare, yet critical events. Moreover, alert processing might be limited under automated vehicle conditions (Van der Heiden et al., 2022), and models from the ICCM community have shown that alerting people at the wrong moment can have detrimental consequences for later tasks (e.g., Borst et al., 2015; Janssen et al., 2012). Therefore, people might want to defer giving up on a task until they reach a “natural breakpoint”.

Janssen et al. (2019) describe transitions of control through the lens of interruption handling (cf. other domains, e.g., Boehm-Davis & Remington, 2009; Borst, et al., 2015). The core of the model is that humans go through a series of stages before fully taking control of a vehicle, as illustrated in Figure 1. While the driver is performing an original task (stage 0), they notice an alert (stage 1), which leads to disengage briefly with the original task (stage 2), to then orient to the driving task (stage 3). If time allows, they interleave between rounding off their original task and orienting to the driving task, before fully suspending (stage 4), and then taking physical control of the vehicle (stage 5).

The first studies indeed suggest that drivers take longer to take control if they are given the opportunity by an early “pre-alert” (Borojeni et al, 2018; Van der Heiden et al., 2017) and that they go through some of these interruption stages (Nagajaru et al., 2021). However, to understand the process deeper, a model implementation is needed of all stages.

Mapping Empirical Data to Processes Stages

As a first modeling step, we mapped the empirical literature to the processing stages proposed in Janssen et al. (2019). Specifically, we analyzed all studies in the meta-

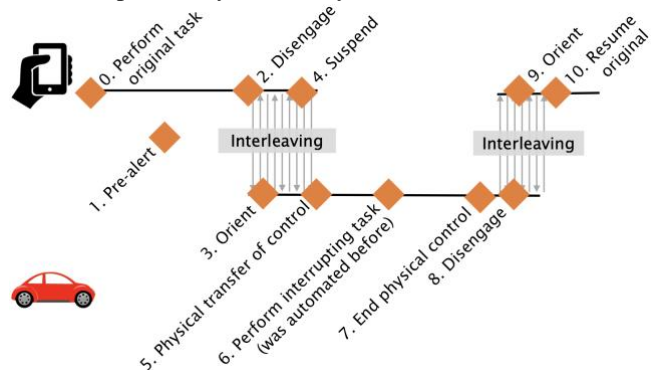


Figure 1: Stages in Janssen et al (2019)'s interleaving model

review of Zhang et al. (2019) to identify which details on response times map to which processing stage. We also annotated the characteristics of each study, such as alert modality and modality of non-driving task.

The software then predicts the best fitting distribution of response times for each stage. In the fitting process, we assume that response time data has positive skewness (cf. Ratcliff, 1993) and that there are no negative response times.

The tool can then be used to visualize likely response times for each stage of the take-over process, and how this varies by study characteristics. As an example, Figure 2 shows the expected distribution times for the interval between initial alert and stages 2, 3, 4, and 5 respectively, with different distributions based on alert modalities. For some stage (e.g., stage 4, bottom-left) there is no data on specific modalities. The Figure also highlights that having a bi-modal alert (red data) tends to systematically impact the early stages of alert processing (stages 2, 3, and 4), but less the eventual response time (stage 5). This is consistent with more general literature that shows the benefits of multi-modal alerts, but adds a level of model precision by quantifying the times for each stage.

General Discussion

The presented model is only a first step towards understanding the process of transition of control in more detail. We implemented a theoretical model in such a way that it can map to existing literature and generate process model related predictions. These more detailed predictions can be used to calibrate even more detailed cognitive models. More generally, the tool can be used to guide design decisions and estimate in what stage a design intervention (such as alert modality) might impact performance most.

References

Boehm-Davis, D. A., & Remington, R. (2009). Reducing the disruptive effects of interruption: A cognitive framework for analysing the costs and benefits of intervention strategies. *Accident Analysis & Prevention*, 41(5).

Borojeni, S. S., Weber, L., Heuten, W., & Boll, S. (2018.). From reading to driving: priming mobile users for take-over situations in highly automated driving. In *Proceedings Mobile HCI*.

Borst, J. P., Taatgen, N. A., & van Rijn, H. (2015). What makes interruptions disruptive? A process-model account of the effects of the problem state bottleneck on task interruption and resumption. In *Proceedings of CHI*.

Gold, C., Damböck, D., Lorenz, L., & Bengler, K. (2013). "Take over!" How long does it take to get the driver back into the loop?. In *Proceedings HFES*. Los Angeles, CA: Sage Publications.

van der Heiden, R. M., Iqbal, S. T., & Janssen, C. P. (2017). Priming drivers before handover in semi-autonomous cars. In *Proceedings of CHI*.

van der Heiden, R. M., Kenemans, J. L., Donker, S. F., & Janssen, C. P. (2022). The effect of cognitive load on auditory susceptibility during automated driving. *Human factors*, 64(7).

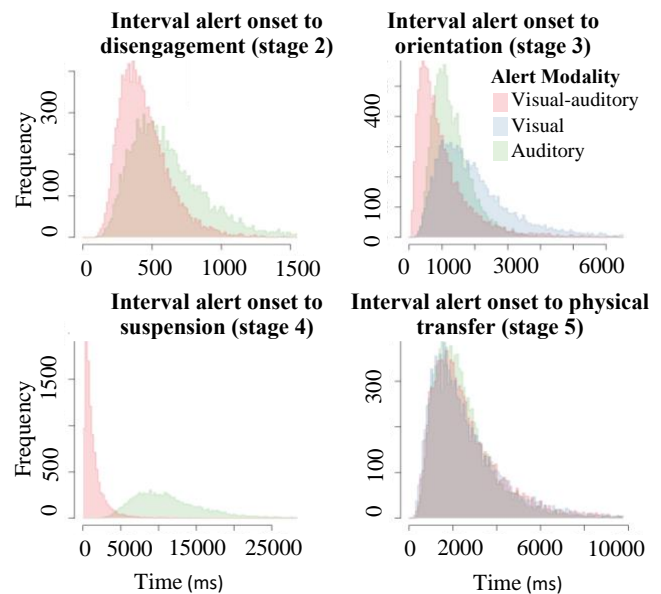


Figure 1: Distribution of the simulated data considering different alert modalities.

Janssen, C. P., Baumann, M., Oulasvirta, A., Iqbal, S. T., & Heinrich, L. (2022). Computational Models of Human-Automated Vehicle Interaction (Dagstuhl Seminar 22102). In *Dagstuhl Reports* (Vol. 12, No. 3). Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Janssen, C. P., Brumby, D. P., & Garnett, R. (2012). Natural break points: The influence of priorities and cognitive and motor cues on dual-task interleaving. *Journal of Cognitive Engineering and Decision Making*, 6(1).

Janssen, C. P., Iqbal, S. T., Kun, A. L., & Donker, S. F. (2019). Interrupted by my car? Implications of interruption and interleaving research for automated vehicles. *International Journal of Human-Computer Studies*, 130.

Nagaraju, D., Ansah, A., Ch, N. A. N., Mills, C., Janssen, C. P., Shaer, O., & Kun, A. L. (2021). How will drivers take back control in automated vehicles? A driving simulator test of an interleaving framework. In *Proceedings of AutomotiveUI*.

Oulasvirta, A. (2019). It's time to rediscover HCI models. *Interactions*, 26(4).

Pfleging, B., Rang, M., & Broy, N. (2016). Investigating user needs for non-driving-related activities during automated driving. In *Proceedings MUM/*

Ratcliff, R. (1993). Methods for dealing with reaction time outliers. *Psychological Bulletin* 114.

SAE International (2018). J3016_201806: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. Retrieved from https://www.sae.org/standards/content/j3016_201806/

Salvucci, D. D. (2009). Rapid prototyping and evaluation of in-vehicle interfaces. *ACM TOCHI*, 16(2).

Zhang, B., De Winter, J., Varotto, S., Happee, R., & Martens, M. (2019). Determinants of take-over time from automated driving: A meta-analysis of 129 studies. *Transportation research part F: traffic psychology and behaviour*, 64.

Modelling the Role of Hanja in the Korean Mental Lexicon: A Second Tier of Spreading Activation

Stephen Jones (s.m.jones@rug.nl)

Bernoulli Institute, University of Groningen
PO Box 72, 9700AB Groningen, Netherlands

Yoolim Kim (ykim6@wellesley.edu)

Wellesley College, 108 Central Street
Wellesley, MA 02481 USA

Abstract

This poster presents interim results from an ACT-R-based statistical model of a series of lexical decision experiments in Korean. The model uses two tiers of spreading activation, one of which represents semantic distance, and the other of which represents the effect of the *Hanja* writing system on the mental lexicon. Modelling the data requires assumptions to be made about the relationship between the tiers of spreading activation, and about the method of computing semantic association. The poster is supported by an interactive browser interface that allows participants to vary these assumptions, as well as the standard ACT-R spreading activation parameters, and explore how this impacts the model fit.

Keywords: ACT-R; lexical decision; priming; spreading activation;

Introduction

Korean uses two writing systems: *Hangul*, an alphabet of individual letters which are written in syllabic groups; and *Hanja*, an ideographic system that uses traditional Chinese characters. Words with a Chinese origin form around two-thirds of the Korean lexicon, and both *Hangul* and *Hanja* written forms are available for these words¹. Figure 1 shows how the word *hakkyo* ‘school’ is represented in the two systems.

(a) 학교 (b) 學校

Figure 1: *Hakkyo* ‘school’ as (a) *Hangul* and (b) *Hanja*

Hangul is the first writing system that children learn and is the more frequently used system overall. However, *Hanja* are often used in South Korea to disambiguate homonyms or in signage and some knowledge of *Hanja* is an expected part of literacy. The South Korean Ministry of Education maintains a list of 1800 high-frequency *Hanja* that are learned as part of the school curriculum: 900 at middle school and a further 900 at high school.

Y. Kim (2019) investigated how knowledge of *Hanja* is stored in memory, looking specifically at its impact on the organisation of the mental lexicon. She carried out two primed lexical decision experiments, one intra-modally with primes presented in *Hangul*, and a one cross-modally with primes

presented aurally. In both experiments, the targets were presented visually in *Hangul*. Stimuli were drawn from sets of four disyllabic words, comprising three primes, categorised as *Unrelated*, *Direct* or *Indirect*, and one target. In each stimulus set, the *Unrelated* prime had no phonological or semantic relationship to the target. The speed of retrieval of the target following the *Unrelated* prime was taken as the baseline retrieval time for the word. The *Direct* prime had a semantic, but not a phonological relationship to the target. Crucially, the *Indirect* prime had neither a semantic nor a phonological relationship with the target. However, the *Indirect* and *Direct* primes in a set contained *Hanja* from the same phonological cohort, and thus the *Hanja* writing system provided an indirect connection between prime and target.

Each participant saw each target only once, preceded by one of the three primes. A Latin square design was used, to balance the conditions across and between participants. Half of the trials were foils, consisting of a random lexical prime paired with a non-word target. In the intra-modal experiment the prime was presented for 300ms, followed by a blank screen for 250ms, before the target was presented for 300 ms. In the cross-modal experiment, the target was presented immediately after the audio signal had finished and again remained on the screen for 300ms. Participants then had 1500ms to give their lexical decision before the next trial.

Y. Kim’s results present a mixed picture. The expected semantic priming effect was not found for all *Direct*–target pairs. However, for those stimulus sets where direct semantic priming was observed, there was also a small but statistically significant priming effect for the *Indirect*–target pairs. Y. Kim concludes that the facilitation observed with an *Indirect* prime arises because of activation of the entire *Hanja* cohort associated with the morphemes of the prime. This activation includes the *Direct* prime, from which there is further spreading of activation because of this word’s semantic association with the target (Figure 2).

The model

The model assumes that variations in decision latency for a particular target across the three conditions arise primarily from differences in the activation of lexical memory. This, following Y. Kim, is assumed to arise from semantic similarity which, for the *Indirect*–target pairs, is moderated by spreading activation between the *Indirect* prime and its asso-

¹Unlike Kanji in Japanese, Korean does not use *Hanja* to represent words of native Korean origin.

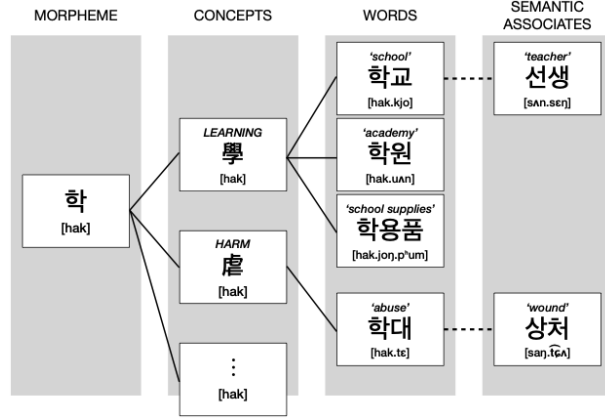


Figure 2: Proposed activation spreading (Y. Kim, 2019)

ciated *Direct* prime. Because the primes were presented for a duration that allowed conscious perception (300ms visually or at a normal speaking rate aurally), priming is also assumed to be conscious. In ACT-R terms, the model assumes when the prime is read or heard, its lexical entry is retrieved and a chunk containing information including its meaning and its associated Hanja with is created and held in the Imaginal buffer. Spreading activation from this chunk is then available when the target appears on the screen.

For the *Direct* prime, we assume that there is one tier of spreading activation, which derives from semantic association. The model's level of semantic activation is based on cosine similarity measures derived from the Korean vector set (Grave, Bojanowski, Gupta, Joulin, & Mikolov, 2018) available for the Fasttext word representation system (Bojanowski, Grave, Joulin, & Mikolov, 2016). However, an alternative approach using Korean WordNet (J. Kim, 2018) and the WN-LEXICAL module developed by Emond (2006) is also possible.

For the *Indirect* prime, we propose that there are two tiers of spreading activation. The first of these is calculated through symbolic representations of Hanja knowledge, and the second tier is derived from semantic association as for the *Direct* prime. Spreading activation in ACT-R is determined by equation (1), where i represents the chunk receiving spreading activation, j represents chunks held in the slots of the chunk in a buffer that produces spreading activation, and k represents the buffers that produce spreading activation.

$$S_i = \sum_k \sum_j W_{kj} S_{ji} \quad (1)$$

In the model, we assume that only the Imaginal buffer is spreading activation and so $k = 1$. Parameter j is the cohort size of a syllable, the number of Hanja that are associated with a particular Hangul morpheme. This varies not only per Hangul, but also with the level of knowledge of the speaker. For example, a speaker with Middle School knowledge of Hanja can associate the morpheme [pjaŋ] with 3

Hanja, whereas a speaker with High School knowledge will have learned two further Hanja, giving a cohort size of 5. For the morpheme [sa], the corresponding cohort sizes are 4 and 12 respectively.

The strength of association S_{ji} is determined by the size of the fan of j , that is, the number of words in the vocabulary that contain Hanja j . Initial assumptions for this parameter take into account lexical frequency as well as the level of Hanja knowledge: work to refine this element of the model is ongoing.

Alongside the poster, a Shiny app (Chang et al., 2023) has been produced that shows how the model fits with experimental data. This allows interactive exploration of the effect of parameter values on the overall fit.

References

- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.
- Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., ... Borges, B. (2023). shiny: Web application framework for r [Computer software manual]. Retrieved from <https://shiny.rstudio.com/> (R package version 1.7.4.9002)
- Emond, B. (2006). WN-LEXICAL: An ACT-R module built from the WordNet lexical database. In *Proceedings of the Seventh International Conference on Cognitive Modeling* (pp. 359–360).
- Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the international conference on language resources and evaluation (lrec 2018)*.
- Kim, J. (2018). *Korean WordNet*. Retrieved 2023-03-29, from <http://wordnet.kaist.ac.kr/>
- Kim, Y. (2019). *The mental representation of Hanja*. Doctoral dissertation, University of Oxford.

Errors Are The Stepping Stones to Learning: Trial-by-Trial Modeling Reveals Overwhelming Evidence for Mediator Retrievals of Previous Errors in Memory Consolidation

Bridget Leonard (bll313@uw.edu)

Department of Psychology, University of Washington, Seattle, WA 98195 USA

Andrea Stocco (stocco@uw.edu)

Department of Psychology, University of Washington, Seattle, WA 98195 USA

Abstract

Recent studies suggest that errors facilitate learning in certain conditions. Despite this, reinforcement paradigms dominate learning methods, subscribing to the narrative that errorless learning is the foundation of an ideal learning environment. If we continue to view learning from this restrictive perspective, we may fail to capture and apply the benefits of errors. In this paper, we investigate two potential mechanisms of learning from errors. Participants ($N = 61$) learned word pairs in either a study or error trial before taking a final test. Supporting past error learning literature, errors before a study opportunity led to better performance on a final test. Differences in reaction times between conditions support the theory that errors increase learning by acting as a mediator, or secondary cue, to the correct answer on subsequent tests. Individual differences in model fit using log-likelihood trial-by-trial calculations solidified support for the mediator method.

Keywords: learning from errors; memory; retrieval; elaboration; mediation; computational models; ACT-R

Introduction

Many people believe that ideal learning is errorless. Errors are often viewed as detrimental to learning with the expectation that they will interfere with future retrievals of correct information (Ceraso, 1967; Anderson & Reder, 1999). These concepts stem from studying more procedural behavior where learning is a constant reinforcement process; thus, errorless learning minimizes opportunities to reinforce errors. However, errors committed with high confidence are more likely to be corrected (Butterfield & Metcalfe, 2001). It appears reinforcement paradigms do not generalize well to learning by memorization compared to more procedural learning. Successful memorization requires the ability to retrieve information; errors may increase this type of learning by promoting better encoding and thus, more successful recall. We may be able to enhance the most common form of learning for humans by critically examining how and when errors are beneficial to memorization and subsequent retrieval.

Derivatives of associative learning tasks have introduced paradigms to study post-error learning, or improvements in subsequent recall after a memory error. In the pretesting paradigm, participants generate an answer before studying it; with no prior exposure to the correct answer, participants are likely to generate an incorrect response (Mera, Rodriguez, & Marin-Garcia, 2021). Although this paradigm encourages errors, pretesting information is more beneficial than simply studying it. Many studies have confirmed this

finding, extending the benefits of pretesting to real-world materials (Kornell, 2014), educational materials (Kapur & Bielaczyc, 2012), and older adults (Cyr & Anderson, 2015).

Although post-error learning is now well-documented, an investigation into its underlying mechanisms is sparse. Two prominent theories have arisen out of this research; the elaborative hypothesis and the mediator hypothesis. To go beyond speculation, both must be examined empirically to successfully leverage post-error learning.

The Elaborative Hypothesis

The elaborative theory posits that unsuccessful retrieval attempts allow for a richer encoding of the correct answer. Retrieval attempts activate various semantically related candidates, one of which is the correct answer, thus setting up a network among the cue and target words (Mera et al., 2021). In the pretesting paradigm, an error could help form a more meaningful relationship between the cue and target. For example, one may generate the word “swims” as a free associate in response to the cue “whale” when the target is “tail.” Instead of simply encoding the pair “whale–tail,” the individual may use the error to create a more robust network between the two words, perhaps thinking of a whale using its tail for swimming. The underlying idea is that prompted retrieval of the target word following the presentation of a cue word evokes several semantically related items. Merging these concepts forms an elaborative memory trace at the time of encoding, which is more likely to be retrieved later at subsequent cue presentations (Huelser & Metcalfe, 2012; Karpicke, 2017). One important finding supporting this theory is that weakly associated word pairs produce stronger learning than strong associates (Carpenter, 2009). Weakly associates prompt participants to generate many related words to recall the target while strongly related pairs are only associated with each other. Elaboration enhances future retrieval because additional semantically related items are encoded alongside the cue and target words.

The Mediator Hypothesis

The mediator hypothesis proposes errors act as secondary cues to retrieve correct answers. In a paired-associate task, generating a non-target word related to the cue could mediate between the cue and target words (Huelser & Metcalfe, 2012; Mera et al., 2021). Instead of solely using a cue during retrieval, one can retrieve the error from the cue

and the target from the error. Referring to the previous example, at subsequent presentations of the word “whale,” one may recall their previous error, “swims,” and from it, the correct target word, “tail.”

This theory finds its strength in an episodic context account of memory retrieval; people encode information about learning events and the episodic and temporal context in which they occur (Howard & Kahana, 2002). This episodic context may be restored during retrieval to facilitate correct recall (Lehman & Malmberg, 2013). In retrieval-based learning, retrieval increases recall because individuals think back to and reinstate their prior learning contexts (Karpicke, 2017).

Models of Post-Error Learning

Both hypotheses point to distinct mechanisms of post-error learning. Examining these mechanisms could establish one as superior to the other. One way to do this is with the use of formal computational frameworks. Here, we used the Adaptive Control of Thought–Rational (ACT-R) cognitive architecture. ACT-R’s mechanisms reflect brain circuits and computations using its functional components and their parameters. ACT-R’s reliable declarative memory module makes it particularly suitable to model various memory paradigms (Ritter, Tehranchi, & Oury, 2019; Taatgen, Lebiere, & Anderson, 2006). ACT-R encodes memories in its declarative memory module as chunks in a semantic network. Each chunk i has a corresponding base-level activation B_i based on the recency and frequency of its presentation as seen in Eq. (1):

$$B_i = \ln\left(\sum_{j=1}^n t_j^{-d}\right) \quad (1)$$

where n is the number of presentations of chunk i ; t_j is the time since the j th presentation of i ; d is the decay parameter reflecting how quickly chunks are forgotten. In addition to this base-level activation, the probability of retrieving i is also a function of spreading activation and noise. Altogether, chunks matching a retrieval request compete for successful retrieval following the formula, seen in Eq. (2)

$$A_i = B_i + \sum_k \sum_j W_{kj} S_{ji} + \varepsilon \quad (2)$$

where the sum of k sums spreading activation across all buffers set to provide it; the sum of j refers to the potential sources of activation that spread to chunks in buffer k ; W_{kj} is the weight or amount of activation spread from source j to chunk i ; S_{ji} is the strength of association from source j to chunk i . Lastly, ε reflects noise to model the noise of a human brain. ACT-R accurately models forgetting and errors, producing results that closely fit human behavioral data on paired associate tasks (Anderson, 1981; Anderson & Reder, 1999; Pavlik & Anderson, 2005). Using this

theoretical foundation, we can create models that implement both hypotheses of post-error learning and compare their predictions and their relative fit to the empirical data.

Spreading activation in ACT-R can be used to properly model elaborative encoding via errors in a paired-associate task. ACT-R’s declarative memory encodes relationships between chunks by linking words in a lexical semantic network. When an error is committed, and feedback is provided, chunks linking the cue and target words could merge with chunks containing the cue and error words to form one elaborative chunk. In the previous paired-associate example, *whale-tail* would be merged with *whale-swims* to create a chunk: *whale-tail-whale-swims*. This chunk could represent the previously discussed meaningful links between cue and target words (i.e., the whale swims with its tail) or simply *whale-tail*, not *whale-swims*. Subsequent presentations of the cue spread more activation to this elaborative chunk; multiple references of the cue word within the chunk increase their strength of association (Figure 1). Overall, this elaborative encoding of the error alongside the cue and target increases its activation.

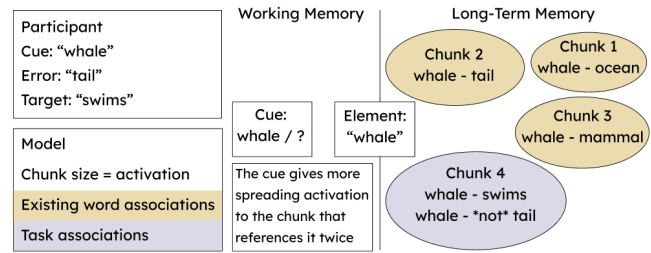


Figure 1: Using spreading activation to model elaborative error learning.

In addition to the declarative module, ACT-R’s procedural module articulates cognitive steps (Anderson et al., 2004). The mediator hypothesis relies on remembering the error itself, suggesting that a cognitive process occurs when remembering and recalling an error. Thus, a production rule that checks for an error can model a mediator explanation of post-error learning (Figure 2). If a previous error is detected, another production fires to retrieve the error as a second cue.

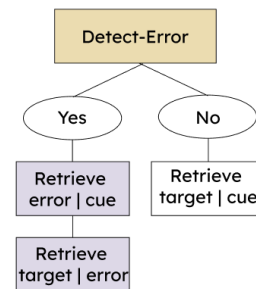


Figure 2: Using an additional production rule to model mediator error learning.

Theoretical Predictions Derived from the Models

It is possible to derive ordinal predictions from these models. Both models predict that error items would be associated with greater response accuracy. In the case of the elaborative model, this is due to the additional spreading activation and, in the case of the mediator model, to the existence of two retrieval routes.

The two models, however, make *opposite* predictions about the relative response times associated with study and error items, respectively. In ACT-R, response times depend on a number of factors, including non-retrieval times spent on perceptual and motor processes, indicated as T_{ER} , and the retrieval time associated with an item i , indicated as $R(i)$. Thus, in general, the response time for the i -th item is:

$$RT = T_{ER} + R(i) \quad (3)$$

In turn, $R(i)$ depends on the activation $A(i)$ of item i , which is the sum of its base-level and spreading activation. Specifically, retrieval times are related to activation by the equation (Anderson et al., 2004):

$$R(i) = \lambda e^{-A(i)} = \lambda / e^{A(i)} \quad (4)$$

Where λ is another individual-specific parameter that scales the retrieval latency.

Both models assume that all study items have been encoded in the same session and practiced the same number of times, so they have comparable activations. The two models make different predictions for the times to retrieve an error item e .

In the *elaborative* model, the additional information encoded in the error item provides additional spreading activation, which sums up the global activation of the error item $A(e)$. We will indicate this additional activation as $S(e)$ so that $A(e) = A(i) + S(e)$. So, the retrieval time for an error item $R(e)$ is:

$$\begin{aligned} R(e) &= \lambda e^{-[A(i)+S(e)]} \\ &= \lambda / e^{[A(i)+S(e)]} \\ &= [\lambda / e^{A(i)}] / e^{S(e)} \\ &= R(i) / e^{S(e)} \end{aligned} \quad (5)$$

Note that, because $S(e) > 0$, $e^{S(e)} > 1$, and thus $R(e) < R(i)$.

According to the *mediator* hypothesis, error items do not differ in terms of activation but in terms of retrieval attempts. That is, on a fraction f of trials involving error items, participants would first retrieve an incorrect target, then detect the error, and finally retrieve the correct item. Both the correct and incorrect items would have comparable activation levels and thus take approximately the same retrieval time as a study item, $R(i)$. Thus, if we indicate the fraction of trials f in which an error is retrieved, we obtain

$$R(e) = (1-f)R(i) + f[R(i) + R(i)] = (1+f)R(i) \quad (6)$$

Because $0 < f < 1$, response time will be *longer* for error items, with the specific amount depending on f .

Thus, although both models leave much room for individual differences across participants (due to differences in the T_{ER} , $S(e)$, λ , and f parameters), the models make clearly opposite predictions about the relative time to respond to study and error items.

Experimental Predictions

Based on the previous theoretical analysis, we can make the following predictions. Firstly, we expect to confirm the results of previous pretesting research (Huelser & Metcalfe, 2012; Kornell et al., 2009). That is, participants should perform better on error generation items compared to study items on the final test of our first experiment.

Additionally, we expect to find a difference in response times on the final test between conditions. However, we are unsure about the directionality of this difference. Longer reaction times in the error condition suggest the majority of participants are learning from errors in a mediator method. Shorter reaction times in the error condition suggest the majority of participants are learning from errors in an elaborative method. Finally, by fitting each model to each participant trial-by-trial using maximum log-likelihood, we expect to gain additional insight into error learning mechanisms. Specifically, we may find out whether certain participants are better fit by mediation and others, elaboration (i.e., mediator learners vs elaborative learners).

Materials and Methods

Participants

A total of 61 University of Washington undergraduate students were recruited for the pretesting task and provided with course credit for their participation.

Pretesting Task

To replicate Huelser and Metcalfe (2012), 60 weakly related word pairs were selected from Nelson, McEvoy, and Schreiber's (1998) norms. This experiment had three phases: learning, distractor, and final test. In the learning phase, the task randomly interleaved study trials and test trials. In study trials, the cue word (e.g., "whale") and its corresponding target (e.g., "tail") were presented simultaneously on the screen for 10 seconds. On test trials, only the cue word was presented on the screen (e.g., "whale"). Participants were asked to respond by typing what they thought the target word was in a textbox (e.g., "swims"). They were given 5 seconds to respond before they were shown the cue word and correct target word simultaneously for 5 seconds. After learning all pairs once, participants played a visuospatial game for 5 minutes as a distractor to prevent rehearsal. Finally, participants took a self-paced final test containing all 60 word pairs.

Results

Replicating the Pretesting Effect

On average, participants had higher recall accuracy on error items ($M = 0.70 \pm 0.16$) than study items ($M = 0.60 \pm 0.18$), as seen in Figure 3. Mixed linear models were used to account for variability and individual differences. Specifically, we fitted a mixed model to all of the experimental trials, including the particular trial condition (Study vs. Error) as a fixed effect and the participant-level intercept as a random effect; the latter accounts for individual differences in response accuracies. Because accuracy is a binary variable, the model used a binomial distribution to capture the predicted variable. The model uncovered a large main effect of condition ($\beta = -0.47$, $SE = 0.08$, $t = -6.13$, $p < 0.0001$). The complete results of the model are shown in Table 1. These findings confirm the results of previous studies (Huelser & Metcalfe, 2012; Kornell et al., 2009).

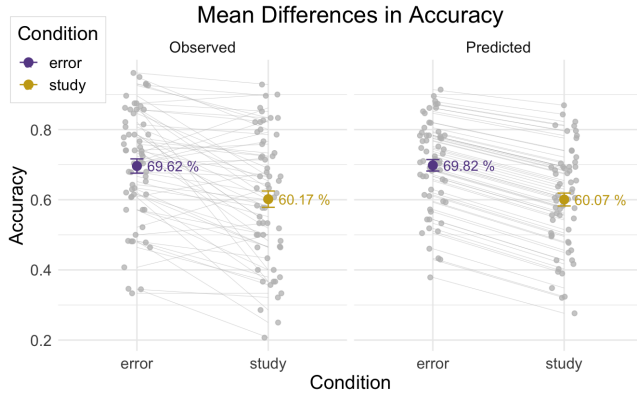


Figure 3: Differences in average final cued-recall accuracy split by condition. Gray dots and lines represent data for individual participants; colored dots and error bars represent means \pm SE for the Error (purple) and Study (yellow) conditions

Table 1: Results of the Mixed-Level Model for Accuracy

Statistical Test	β estimate	SE	t	p
(Intercept)	0.925***	0.107	8.651	5e-18
Condition	- 0.468***	0.076	- 6.130	8e-10

* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

Reaction Times

To remove extreme values from our data, we used a maximum cutoff point of 15,000ms and a minimum cutoff point of 200ms. Only correct trials were included.

On average, participants had longer response times on error items ($M = 4,104 \pm 779$ ms) than study items ($M = 3,920 \pm 936$ ms), as seen in Figure 4. The difference between condition response times was compared with a

mixed linear model. As in the previous case, the model includes each trial condition (Study vs. Error) as a fixed effect and the participant-level intercept as a random effect; the latter accounts for individual differences in response latencies. Unlike the previous case, the model used a Gaussian distribution to model the dependent variable. Additional random factors, such as random slopes to account for different effects for each participant, did not improve the fit of the model. The model confirmed a large and significant main effect of condition ($\beta = -261.40$, $SE = 88.79$, $t = -2.944$, $p < 0.005$). The complete results of the model are shown in Table 2.

To examine the possibility that different individuals might use different strategies, a second mixed linear model was created, which included the participant-level slope as a random effect. This model allows for different individuals to have either shorter or longer RTs in the error conditions, thus allowing the possibility that some individuals might use an elaborative strategy. This second model replicated the results of the first, finding a significant main effect of the condition ($\beta = -255.99$, $SE = 89.18$, $t = -2.87$, $p < 0.005$). An ANOVA test confirmed that the second model does not provide a greater fit than the first ($\chi(3) = 0.38$, $p > 0.94$); furthermore, all the fitted slopes in the ensuing model were negative, suggesting that the apparent differences in slopes in Figure 4 are due to outlier responses, rather than systematic use of the elaborative model.

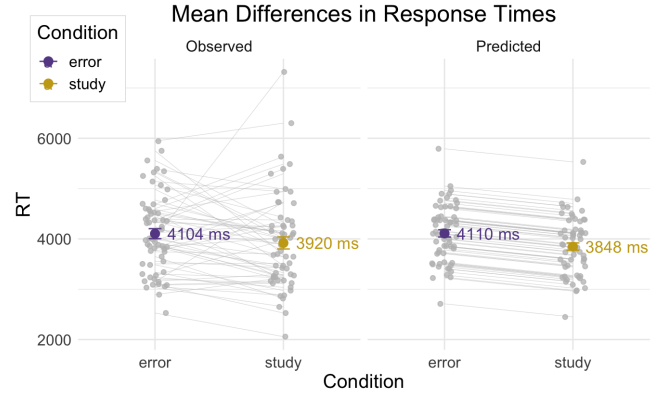


Figure 4: Differences in final test response times by condition. Gray dots and lines represent data for individual participants; colored dots and error bars represent means \pm SE for the Error (purple) and Study (yellow) conditions

Table 2: Results of the Mixed-Level Model for Response Times.

Statistical Test	β estimate	SE	t	p
(Intercept)	4109.70***	103.06	39.877	2e-16
Condition	-261.40**	88.79	- 2.944	0.003

* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

Individual Model Fitting and Comparison

Although the experimental results are strongly in favor of the mediator hypothesis, considerable variability exists in the data, as shown in the averages of Figures 3 and 4. Despite allowing for the fit of individual data, MLM models are by nature poorly fit to capture the nonlinear dynamics of memory. We are also interested in seeing precisely how well each model fits each individual to examine individual differences comprehensively. For this reason, we implemented the two hypotheses by running ACT-R models and used convex optimization techniques to maximize their fit to every individual. The fit of the two models was compared by estimating their likelihoods given the data. The likelihood of a model M given a vector of data \mathbf{x} , $L(M|\mathbf{x})$, is the probability of observing the data, given M : $L(M|\mathbf{x}) = P(\mathbf{x}|M)$. Our data consists of multiple independent responses x_1, x_2, \dots, x_N , thus

$$L(M|\mathbf{x}) = P(x_1|M) \cdot P(x_2|M) \cdot \dots \cdot P(x_N|M) = \prod_i P(x_i|M)$$

Because the product of probabilities becomes vanishingly small, it is common to use *log*-likelihood:

$$\log L(M|\mathbf{x}) = \log \prod_i P(x_i|M) = \sum_i \log P(x_i|M)$$

In our case, each model was jointly fitted to two behavioral measures: The responses x and its corresponding response time RT .

Responses Probabilities The retrieval probability P_R of a specific response x can be computed using the Boltzmann equation:

$$P(x) = e^{A(x)/s} / \sum_i e^{A(i)/s} \quad (7)$$

Where s is a noise or temperature parameter, and the sum in the denominator refers to all of the *competing* memories for a given cue, including errors and potentially associated concepts. To get a realistic representation of the pool of competing memories, all of the responses produced by all participants for a given word were included as potential distractors in the model and given a trace at time 0.01 resulting in a minimal activation value.

In the case of the elaborative hypothesis, Equation 7 can be directly applied to the given response x . In the case of the mediator hypothesis, a correct response can be made either by directly retrieving the correct response or by first retrieving the error-generated response. Thus, the probability of observing a correct response is given by $P(x=correct) + P(x=error)$.

Response Time Probabilities The probability of observing a given response time for the response x can be computed from Equation 4 if the distribution of the noise term s is known. In ACT-R, declarative memory noise follows a logistic distribution with a mean of zero and a standard deviation of $\pi*s/\sqrt{3}$ (Anderson et al., 2004). Applying Equation 4 to the probability density function of the noise

gives the probability density function of different response times, as shown in Figure 5:

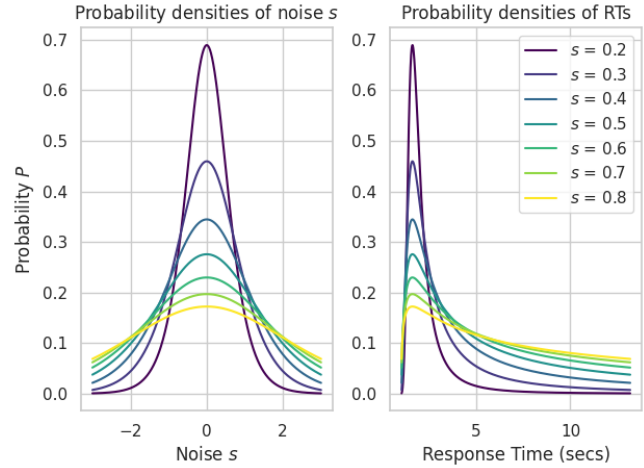


Figure 5: Different response time probability density functions for different levels of noise s .

Model Fitting Each model was fit to all the choices of every participant by identifying parameter values that maximize the log-likelihood function. Because no closed-form solutions exist to derive the maximum likelihood of ACT-R equations, the best-fitting parameter values were identified using Powell's (1964) optimization algorithm as implemented in Python's SciPy package (Virtanen et al., 2020). This method was chosen because it does not require explicit derivatives and allows us to specify meaningful bounds for parameter values.

Using Powell's method, three parameters were fit for each individual: the decay rate d (which is known to vary significantly across people: Sense et al., 2016), the noise s , and the non-decision time T_{ER} . Parameters W (Eq. 3) and λ (Eq. 4) were kept constant to a default value of $W = \lambda = 1$. This was done because of the parameter identifiability problem: W is difficult to separate from d in Equation 2, and λ has similar effects to T_{ER} in Equation 4.

Results To perform the trial-by-trial analysis, the models were implemented directly in Python, rather than using the ACT-R architecture. In a way that mimicked the behavioral task, for each participant on each trial (i.e., cue presentation), the model calculated the probability of retrieving the participant's actual response given word activations within the model's declarative module as well as the probability of the participant's response time given word activations and the number of retrievals. In this way, the elaborative model assigned higher probabilities to shorter response times on error items due to spreading activation, while the mediator model assigned higher probabilities to longer response times on error items due to multiple retrievals. After fitting the values of the d , T_{ER} , and s parameters, and computing the maximum log-likelihoods for each participant, we found the greatest majority of

participants were best fit by the mediator model ($N = 55$), with only a few best fit by the elaborative model ($N = 6$), as seen in Figure 6.

The group-level difference in the two models can be computed by aggregating their participant-level differences (Yang & Stocco, 2021). In this case, the cumulative difference in log-likelihood between the two models is 1,728. Because of the definition of log-likelihood, this difference indicates an odds ratio of $e^{1,728}$, indicating that the mediator model is $e^{1,728}$ more likely to fit the data than the elaborative model.

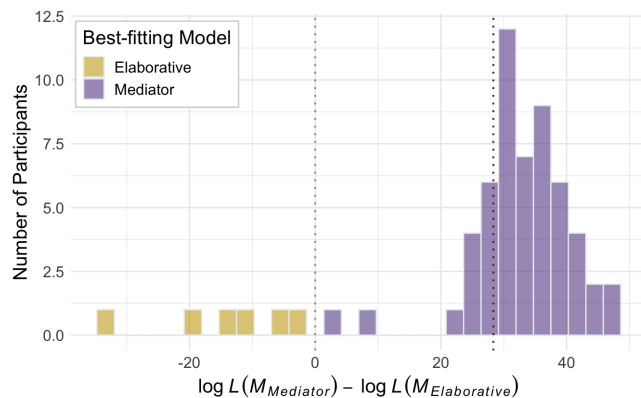


Figure 6: Histogram log-likelihood differences between the mediator and elaborative models across all participants: yellow indicates participants best fit by the elaborative, and purple by the mediator, models. The green dotted line indicates the point at which the difference is zero; the red line indicates the mean difference (27.9) across all models.

Discussion

In this paper, we examined if errors facilitate learning, including an investigation to probe potential underlying mechanisms. While the pretesting paradigm's success in improving learning compared to just studying is well documented (Huelser & Metcalfe, 2012; Kornell et al., 2009), research into the underlying cognitive processes facilitating this phenomenon remains speculative. Without understanding how post-error learning works, applications remain restrictive. Revealing underlying mechanisms may allow us to increase the efficacy of current learning and memory models by harnessing the power of errors. Our study aims to provide a baseline to research post-error learning mechanisms using cognitive models.

Importantly, we were able to replicate the pretesting results of existing literature; final cued-recall accuracy was higher in the error-generation condition than in the study-only condition. This helps to confirm the benefit of retrieval attempts before study opportunities and advocates for further research into the mechanisms of this process.

Different reaction time hypotheses arose from our two post-error learning models based on the ACT-R architecture. First, an elaborative model predicts that error learning results in quicker response times on subsequent tests. This is

because elaboration works through spreading activation, adding activation to the correct answer which speeds up retrieval and response times. Alternatively, a mediator model predicts that error learning results in slower response times on subsequent tests. Mediation uses an extra step to retrieve an error as a secondary cue to get the correct answer. Although this procedure increases accuracy, it also costs extra time, resulting in longer response times. Results from the current study demonstrate that average response times are longer on error items than on study items. This supports the mediator hypothesis of post-error learning. Additional investigation into trial-by-trial fits of models revealed overwhelming support for the mediator model.

The most notable limitation of this study is the generalizability of the pretesting paradigm. Errors as we think of them are often made after a study opportunity. Looking at encoding errors in learning would extend this research to more real-world errors. While the mediator and elaborative hypotheses are some of the most notable explanations of post-error learning, other explanations exist which may also account for reaction time differences. Analyzing these errors in a different framework may allow us to investigate more theories which is critical for the advancement of learning optimization. Additionally, this paradigm does not look at memory over longer periods of time. Deeper encoding processes resulting from error commission may lead to facts that are more resistant to forgetting over days, weeks, and months. An alternative paradigm is the adaptive fact-learning system developed by Sense and van Rijn (2022; Sense et al., 2016), in which new paired associations are presented at a pace individualized to each participant to optimize retention. Importantly, their paradigm internally makes use of ACT-R to model each individual's memory, and yields highly reliable estimates of each individual and each item's decay rate (Sense et al., 2016). A modification of this paradigm that includes an error-generating phase provides important information as to whether, for example, error items are forgotten at lower speeds, rather than (or in addition to) having additional retrieval routes.

Models are unique in their ability to reconceptualize behavioral results. By decoding human behavior, models begin to reveal cognition by stabilizing the messiness of data. As such, the proposed cognitive models in this paper can help identify mechanisms of post-error learning. These models could distinguish different learners from one another and propose ways to manipulate post-error learning by targeting the relevant cognitive processes. Moreover, these findings could extend to fields outside of cognitive psychology, advocating for the benefit of making mistakes in various educational settings and assisting in developing AI and machine learning advancements that update comprehensive feedback histories with each new learning experience.

References

- Anderson, J. R. (1981). Interference: The relationship between response latency and response accuracy. *Journal of Experimental Psychology: Human Learning and Memory*, 7(5), 326–343.
- Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological review*, 111(4), 1036.
- Anderson, J. R., & Reder, L. M. (1999). The fan effect: New results and new theories. *Journal of Experimental Psychology: General*, 128(2), 186–197.
- Butterfield, B., & Metcalfe, J. (2001). Errors committed with high confidence are hypercorrected. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 27(6), 1491–1494.
- Carpenter, S. K. (2009). Cue strength as a moderator of the testing effect: The benefits of elaborative retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(6), 1563–1569.
- Ceraso, J. (1967). The interference theory of forgetting. *Scientific American*, 217(4), 117–127.
- Cyr, A., & Anderson, N. D. (2015). Mistakes as stepping stones: Effects of errors on episodic memory among younger and older adults. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 41(3), 841–850.
- Howard, M. W., & Kahana, M. J. (2002). A distributed representation of temporal context. *Journal of Mathematical Psychology*, 46(3), 269–299.
- Huelser, B. J., & Metcalfe, J. (2012). Making related errors facilitates learning, but learners do not know it. *Memory & Cognition*, 40(4), 514–527.
- Kapur, M., & Bielaczyc, K. (2012). Designing for productive failure. *Journal of the Learning Sciences*, 21(1), 45–83.
- Karpicke, J. D. (2017). Retrieval-based learning: A decade of progress. In *Learning and Memory: A Comprehensive Reference* (pp. 487–514). Elsevier.
- Kornell, N. (2014). Attempting to answer a meaningful question enhances subsequent learning even when feedback is delayed. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 40(1), 106–114.
- Kornell, N., Hays, M. J., & Bjork, R. A. (2009). Unsuccessful retrieval attempts enhance subsequent learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(4), 989–998.
- Lehman, M., & Malmberg, K. J. (2013). A buffer model of memory encoding and temporal correlations in retrieval. *Psychological Review*, 120(1), 155–189.
- Mera, Y., Rodríguez, G., & Marin-Garcia, E. (2021). Unraveling the benefits of experiencing errors during learning: Definition, modulating factors, and explanatory theories. *Psychonomic bulletin & review*, 29(3), 753–765.
- Nelson, D., McEvoy, C., & Schreiber, T. (1998). *University of South Florida Free Association Norms*.
- Pavlik Jr, P. I., & Anderson, J. R. (2005). Practice and forgetting effects on vocabulary memory: An activation based model of the spacing effect. *Cognitive science*, 29(4), 559–586.
- Powell, M. J. D. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, 7 (2): 155–162
- Ritter, F. E., Tehranchi, F., & Oury, J. D. (2019). ACT-R: A cognitive architecture for modeling cognition. *WIREs Cognitive Science*, 10(3), e1488.
- Sense, F., Behrens, F., Meijer, R. R., & van Rijn, H. (2016). An individual's rate of forgetting is stable over time but differs across materials. *Topics in Cognitive Science*, 8(1), 305–321.
- Sense, F., & van Rijn, H. (2022, January 27). Optimizing Fact-Learning with a Response-Latency-Based Adaptive System.
- Taatgen, N., Lebiere, C., & Anderson, J. (2006). Modeling paradigms in ACT-R. In *Cognition and multi-agent interaction: From cognitive modeling to social simulation* (pp. 29–52).
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & Van Mulbregt, P. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods*, 17(3), 261–272.
- Yang, Y. C., Karmol, A. M., & Stocco, A. (2021). Core Cognitive Mechanisms Underlying Syntactic Priming: A Comparison of Three Alternative Models. *Frontiers in Psychology*, 12, 662345.

Cognitive Modeling of Category Learning and Reversal Learning

Marcel Lommerzheim (marcel.lommerzheim@lin-magdeburg.de)

Combinatorial NeuroImaging, Leibniz Institut für Neurobiologie, Brenneckestraße 6
39118 Magdeburg, Germany

André Brechmann (brechmann@lin-magdeburg.de)

Combinatorial NeuroImaging, Leibniz Institut für Neurobiologie, Brenneckestraße 6
39118 Magdeburg, Germany

Nele Russwinkel (russwinkel@uni-luebeck.de)

Human-Aware AI, IFIS, Universität zu Lübeck, Ratzeburger Allee 160
23562 Lübeck, Germany

Abstract

During learning humans often test new hypotheses to infer causal relations between objects and actions. One very common example of learning is category learning in which humans learn to differentiate between different stimuli based on their features. The rational aspects of category learning in form of hypotheses testing need to be taken into consideration for improving computational models. Compared to reinforcement learning models that assume gradual learning, cognitive modeling allows to implement hypotheses testing and thus enabling steep transitions in learning. Here we extend our previously developed ACT-R model in a systematic way to further improve its fit to an auditory category learning and reversal learning experiment. For the initial category learning phase we optimized the model by enabling it to use two stimulus features right from the start. For improving the model's performance in the reversal phase, we introduced an additional mechanism of switching the motor-response for a given categorization. With these two changes we significantly increased the model's performance in our task. By comparing the backward learning curves of the participants to those of our model we observed that our model exhibits steep transitions during the initial category learning phase, a feature that reinforcement learning models have difficulties to reproduce.

Keywords: ACT-R; Category Learning; Reversal Learning; Cognitive Modeling

Introduction

When facing new complex problems, we often approach a solution by trial and error. During the course of learning, causal relations between objects and actions become evident which may result in hypotheses and strategies that are tested in subsequent trials. A prime cognitive function that has been the focus of several computational models for understanding human learning is categorization. How human category learning works and which modeling method is best suited to understand the involved processes is still an open question. Most current modeling approaches are based on reinforcement learning algorithms that make the assumption of a gradual learning process. Since human category learning often shows steep transitions in performance, such models often cannot fully explain learning data. Thus, modeling accounts are required that can capture sudden transitions in performance (Smith & Ell, 2015). An alternative approach to reinforcement based models for explaining human category learning is ACT-R. A vast number of cognitive tasks have been implemented with ACT-R that predict human behavior.

Especially task models using memory and learning mechanisms have shown good correspondence to human behavior (e.g. (Nijboer, Borst, van Rijn, & Taatgen, 2016; Morita et al., 2020)).

In previous work, we have implemented a basic ACT-R model to explain the average learning data of humans in a multidimensional auditory category learning and reversal learning task (Prezenski, Brechmann, Wolff, & Russwinkel, 2017), and improved its performance by adapting the salience level of stimulus dimensions that are selected for initial decisions of the human learners (Lommerzheim, Prezenski, Russwinkel, & Brechmann, 2020). However, we still found discrepancies between the overall performance of the model and that of human learners, especially during the initial learning phase.

The aim of the present study was therefore to improve the existing model in order to minimize the difference to the experimental data and to test it on a new set of learning data. The initial focus was on improving the initial speed of category acquisition that was still behind to that of the human participants. We then proceeded by investigating this model's performance during two reversals of the assignment of the target button and found that the model needed further refinements to capture the participants' performance. Moreover, we investigated whether the model is able to reproduce steep transitions that occur during the initial category learning as evident in the backward learning curves (BLC) of the participant's performance which are not easily explained in reinforcement models (Jarvers et al., 2016).

Methods

Experiments

Participants 55 subjects participated in experiment I (27 female, 28 male, age range between 21 and 30 years, all right handed, with normal hearing). 22 subjects participated in experiment II (11 female, 11 male, age range between 18 and 34 years, all right handed, with normal hearing). Both experiments took place inside a 3 Tesla MR scanner. All subjects gave written informed consent to the studies, which was approved by the ethics committee of the University of Magdeburg, Germany.

Stimuli A set of 160 different frequency-modulated tones served as stimuli for the categorization task. The sounds had five different features, with one of two possible categorical values: duration (short, 400 ms, vs. long, 800 ms), direction of frequency modulation (rising vs. falling), volume (low, 76-81 dB, vs. high, 86-91 dB), frequency range (five low frequencies, 500-831 Hz, vs. five high frequencies, 1630-2639 Hz), and speed of modulation (slow, 0.25 octaves/s, vs. fast, 0.5 octaves/s). The task relevant features were the direction of frequency modulation and sound duration, resulting in four sound categories: short/rising, short/falling, long/rising, and long/falling. For each participant, one of these categories constituted the target sounds (25%), while the other three categories served as non-targets (75%). As feedback stimuli, we used naturally spoken utterances (e.g., ja, “yes”; nein, “no”) as well as one time-out utterance (zu spät, “too late”) taken from an evaluated prosodic corpus (Wolff & Brechmann, 2012, 2015).

Task The experiments lasted about 33 minutes in which a large variety of sounds were presented in 240 trials in pseudo-randomized order and with a jittered inter-trial interval of six, eight, or ten seconds. The participants were instructed to indicate via button-press whether they considered the sound in each trial to be a target (right index finger) or a non-target (right middle finger). They were not informed about the target category but had to learn by trial and error. Correct responses were followed by positive feedback, incorrect responses by negative feedback. If participants failed to respond within two seconds following the onset of the sound, the time-out feedback was presented.

In experiment I a break of 20 seconds was introduced after 120 trials. From the next trial on, the contingencies were reversed such that the target stimulus required a push of the right instead of the left button. The participants were informed in advance about a resting period after finishing the first half of the experiment but they were not told about the contingency reversal. In experiment II two such reversals happened, the first after 80 trials and the second switch back to the initial assignment of the target button after 160 trials.

Model

The experimental task was modeled with the ACT-R framework, a cognitive architecture that provides a set of different cognitive functionalities called modules that interact with each other coordinated by a central production system. We will first describe the general modeling approach as originally implemented by Prezenski et al. (2017) and then outline the changes that we made to improve the model fit of the data of experiment I further in order to test this model on the experimental data of experiment II.

General Approach Our model uses the motor, the declarative, the imaginal, the goal, the aural, and the procedural module of ACT-R. The motor module is responsible for the motor output, i.e. button press. The declarative module repre-

sents the long-term memory of ACT-R in which all representation units (chunks) are stored and retrieved. The imaginal module acts as the working memory that holds and modifies the current problem state. The goal module represents the control states. The aural module is responsible for processing auditory information. The procedural module is central for ACT-R as it coordinates the other processing units by selecting production rules (representing procedural knowledge) based on the current state of the modules.

In order to specify a model in ACT-R, the production rules and the chunks (representing background knowledge) need to be defined. Chunks are the smallest units of information and can be exchanged between buffers. Production rules or productions have a condition and an action part. They are selected sequentially. Thus, only one production can be selected simultaneously. They are only selected if the condition part of the production matches the state of the modules. Subsequently, the action part modifies the chunks in the modules. In the case that more than one production matches the state of the modules, a subsymbolic production selection process chooses which production is selected. Another subsymbolic process of ACT-R is the activation of a chunk. The activation of a chunk determines whether a chunk can be retrieved from memory and how long this takes. A chunk’s activation value is determined by its past usefulness (base-level activation), its relevance in the current context (associative activation) and a noise parameter.

Table 1: Strategy and control chunks.

Strategy chunks	
Slot	Value
complexity	(one or two)
1.feature-value-pair	(e.g. duration short)
2.feature-value-pair	(nil, e.g. volume high)
response	(0 or 1)
unsuccessful	(nil-yes)
first attempt	(nil-yes)
1.count	(nil, 1,2...threshold)
2.count	(nil, 1,2...threshold)
Control chunks	
Slot	Value
complexity	(one or two)
uncertain	(nil - yes)
environment	(nil-change)

The model is equipped with two different types of chunks depicted in Table 1: strategy chunks and control chunks. Strategy chunks represent the strategies in form of examples of feature-value pairs (i.e. duration is long, volume is loud) and responses (i.e. left or right button). These strategy chunks are stored in and retrieved from long-term memory (declarative module). The currently pursued strategy is stored in

working memory (imaginal module). A strategy chunk contains the following information: Which feature(s) (i.e. volume) and what corresponding categorical value(s) (i.e. loud) are relevant, the proposed response, and the degree of complexity. The degree of complexity determines whether the model attends to just one feature (one-feature strategy) or to two features (two-feature strategy). This approach of storing the environmental state, the action and the response within one chunk resembles that of Instance-based learning originally proposed by Gonzalez, Lerch, and Lebiere (2003). Furthermore, it contains a meta-cognitive process that includes noting whether and how often a strategy was successful. All of the different possible strategy chunks are stored in declarative memory right from the start and randomly one of them is chosen at the beginning of each model run. Control chunks represent other meta-cognitive aspects of the model. They are stored in the goal buffer of the model. They include first the level of rule complexity used, second whether or not a long-time successful strategy resulted in negative feedback, and third whether external changes occurred that require a new search for a strategy.

The production rules used for defining the task are described now in detail with the specific names in parentheses as shown in Figure 1. When a sound is presented to the model, it enters the aural location buffer (listen). Subsequently it is encoded in the aural buffer (encode). This results in a chunk with all audio information necessary (duration, direction of pitch change, volume, and frequency range) stored in the aural buffer. This audio chunk is then compared to the strategy chunk in the imaginal buffer (compare). If the specific features of the strategy chunk match those of the audio chunk, the response is chosen according to the strategy chunk (react-same), if not, the opposite response is selected (react-different). The model then listens to the feedback and holds it in the aural-location buffer (listen-feedback). Subsequently it is encoded in the aural buffer (encode-feedback). In case of positive feedback, the current strategy is maintained and the count-slot is updated (feedback-correct). In case of negative feedback, the strategy usually is altered depending on previous experiences (feedback-wrong).

The strategy updating is implemented in the following way: in case that a one-feature strategy fails in the first attempt, a different motor response is selected for this feature-value pair. Otherwise, the feature-value pair is changed while the response is retained. When a one-feature strategy was successful often and then fails once, it is not directly exchanged, but re-evaluated and it is noted that the strategy has resulted in negative feedback. A switch from a one-feature strategy to a two-feature strategy can occur in two conditions: Either no successful one-feature strategy is left or an often successful one-feature strategy fails repeatedly. For switches of two-feature strategies the following rules apply: If the first attempt of a two-feature strategy fails, any other two-feature strategy is used. In the case a two-feature strategy that was initially successful fails, a new strategy that retains one of the

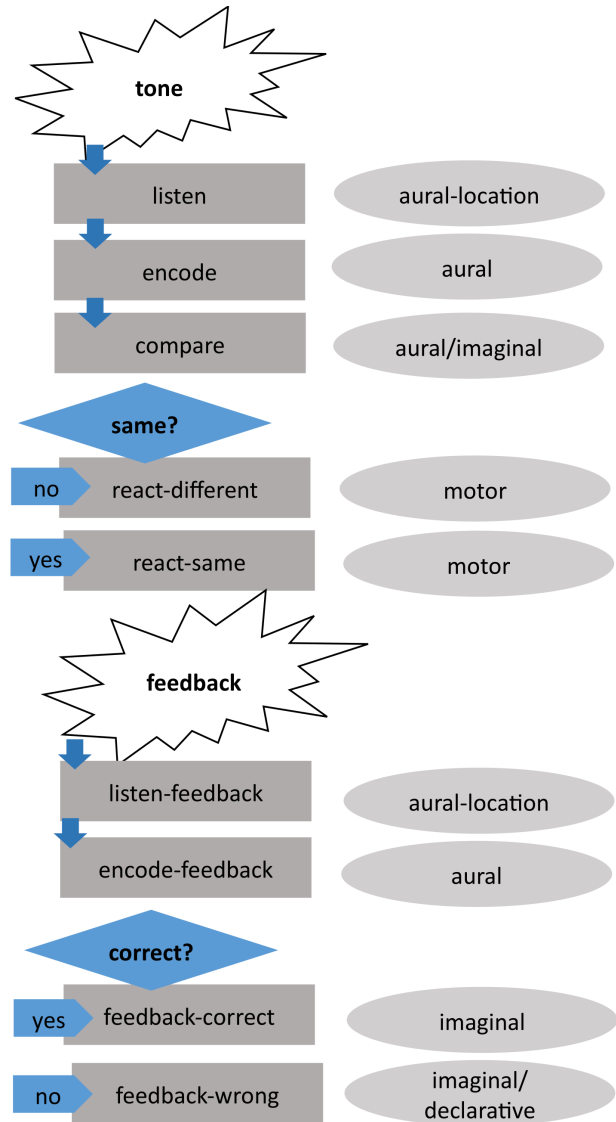


Figure 1: The productions used by the model to run through a trial (taken from Prezenski et al. (2017))

feature-value pairs is selected and the response is transferred to the imaginal buffer. When an environmental change is detected, an often successful two-feature strategy will fail and a retrieval of another two-feature strategy takes place.

In order to improve the models performance and to better fit the human experimental data, in a subsequent work we added preferences to relevant strategies (Lommerzheim et al., 2020). While the probability of selecting a strategy in the first model was equal for all one-feature strategies, we changed the model by increasing the probability (by changing the utility of the production) of being selected for different strategy chunks. We did so by setting the activation values of strategies that use duration and/or direction of frequency modulation as features to 1.0.

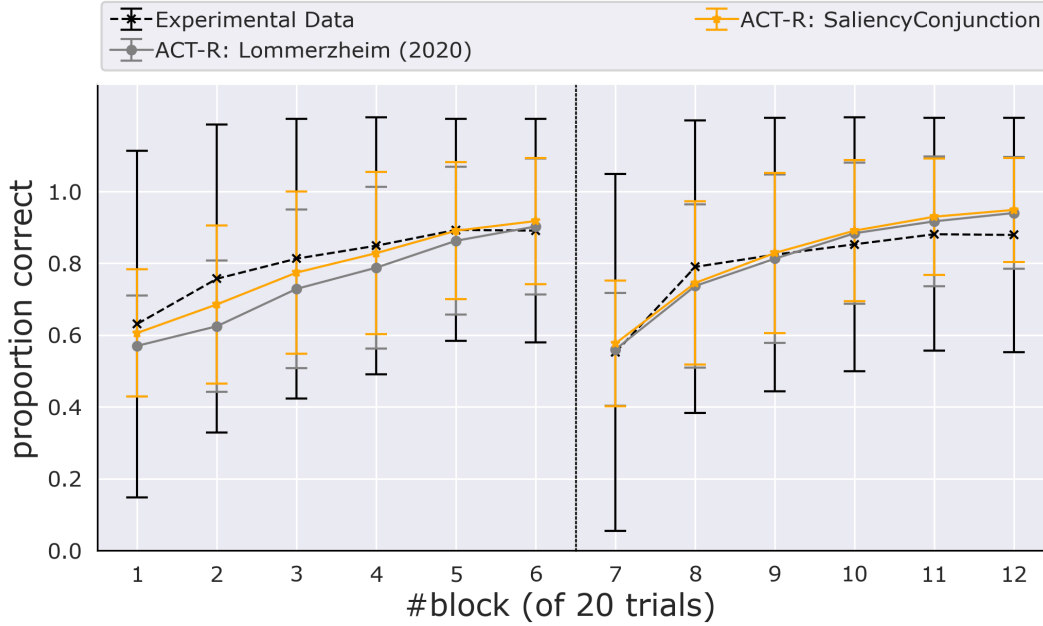


Figure 2: Mean learning curves of the participants of experiment I, the previously published modeling results and the results of the adapted model that allows to use two features right from the start. Error bars depict standard deviance of the mean.

Improving Initial Category Learning As there was still a significant difference between the mean human performance and the model especially in the first half of experiment I, we considered different ways of further improving the model’s performance. Ultimately, based on hypothetical considerations that some participants might think of a combination of two features as relevant right from the start, we added productions to allow the model to also start with a 2-feature strategy at the beginning of each model run (*saliencyConjunction*).

Learning to Switch Back As we found that this model is not able to profit from the previously found categorization when switching back in the third phase of experiment II, we added productions that allow the model to just switch the response of the current strategy when it encounters situations in which a previously successful strategy unexpectedly failed (*saliencyConjunctionSwitch*).

Backward Learning Curves To obtain backward learning curves, we used an approach similar to that of Smith and Ell (2015): we determined the first block of 12 trials in which each participant reached a learning criterion of three correct responses to each of the different categories in a row. Subsequently, the individual learning curves were aligned to this block 0 and then averaged across participants. Thus, block 0 shows high performance by definition. However, comparing the performance of the previous block -1 to the following block +1 allows to identify the true nature of the underlying transition: if the performance in block -1 is near the performance of the learning criterion there was only a gradual improve in performance. Models based on gradient descent can

perfectly well capture such transitions. On the other hand, a performance in block -1 that is still far away from the learning criterion suggests a form of sudden learning that cannot be explained by models based on gradual changes but instead need different types of categorization models as pointed out by Smith and Ell (2015).

Results

Initial Category Learning

Initially we aimed for a better fit of the initial category learning. Accordingly, Figure 2 shows the results obtained for experiment I. Here, the previously published model still showed a rather large difference especially for the first 4 blocks of the experiment. By allowing the model to also start with a 2-feature strategy the learning speed at the beginning of the experiment could be improved nearly matching that of the participants while the ability to relearn after the reversal remained unaffected.

Reversal Learning

We tested the models ability to learn reversals further by comparing its performance to the participants experiment II as shown in Figure 3. Especially in block 9 right after the second reversal the participants are relearning faster than the best fitting model of experiment I. This difference required us to adapt our model further. By adding additional productions for just switching the response after receiving unexpected negative feedback we could drastically increase the performance of the model after the second reversal, suggesting that this modification captures the behavior of some participants well.

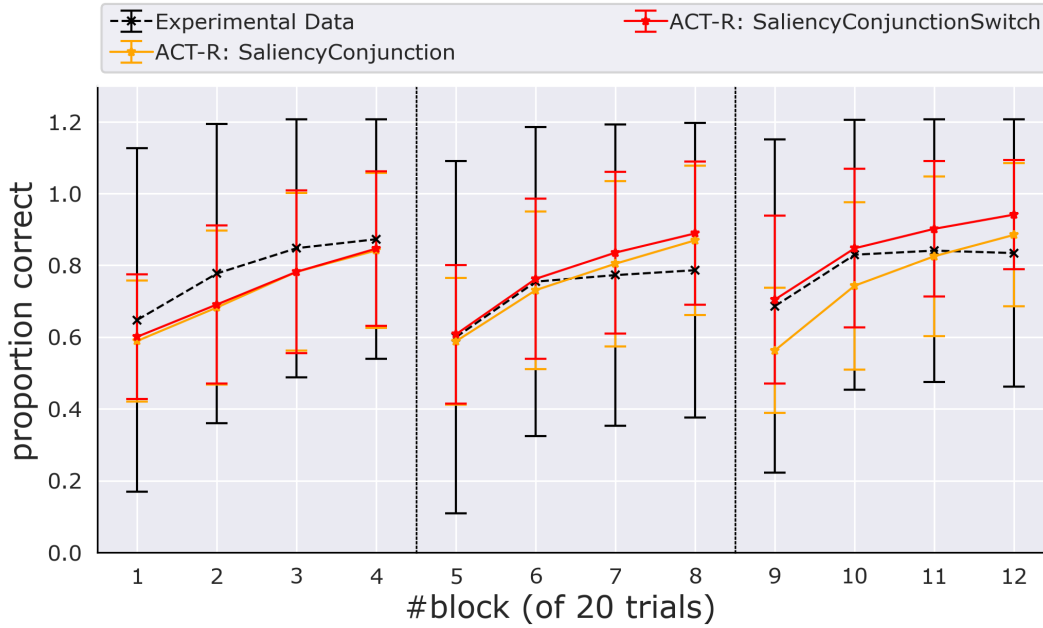


Figure 3: Mean learning curves of the participants of experiment II, the best fitting model for experiment I and the model enhanced with productions for just switching the response. Error bars depict standard deviation of the mean.

Sudden Learning

To compare the learning behavior of participants and our model in more detail, we calculated their BLCs during the initial category learning. Figure 4 clearly shows that in the blocks before reaching the criterion the participants performance is around 0.75 and suddenly increases to a performance of around 1 indicating a discontinuous learning process. The ACT-R model shows a similar sudden increase in performance with a transition from around 0.5 to 1 from block -1 to 1. Even though there is a quantitative difference between the participants and our model, the main finding is the same steep increase in performance when reaching the learning threshold which is something that reinforcement learning models based on gradient descent are not able to reproduce.

Discussion

The purpose of this study was to improve our existing model of categorization and reversal learning and to adapt it for the additional findings in our experiment with a repeated reversal. The results show that the difference to our experimental data can be further reduced by allowing the model to pay attention to two stimulus features right from the start. We did not include this possibility in the initial model because studies on rule-based category learning have claimed that during the initial phase of learning, humans typically apply one-feature strategies (Ashby, Alfonso-Reese, Turken, & Waldron, 1998). However, strategies based on more than one feature can be learned (Ell, Smith, Deng, & Hélie, 2020), but are more difficult (Ishizaki, Morita, & Morita, 2015). Since the additional mechanisms captures the behavior of the human learners well

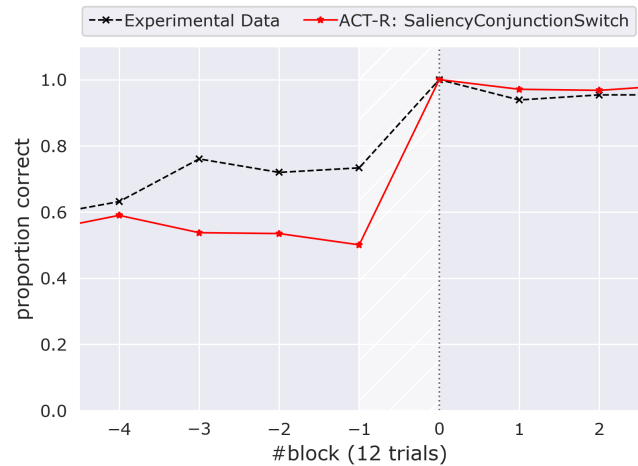


Figure 4: Backward learning curves of the participants and the model in experiment I.

suggests that two-feature strategies might be adopted by the participants early on during initial learning. Testing this optimized model on the new data of experiment II showed that the participants' performance after the second reversal was not well explained. This failure of replicating the significantly better performance after the second compared to the first reversal was also evident in a previous modeling approach with reinforcement based algorithms (Jarvers et al., 2016). In our ACT-R model, we could compensate this limitation by introducing the additional mechanism for switching the motor

response.

By analyzing backward learning curves, in addition to the forward learning curves, we showed that our model qualitatively produces similar learning transitions as the human learners. This is an advantage to reinforcement learning models because of their gradient descent algorithms.

One remaining discrepancy between our model and the experimental data is the performance towards the end of experiment II. The model performs better than our participants which might be due to reduced attention/mental exhaustion of our participants. How such effects of fatigue can be modeled within our ACT-R model will be addressed in future research.

Another limitation of our model might be the assumption that all features of each tone are perceived and compared to the feature-value pair of the current strategy chunk. This could be changed by implementing an attentional mechanism in the way the model listens to the tones. Even though this would initially reduce the performance of the model it might provide room for further improvements: an additional working memory mechanism could be implemented that increases the probability of picking up strategies which were compatible with the obtained feedback.

Another point for future investigations is the high variance in performance of the human learners which the model cannot account for in the present state. While our model solely relies on an explicit learning system that generates and tests hypotheses, some of the participants might take another approach and try to solve the task subsymbolically. Another reason for the high variance seems to be that some participants were satisfied with receiving positive feedback in 3 out of 4 trials by just paying attention to one feature. Our model on the other hand always strives to avoid all negative feedback. In future versions of the model this might be mitigated by introducing an additional parameter that determines the model's aspiration of obtaining correct feedback.

Another line of future research to test the model's predictions would be to compare the activation of the ACT-R model to the fMRI activation in corresponding brain regions. Borst and Anderson (2015) showed how the activation of the different modules can be used to make predictions for the BOLD signals of different brain regions that are thought to represent these modules. By comparing these predictions to our experimental findings the model could be further evaluated and refined. Figure 5 shows the mapping of the declarative module to the dorsolateral prefrontal cortex and the error-related activation we found for the contrast of incorrect vs. correct feedback (Wolff & Brechmann, 2022). Our model also predicts a different activation for those trial types within the declarative module: while correct feedback merely leads to an update of the currently held strategy in the imaginal buffer, incorrect feedback requires the model to switch to another compatible strategy and thus leads to a higher activation within the declarative module.

In summary, we showed how our existing model's capability for category learning and reversal learning could be further

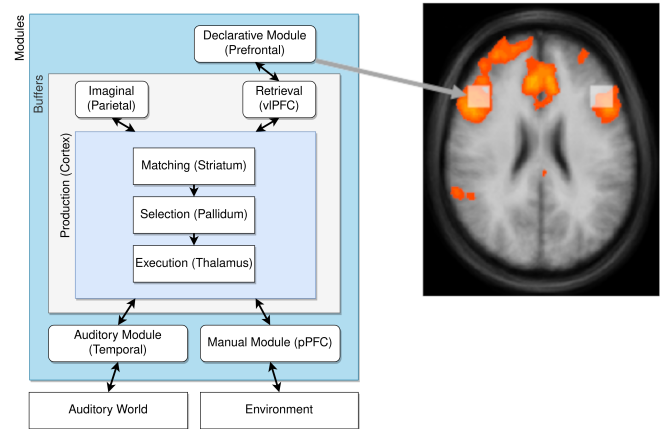


Figure 5: Left part shows the different ACT-R modules (adapted from Borst and Anderson (2015)), right part shows the claimed location of the declarative memory within the brain (grey box) and the contrast of incorrect vs. correct feedback measured during experiment I.

improved and that it is able to replicate essential characteristics of transitions in learning observed in humans performing a rule-based category learning task.

Acknowledgments

This work was supported by the German science Foundation (DFG BR 2267/9-1).

References

- Ashby, F., Alfonso-Reese, L., Turken, A., & Waldron, E. (1998). A neuropsychological theory of multiple systems in category learning. *Psychological Review*, 105(3), 442–481. doi: 10.1037/0033-295X.105.3.442
- Borst, J. P., & Anderson, J. R. (2015). Using the ACT-R Cognitive Architecture in Combination With fMRI Data. In B. U. Forstmann & E.-J. Wagenmakers (Eds.), *An Introduction to Model-Based Cognitive Neuroscience* (pp. 339–352). New York, NY: Springer New York. doi: 10.1007/978-1-4939-2236-9_17
- Ell, S., Smith, D., Deng, R., & Hélie, S. (2020). Learning and generalization of within-category representations in a rule-based category structure. *Attention, Perception, and Psychophysics*, 82(5), 2448–2462. doi: 10.3758/s13414-020-02024-z
- Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003, July). Instance-based learning in dynamic decision making. *Cognitive Science*, 27(4), 591–635. doi: 10.1207/s15516709cog2704_2
- Ishizaki, T., Morita, H., & Morita, M. (2015). Feature integration in the mapping of multi-attribute visual stimuli to responses. *Scientific Reports*, 5, 1–7. doi: 10.1038/srep09056
- Jarvers, C., Brosch, T., Brechmann, A., Woldeit, M. L., Schulz, A. L., Ohl, F. W., ... Neumann, H. (2016, Novem-

- ber). Reversal Learning in Humans and Gerbils: Dynamic Control Network Facilitates Learning. *Frontiers in Neuroscience*, 10. doi: 10.3389/fnins.2016.00535
- Lommerzheim, M., Prezenski, S., Russwinkel, N., & Brechmann, A. (2020). Category Learning as a Use Case for Anticipating Individual Human Decision Making by Intelligent Systems. In T. Ahram, W. Karwowski, A. Vergnano, F. Leali, & R. Taiar (Eds.), *Intelligent Human Systems Integration 2020* (Vol. 1131, pp. 159–164). Cham: Springer International Publishing.
- Morita, J., Miwa, K., Maehigashi, A., Terai, H., Kojima, K., & Ritter, F. E. (2020, October). Cognitive Modeling of Automation Adaptation in a Time Critical Task. *Frontiers in Psychology*, 11, 2149. doi: 10.3389/fpsyg.2020.02149
- Nijboer, M., Borst, J., van Rijn, H., & Taatgen, N. (2016, May). Contrasting single and multi-component working-memory systems in dual tasking. *Cognitive Psychology*, 86, 1–26. doi: 10.1016/j.cogpsych.2016.01.003
- Prezenski, S., Brechmann, A., Wolff, S., & Russwinkel, N. (2017, August). A Cognitive Modeling Approach to Strategy Formation in Dynamic Decision Making. *Frontiers in Psychology*, 8, 1335. doi: 10.3389/fpsyg.2017.01335
- Smith, J. D., & Ell, S. W. (2015, September). One Giant Leap for Categorizers: One Small Step for Categorization Theory. *PLOS ONE*, 10(9), e0137334. doi: 10.1371/journal.pone.0137334
- Wolff, S., & Brechmann, A. (2012, January). MOTI: A motivational prosody corpus for speech-based tutorial systems. In (pp. 1–4).
- Wolff, S., & Brechmann, A. (2015, February). Carrot and stick 2.0: The benefits of natural and motivational prosody in computer-assisted learning. *Computers in Human Behavior*, 43, 76–84. doi: 10.1016/j.chb.2014.10.015
- Wolff, S., & Brechmann, A. (2022, December). Dorsal posterior cingulate cortex responds to negative feedback information supporting learning and relearning of response policies. *Cerebral Cortex*, bhac473. doi: 10.1093/cercor/bhac473

An Integrative Model of Human Response Processes in Raven’s Matrices

Can Serif Mekik (mekik.can_serif@courrier.uqam.ca)

Université du Québec à Montréal
Rensselaer Polytechnic Institute

Ron Sun (rsun@rpi.edu)

Rensselaer Polytechnic Institute

David Yun Dai (ydai@albany.edu)

State University of New York at Albany

Abstract

This paper presents Xyrast, an integrative model of human response processes on the Raven’s Matrices family of fluid intelligence tests, and reports on a simulation study addressing its response characteristics and verisimilitude. Xyrast is implemented in the Clarion cognitive architecture and models the influence of response strategy, working memory capacity, and persistence on performance. Simulations suggest that the model captures a wide range of phenomena offering, in some cases, novel explanations for observed results. These findings suggest several avenues for future research.

Keywords: clarion; cognitive modeling; cognitive architectures; raven’s matrices; fluid intelligence; response process; motivation

Introduction

Fluid intelligence (gF) may be defined as “[t]he ability to solve problems in unfamiliar domains using general reasoning methods” (Kyllonen and Kell, 2017). In brief, gF is one of the broadest and most prominent psychometric ability constructs in existence and appears to factor in many forms of thinking and reasoning (deductive, inductive, quantitative, analogical, etc.). Thus, gF is a natural candidate for cognitive modeling work at the intersection of psychometrics and cognitive science.

The Raven’s Matrices tests are among the most popular measures of gF , owing to their elegant format and strong psychometric properties (Kyllonen and Kell, 2017). By one estimate, up to 60% of score variance in Raven’s Matrices may be attributed to gF , with another 25% attributable to test-specific variance (see Gignac, 2015). This paper presents *Xyrast*, a new integrative model of human response processes on Raven’s Matrices. Xyrast is implemented in the Clarion cognitive architecture (Sun, 2016) and solves matrix problems in the style of the *Sandia Matrices* (Matzen et al., 2010), an open variant of Raven’s Matrices designed for use in psychological research. As evidenced below, the model accounts for important empirical phenomena and poses several questions for future research.

Model

Xyrast is a granular discrete-time model of human response processes in Raven’s Matrices. The model in-

cludes three main psychological parameters and introduces a novel visual reasoning complex for the Clarion cognitive architecture. Its design is closely informed by existing psychological findings, which are briefly reviewed below, and builds on existing cognitive models of the task (e.g., Carpenter et al., 1990; Kunda et al., 2013; Lovett and Forbus, 2017; Stocco et al., 2021).

The relevant psychological parameters are response strategy, working memory capacity (WMC), and persistence. The model simulates multiple response strategies following Newell’s (1973) injunction to “never average over methods” (p. 13). WMC is included because it is an important correlate of gF (Engle et al., 1999). Finally, persistence is included to account for motivational influences on performance, which have remained relatively unexplored despite theoretical and empirical evidence pointing to their importance (e.g., Raven, 2008; Wieber et al., 2010).

Psychological Considerations

Raven’s Matrices tests consist of a series of *matrix problems* in increasing order of difficulty (Raven, 2008). As depicted in Figure 1, matrix problems are multiple-choice pattern-completion problems. Subjects are asked to complete a visual figure matrix (i.e., a square array of figures where the bottom right entry is missing) by choosing the alternative that is the best fit. Matrix problems are constructed by varying visual features along matrix rows or columns according to various patterns. The choice of a correct alternative is thus taken to signal that the subject has correctly induced all patterns present in the matrix (see e.g., Carpenter et al., 1990).

Human response processes in Raven’s Matrices appear indeed to be characterized by an incremental process of pattern discovery, as revealed by analyses of errors, think-aloud protocols, and eye movements (e.g., Carpenter et al., 1990; Hayes et al., 2011; Vodegel Matzen et al., 1994). Discovered patterns are thought to inform response selection according to two strategies typically associated with multiple-choice tests (e.g., Hayes et al., 2011; Vigneau et al., 2006). The *constructive matching* strategy involves the construction of a response hypothesis that is matched to available alternatives for response selection. The *response elimination* strategy, on

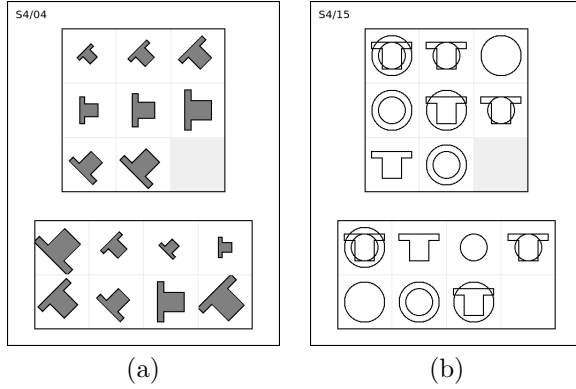


Figure 1: Matrix problems in the style of Sandia Matrices (Matzen et al., 2010).

the other hand, involves more direct evaluation of alternatives, via surface features, for their fit with accepted pattern hypotheses.

Crucially, the pattern discovery process appears to involve identifying and comparing properties of and relations among visual objects. The incremental nature of the pattern discovery process may be explained in part by human visual architecture. Indeed, the detection of visual relations appears to require sequential processing of visual objects (Franconeri et al., 2012). Thus, an important demand of the task appears to be the resolution of perceptual ambiguities (for identification of visual objects in crowded scenes) through the use of attentional resources (Primi, 2001). Comparing and retaining information about patterns may also be an important source of the task’s working memory demands, though the exact nature of these demands is a subject of debate (e.g. Unsworth and Engle, 2005). Finally, the evidence suggests that increased motivation improves performance through greater time investments (Wieber et al., 2010), which may reduce the chances of failing to account for the complete set of patterns governing an item (see Vodegel Matzen et al., 1994). That said, the relationship between item scores and latencies appears to be rather complex (Goldhammer et al., 2015).

Implementation Overview

In light of the considerations above, Xyrast tackles matrix problems using an interruptible inductive search that incrementally discovers patterns in item matrices. This discovery process involves sampling and testing pattern hypotheses using visual information from the item matrix.

Much of the model is structured according to the Clarion cognitive architecture (Sun, 2016). Clarion is a neurosymbolic hybrid cognitive architecture. It consists of four subsystems and exhibits a unique two-level representational structure. The architecture may be viewed as a network of neural networks where the *top level* cap-

tures explicit knowledge via rule-based processing on localist representations, and the *bottom level* captures implicit knowledge via associative networks and distributed representations. The two levels also interact via *chunks*, which link chunk nodes in the top level to (micro)feature nodes in the bottom level.

At the highest level, the architecture is organized as follows. The action-centered subsystem (ACS) is the primary locus of action control in Clarion and coordinates both overt behavior and cognitive processing. The non-action centered subsystem (NACS) maintains declarative knowledge (in both implicit and explicit forms) and is capable of retrieving task-relevant declarative memory chunks. The motivational subsystem (MS) maintains motivational states, which consist of motivational drive strengths in the bottom level and explicit goals in the top level. The metacognitive subsystem (MCS) houses modules dedicated to the maintenance, monitoring, and control of various internal parameters. Finally, the working memory module (WM) temporarily stores chunks retrieved by the NACS, monitors the similarity of such chunks to perceptual inputs and/or current cognitive states, and maintains a system of (implicit) flags that serve as internal cues for the ACS.

Xyrast simulates processing in 50 ms increments. At each time step, the model issues various external or internal actions. External actions include fixating on one of the sixteen panels in the current matrix problem, marking an alternative as a response choice, and submitting a response. Internal actions allow the model to control cognitive processing by setting attentional parameters, updating internal procedural cues, and retrieving declarative memory chunks into working memory. The model’s behavior is governed by a set of 916 fixed action rules, which are defined over a space of 1430 (micro)features. These rules are housed in the top level of the ACS and control behavior through a competitive process that accounts for the model’s current perceptual, cognitive, and motivational states.

Pattern discovery is enabled by Xyrast’s visual architecture, which is inspired by Feature Integration Theory (Treisman and Gelade, 1980). The visual architecture is founded on a dimensional *visual code*, in which visual objects are represented via synchronous activation of matching basic visual features. When multiple objects are present in a scene, the visual code is subject to a classic instance of the binding problem (see e.g., von der Malsburg, 1999). Thus, to work around this problem, Xyrast introduces a selective attentional filter that suppresses unwanted objects as directed by the ACS.¹ The model detects visual relations by sequentially fixating on visual objects. The resulting sequence of object representations is processed by an autoregres-

¹Details of the filtering process are omitted due to space limitations.

sive neural network in the bottom level of the NACS and drives the activation of relevant relational feature nodes. Using this complex of architectural components, Xyrast processes patterns pertaining to object size, orientation, shape, shading, and numerosity.

Visual patterns and response hypotheses are represented by chunks maintained in the NACS. For each pattern hypothesis, the model is programmed to retrieve into WM a corresponding pattern chunk. Under the constructive matching strategy, the model is additionally programmed to retrieve a response hypothesis chunk for each accepted pattern hypothesis. The model is not pre-loaded with any pattern or response hypothesis chunks. Instead, it automatically constructs new chunks in the NACS based on the novelty of the activation patterns in the bottom level. A second selective attentional filter, also directed by the ACS, controls the flow of bottom-up activations, allowing pattern hypotheses to guide chunk construction.

The model is programmed to test patterns and alternatives by monitoring perceptually-driven, bottom-up activations of relevant NACS chunks. A prerequisite for conditioning model behavior on NACS chunk activations in this way is to have the relevant chunks in WM. Reliable retrieval is therefore essential for performance. The retrieval process involves a competition among all chunks in the NACS. The probability that a chunk will be retrieved into WM from the NACS is given by a Boltzmann distribution over chunk activations. Generally speaking, the more activated a chunk, the more likely it is to be retrieved. The distribution has a temperature parameter that controls how strongly retrieval depends on chunk strengths and is (consequently) used to model WMC. As the temperature increases, retrievals become increasingly random.

Finally, the model persists on a given item if its motivation to solve the item outweighs the cost of time already sunk into that item. The motivation to solve an item is calculated on the basis of drive strengths in the MS and the relevance of the task goal to the agent's drives (see Sun et al., 2022). The cost associated with working on an item is calculated as a function of the base level activation (BLA) associated with the goal to solve the item. In Xyrast, goal BLAs decay monotonically after goal creation. Thus, the cost term tracks the time that has elapsed since the goal was set.

Simulations

Xyrast's response characteristics and verisimilitude are addressed in the following simulation study, which examines the model's behavior and ability to capture salient phenomena from the empirical literature. The study focuses on phenomena from three human-participants studies (discussed below), each of which addresses one of the model's three main psychological parameters: Gold-

hammer et al. (2015), Unsworth and Engle (2005), and Vigneau et al. (2006).

The Vigneau et al. (2006) study presents a window into the validity of the Xyrast's procedural programming. A key element of that study is an examination of correlations between scores, latencies, and eight eye-movement indices pertaining to response strategy. The matrix of correlations obtained from Vigneau et al.'s sample may be used to examine the verisimilitude of Xyrast's fixation patterns, which depend primarily on the implementation of its two strategies.

The Unsworth and Engle (2005) study is ideal for addressing the verisimilitude of Xyrast's WM mechanisms. The study investigates how the correlation between WMC and item scores varies as a function of item difficulty in order to examine the role of WMC in performance, predicting that, if WMC demands are primarily storage-related, WMC-score correlations should increase as a function of item difficulty (as reflected by item ordering) whereas, if these demands are primarily related to attention control, these correlations should not vary systematically with item difficulty. Empirically, WMC-score correlations are found to be relatively steady for easy and moderate items, supporting the attention control explanation. Unexpectedly, the correlations are found to decrease among the most difficult items.

Finally, the Goldhammer et al. (2015) study presents an opportunity to examine the verisimilitude of Xyrast's motivational mechanisms. Goldhammer et al. analyze how the relationship between item score and latency varies as a function of subject ability and item difficulty using generalized linear mixed models. They find a negative fixed effect of item latency on scores that appears to diminish (possibly even reverse) among lower-ability subjects and higher-difficulty items.

Setup

The simulations include a total 8,640 responses from 144 simulated subjects on a set of 60 generated items in the style of the Sandia Matrices (Matzen et al., 2010). Between items, simulated subjects' WMs and declarative memories are cleared. The simulation design is fully crossed and includes 5 item classes (see below), 2 strategy levels (re, cm), 4 WMC levels (temperature settings of 2^{-1} , 2^{-3} , 2^{-5} , and 2^{-7}), and 3 persistence levels (.33, .66, and 1.0).² There are therefore 6 subjects and 12 items per cell. For each item-subject pair, scores, latencies, fixation sequences, and fixation durations are recorded.

Persistence levels are varied using the strength of the model's achievement drive (ach). These achievement drive settings correspond, in respective order, to expected time investments of 24.75, 45.5, and 75 seconds

²WMC and persistence levels were selected manually to maximize variation in scores.

in simulated time. Simulation runs are limited to 90 simulated seconds per subject-item pair, after which the simulation is stopped and a timeout event is recorded. This measure serves to abort simulation runs which have entered a corrupted state.

The five item classes vary the source and level of item difficulty and are named, in increasing order of theoretical difficulty (see, e.g., Matzen et al., 2010; Primi, 2001), OT-1, OT-2, OT-3, LP-2, and LP-3. The OT prefix designates *object transformation* problems, which vary in their *structural complexity*, as measured by the number and type of distinct patterns used to generate attribute variation. The LP prefix designates *logic problems*, which vary in their *perceptual complexity*, as measured by the maximum number of overlapping objects allowed in individual matrix panels. In Figure 1, items (a) and (b) are OT-2 and LP-3 problems, respectively.

Results

Of all item responses, only 1.67% ($N = 144$) are associated with timeout events, the vast majority of which are on LP problems ($N = 139$). These responses are excluded from all analyses. Among items included in the analysis, the grand mean for item scores is 51% and the grand mean for response latencies is 39.69 seconds.

Regarding item characteristics, a clear negative relationship exists between the by-item mean scores and latencies ($\tau = -.797$). A similar pattern is observed when scores and latencies are averaged by item class. Average scores show a clear increasing trend, and average latencies a clear decreasing trend, as a function of theoretical item difficulty. The only exception to this pattern is the OT-3 item class which presents characteristics similar to the OT-1 class.³

All considered, the model takes more time to process more difficult items, replicating a well-known phenomenon in human studies (see, e.g., Goldhammer et al., 2015; Vigneau et al., 2006), and simulated item-class difficulty levels accord well with theoretical expectations. The case of the OT-3 class may be explained by a lack of plausible alternatives in these items due to unanticipated limitations in the alternative generation procedure. A similar problem affects the original Sandia Matrix generation software (Matzen et al., 2010). This result therefore suggests that the contribution of alternatives to item difficulty, often neglected in formal analyses of matrix problems, deserves greater scrutiny.

Turning to subject characteristics, Table 1 presents two linear probability models quantifying the influence of the model’s psychological parameters on scores.⁴ These regression models are selected based on a hierarchical analysis comparing four specifications: main effects

³This observation is corroborated by a regression analysis, omitted here due to space limitations.

⁴Similar analyses were also performed for latencies but are omitted due to space limitations.

	(1)	(2)
cm	-21.062*** (2.151)	-22.001*** (3.938)
wmc	42.994*** (3.288)	16.812*** (3.850)
ach	26.443*** (2.761)	26.958*** (3.074)
wmc:cm		27.750*** (4.008)
ach:cm		-19.625*** (3.308)
wmc:ach		18.561*** (4.955)
Intercept	22.906*** (3.235)	29.558*** (3.291)
Observations	8,496	8,496
R^2	0.169	0.184
Adj. R^2	0.168	0.183
F-Stat.	123.209***	85.983***

* $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$

Standard errors clustered by subject and item

Table 1: Regression of subject parameters on scores (%)

only, all two-way interactions except WMC-persistence, all two-way interactions between the three psychological parameters, and a full three-way model. The preferred model is the one that includes all two-way interactions among the three psychological parameters. Table 1 shows the main-effects model (Model 1) and the preferred model (Model 2). In these regressions, response strategy is coded using a dummy variable for constructive matching and WMC is coded to range between 0 and 1. For WMC, 0 corresponds to the lowest level, 1 corresponds to the highest level, and the remaining levels divide the interval evenly. Achievement drive strengths are included in the regressions without any additional processing.

The regression analyses reveal that Xyrast exhibits a negative effect of constructive matching on scores at the aggregate level (Model 1). This result is in tension with findings that, in human samples, constructive matching is typically associated with higher scores and more efficient processing (see, e.g., Hayes et al., 2011; Vigneau et al., 2006). However, it is worth taking note of the substantively large positive interaction between constructive matching and WMC (Model 2). This interaction effect more or less nullifies the score loss incurred by the use of constructive matching at the highest-WMC levels. Mechanically, the negative effect of constructive matching on scores is attributable to the fact that, in Xyrast, constructive matching demands a greater number of successful retrievals (per pattern) than response

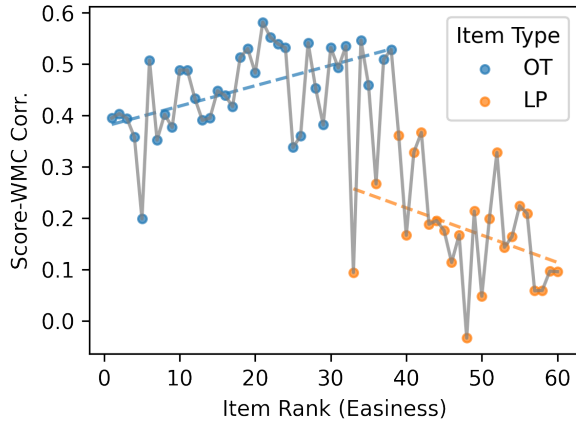


Figure 2: Score-WMC correlations as a function of item rank.

elimination. Taken together, these findings suggest that, in Xyrast, scores are more sensitive to WMC under constructive matching and that the apparent relationship between scores and strategy choice among human subjects may be driven by a process of adaptive strategy selection.

Regarding the model’s ability to capture the target phenomena, the results are as follows. There is moderate agreement between the correlation matrix for performance and strategic eye-movement indices reported in Vigneau et al. (2006) and a corresponding matrix obtained from the simulation data ($\tau = .439$, $p < .001$ by permutation test). This agreement improves if samples are weighted in accordance with the notion that higher-ability (i.e., higher scoring) subjects are more likely to use the constructive matching strategy ($\tau = .518$, $p < .001$ by permutation test). These results support the verisimilitude of Xyrast’s procedural programming.

As depicted in Figure 2, when items are ordered from easiest to most difficult (according to simulated scores), simulated WMC-score correlations vary with item rank in a manner similar to the findings of Unsworth and Engle (2005). Interestingly, when item rank is regressed against scores, a small increasing trend is observed among OT items ($p < .001$) and a similarly small decreasing trend is observed among LP items (n.s.). These results suggest that Unsworth and Engle’s findings may be driven, at least in part, by qualitative differences in item demands.

Last, the model’s behavior also appears to agree with the findings of Goldhammer et al. (2015), as evidenced by replication of the main mixed model from that study. The mixed model predicts the probability of a correct response as a function of (log-transformed) item latencies with by-item and by-subject random effects and slopes. Similar to humans, the simulations exhibit a negative

fixed effect of item latency on the probability of a correct response. Also in agreement with the human results, by-subject intercepts and adjustments to the latency effect are negatively correlated. The same is true for by-item intercepts and adjustments to the latency effect. Thus, Xyrast captures the finding that the negative effect of latency on scores is diminished among low-ability subjects and high-difficulty items.

Discussion

Xyrast captures many aspects of human response processes in Raven’s Matrices, as evidenced in the preceding discussion. Indeed, Xyrast’s behavior agrees with a rather wide range of empirical findings. Furthermore, in many cases, the model offers novel explanations for observed phenomena which warrant further investigation.

That said, Xyrast’s most important contribution is its ability to account for interactions among multiple psychological parameters. The model encourages a fundamentally conative analysis of the Raven’s Matrices task, where cognitive factors (e.g., item demands, strategy choice, WMC) influence time costs while motivational factors contribute to time investments. Crucially, this analysis integrates classical cognitive considerations with motivational considerations, providing for a nuanced and detailed analysis of human response processes.

To close, it is notable that the relationship between the model’s performance and its basic psychological parameters appears to depend on contingent factors like strategy choice. This phenomenon may have some important implications for developing mechanistic theories of gF . Namely, extraneous variability in the relationship between performance on gF -test items and basic psychological variables, like WMC or persistence, presents a potential challenge for explaining gF in terms of more basic psychological processes, echoing some concerns raised in Hunt (1987). That said, such variability need not pose a problem if, for instance, it contributes only to test-specific variance. Either way, closer collaboration between empirical and modeling work seems necessary to advance our understanding of the mechanisms underlying fluid intelligence.

Acknowledgements

This work was supported in part by ARI grant W911NF-17-1-0236 to Ron Sun and David Yun Dai. Can Serif Mekik held a HASS Graduate Fellowship at the Rensselaer Polytechnic Institute for part of this work. The authors thank Dr. Laura E. Matzen for providing them with the Sandia Matrices and the Sandia Matrix Generation Tool (Matzen et al., 2010).

References

- Carpenter, P. A., Just, M. A., & Shell, P. (1990). What one intelligence test measures: A theoretical account of the processing in the Raven Progressive Matrices test. *Psychological Review*, 97(3), 404–431. <https://doi.org/10.1037/0033-295X.97.3.404>
- Engle, R. W., Tuholski, S. W., Laughlin, J. E., & Conway, A. R. A. (1999). Working memory, short-term memory, and general fluid intelligence: A latent-variable approach. *Journal of Experimental Psychology: General*, 128(3), 309–331. <https://doi.org/10.1037/0096-3445.128.3.309>
- Franconeri, S. L., Scimeca, J. M., Roth, J. C., Helseth, S. A., & Kahn, L. E. (2012). Flexible visual processing of spatial relationships. *Cognition*, 122(2), 210–227. <https://doi.org/10.1016/j.cognition.2011.11.002>
- Gignac, G. E. (2015). Raven's is not a pure measure of general intelligence: Implications for g factor theory and the brief measurement of g. *Intelligence*, 52, 71–79. <https://doi.org/10.1016/j.intell.2015.07.006>
- Goldhammer, F., Naumann, J., & Greiff, S. (2015). More is not always better: The relation between item response and item response time in Raven's Matrices. *Journal of Intelligence*, 3(1), 21–40. <https://doi.org/10.3390/jintelligence3010021>
- Hayes, T. R., Petrov, A. A., & Sederberg, P. B. (2011). A novel method for analyzing sequential eye movements reveals strategic influence on Raven's Advanced Progressive Matrices. *Journal of Vision*, 11(10), 10–10. <https://doi.org/10.1167/11.10.10>
- Hunt, E. (1987). Science, technology, and intelligence. In R. R. Ronning, J. A. Glover, J. C. Conoley, & J. C. Witt (Eds.), *The influence of cognitive psychology on testing*. Lawrence Erlbaum Associates. <https://digitalcommons.unl.edu/buroscogpsych/5/>
- Kunda, M., McGreggor, K., & Goel, A. K. (2013). A computational model for solving problems from the Raven's Progressive Matrices intelligence test using iconic visual representations. *Cognitive Systems Research*, 22-23, 47–66. <https://doi.org/10.1016/j.cogsys.2012.08.001>
- Kyllonen, P., & Kell, H. (2017). What is fluid intelligence? Can it be improved? In M. Rosén, K. Y. Hansen, & U. Wolff (Eds.), *Cognitive abilities and educational outcomes* (pp. 15–37). Springer-Verlag. https://doi.org/10.1007/978-3-319-43473-5_2
- Lovett, A., & Forbus, K. (2017). Modeling visual problem solving as analogical reasoning. *Psychological Review*, 124(1), 60–90. <https://doi.org/10.1037/rev0000039>
- Matzen, L. E., Benz, Z. O., Dixon, K. R., Posey, J., Kroger, J. K., & Speed, A. E. (2010). Recreating Raven's: Software for systematically generating large numbers of Raven-like matrix problems with normed properties. *Behavior Research Methods*, 42(2), 525–541. <https://doi.org/10.3758/brm.42.2.525>
- Newell, A. (1973). You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In W. G. Chase (Ed.), *Visual Information Processing*. Academic Press. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.982.3132&rep=rep1&type=pdf>
- Primi, R. (2001). Complexity of geometric inductive reasoning tasks. *Intelligence*, 30(1), 41–70. [https://doi.org/10.1016/S0160-2896\(01\)00067-8](https://doi.org/10.1016/S0160-2896(01)00067-8)
- Raven, J. (2008). The Raven Progressive Matrices Tests: Their theoretical basis and measurement model. In J. Raven & C. J. Raven (Eds.), *Uses and abuses of intelligence: Studies advancing Spearman and Raven's quest for non-arbitrary metrics* (pp. 17–68). Royal Fireworks Press.
- Stocco, A., Prat, C. S., & Graham, L. K. (2021). Individual differences in reward-based learning predict fluid reasoning abilities. *Cognitive Science*, 45(2). <https://doi.org/10.1111/cogs.12941>
- Sun, R. (2016). *Anatomy of the mind: Exploring psychological mechanisms and processes with the clarion cognitive architecture*. Oxford University Press.
- Sun, R., Bugrov, S., & Dai, D. (2022). A unified framework for interpreting a range of motivation-performance phenomena. *Cognitive Systems Research*, 71, 24–40. <https://doi.org/10.1016/j.cogsys.2021.09.003>
- Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, 12(1), 97–136. [https://doi.org/10.1016/0010-0285\(80\)90005-5](https://doi.org/10.1016/0010-0285(80)90005-5)
- Unsworth, N., & Engle, R. (2005). Working memory capacity and fluid abilities: Examining the correlation between Operation Span and Raven. *Intelligence*, 33(1), 67–81. <https://doi.org/10.1016/j.intell.2004.08.003>
- Vigneau, F., Caissie, A. F., & Bors, D. A. (2006). Eye-movement analysis demonstrates strategic influences on intelligence. *Intelligence*, 34(3), 261–272. <https://doi.org/10.1016/j.intell.2005.11.003>
- Vodegel Matzen, L. B. L., van der Molen, M. W., & Dudink, A. C. (1994). Error analysis of raven test performance. *Personality and Individual Differences*, 16(3), 433–445. [https://doi.org/10.1016/0191-8869\(94\)90070-1](https://doi.org/10.1016/0191-8869(94)90070-1)
- von der Malsburg, C. (1999). The what and why of binding. *Neuron*, 24(1), 95–104. [https://doi.org/10.1016/S0896-6273\(00\)80825-9](https://doi.org/10.1016/S0896-6273(00)80825-9)
- Wieber, F., Odenthal, G., & Gollwitzer, P. (2010). Self-efficacy feelings moderate implementation intention effects. *Self and Identity*, 9(2), 177–194. <https://doi.org/10.1080/15298860902860333>

ACT-R Modeling of Rapid Motor Learning Based on Schema Construction

Kazuma Nagashima (nagashima.kazuma.16@shizuoka.ac.jp),

Jumpei Nishikawa (nishikawa.jumpei.16@shizuoka.ac.jp) ,

Ryo Yoneda (yoneda.ryo.17@shizuoka.ac.jp) ,

Junya Morita (j-morita@inf.shizuoka.ac.jp) ,

Department of Informatics, Graduate School of Science and Technology, Shizuoka University,
3-5-1 Johoku, Naka-ku, Hamamatsu-shi, Shizuoka-ken, 432-8011 Japan

Abstract

The environment surrounding organisms changes dynamically, and humans acquire motor skills by improving the prediction of such environmental changes. The research on cognitive architectures has so far proposed several mechanisms explaining the process of human motor learning. Adaptive Control of Thought-Rational (ACT-R), one of the representative cognitive architectures, has perceptual and motor modules for interaction with the external environment. However, the performance of these modules is insufficient for real-time environments, especially in terms of learning speed. This study proposes a method to simulate human-level rapid motor learning using a pre-trained motor learning module. We assume that in a novel perceptual-motor task, a pre-trained motor schema is rediscovered/recalled. In the simulations, we trained the motor learning module in advance and conducted a simulation where difficulties of rediscovering schemata were manipulated. As a result, we confirmed that the pre-trained phase increased the human-model fitting in motor learning.

Keywords: motor schema theory; perceptual-motor task; cognitive modeling; ACT-R

Background

The environment surrounding organisms change dynamically. To enable quick and accurate action on the environment, humans acquire patterns predicting those changes. Research related to cognitive architecture has long been concerned with such motor learning processes. Adaptive Control of Thought-Rational (ACT-R) (Anderson, 2007), which is one of the representative cognitive architectures (see exhaustive review by Kotseruba and Tsotsos (2020)), has perceptual and motor modules responsible for interaction with the external world. Historically, these modules were developed in another cognitive architecture, Exclusive Process-Interactive Control (EPIC) (Kieras & Meyer, 1997) and later integrated into an official ACT-R through a version called ACT-R/PM (Byrne, 2000) ¹.

The perceptual and motor modules in ACT-R can interact with higher cognitive functions such as memory and goal management. This architectural characteristic allows the flexible model adaptation to the external world. However, these modules only work well in environments requiring no immediate responses. Each of ACT-R's modules has its own dedicated buffer. The continuous signal input to the perceptual module is converted into discrete symbols through the buffer,

and the symbols are sent to the buffer of the motor module. Therefore, there is a large time cost in the perceptual-motor process. On the other hand, Common Model of Cognition (CMC) proposed by Laird, Lebiere, and Rosenbloom (2017) aiming to integrate several cognitive architectures proposes to send the input of the perceptual module directly to the motor module without going through buffers.

There are studies applying ACT-R's perceptual and motor modules to real-time tasks. However, these studies used motor commands developed in a program outside ACT-R basic functions. For example, the driving model developed by Salvucci (2006) simulated human-level complex motor operations such as steering, accelerating, and braking with simple mathematical functions. There is also a module (tracker module) that learns the numerical parameters involved in motor actions to interact with the external world without these external programs (Anderson, Betts, Bothell, Hope, & Lebiere, 2019; Gianferrara, Betts, & Anderson, 2021). However, those studies have not shown that movements involving a large number of parameters are acquired at the same speed as in humans. Based on the above background, we propose an approach to represent human performance in perceptual-motor tasks without using externally programmed motor sequences. In our approach, the model stores the previously learned motor sequences as a schema and rediscovers them in novel situations.

In the following section, we will introduce related studies concerning the abovementioned goal of the study. Following this, the target human behaviors concerning human motor learning in perceptual-motor tasks will be presented. The simulation results will present a model performance compared to human behaviors. In the final section, we will discuss the implications and limitations of this study.

Related Works

This study aims to represent human performance in perceptual-motor tasks in ACT-R. This section introduces two directions of previous studies: human and ACT-R perceptual motor learning.

Human Perceptual Motor Learning

There are cognitive aspects to human motor learning. Some researchers claim that the process of motor expertise is facilitated by verbalization (Suwa, 2008). As a background of

¹See Ritter, Tehranchi, and Oury (2019) as a review of the development history of ACT-R.

such aspect, Fitts (1964) proposed a nonlinear theory, which divides human motor learning into three levels. The levels correspond to a cognitive phase in which prior knowledge is used, an associative phase in which prior knowledge and movement begin to connect directly, and an autonomous phase in which all movement is automatic. Human motor learning can be viewed as a process of transition from the cognitive phase to the autonomous phase.

There is also a theory that explains the learning of motor movement as the acquisition of a schema representing a pattern of connecting perceptual and motor program (Schmidt, 1975). This theory assumes two types of schemata, *recall schema* and *recognition schema*, are acquired through perceptual-motor tasks. The recall schema consists of a set of motor parameters, past motor content, and past motor outcomes, while the recognition schema consists of a set of motor parameters, past motor senses, and past motor outcomes. Through the recall and recognition schemata, the content and sensory information of past motion are recalled respectively. By acquiring and utilizing this abstract knowledge of movement, humans can quickly acquire movement patterns in novel situations.

As described above, there is a link between motor learning and cognitive processes. Cognitive architectures are useful to explain these connections in detail. Therefore, in the following, we refer to the model developed using ACT-R.

ACT-R Model of Perceptual Motor Learning

Several studies have used ACT-R to explain internal human processes during complex perceptual-motor tasks such as a video game environment (Anderson et al., 2019; Gianferara et al., 2021). The main claim of those studies is that the nonlinear mastering process of motor skills (Fitts, 1964) is represented by combining the motor learning module with other higher-order cognitive modules. In addition, those studies implemented a game state buffer that describes temporal relationships between objects projected into the visual field. They reproduced the human performance by making the state of the game state buffer as the condition for activating the motor commands.

The other studies utilize a simpler task to manipulate experimental variables involved in perceptual-motor processes. The ACT-R model constructed by Morita et al. (2020) performs the simple task of manipulating a circular object to follow a scrolling line. In the experiment that the model simulated, participants followed the object either manually or automatically. Under conditions in which the performance of those manipulation modes varied, they investigated the mechanism of dependence on automation. To obtain results compatible with human performance, the manual mode of this model was controlled by the ACT-R's perceptual and motor modules with the support of an external program for determining the timing of key presses ². Concerning this limi-

²See the source code obtained from <https://github.com/j-morita-shizuoka/line-following-tak>

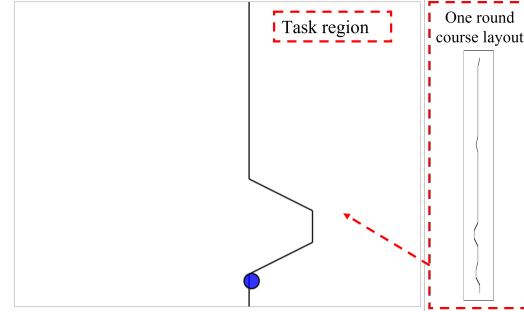


Figure 1: Task interface.

tation, Nagashima, Nishikawa, Yoneda, Morita, and Terada (2022) modified the model to include the ACT-R motor learning module to realize motor function in this task without any programmatic assistance. Although Nagashima et al. (2022) succeeded in simulating the interaction between motor learning and task arousal, which was the target of their study, they failed to simulate human-level task performance.

As shown in the above, ACT-R can simulate perceptual-motor tasks that require immediate responses with some limitations. However, as already mentioned, existing research has not revealed a mechanism to replicate human learning rates in situations involving a large number of parameters. The motor learning module of ACT-R learns parameters that map perception to motor action from a trial-error history. Such model learning usually requires more trials than human learning. Against this background, this study investigates the mechanism of convergence of parameters coordinating between perception and motor actions in a short period.

Human Data

To obtain data on human motor learning, we set up the line-following task, as in previous studies. Figure 1 shows the task interface. In this task, participants manipulate a blue circle called a vehicle to follow a polyline that automatically scrolls from top to bottom on the screen. The line changes according to a predefined course, as shown on the right side of Figure 1. The line is drawn by combining 48 pixels high line patterns of varied angles (30, 45, 90, 135, and 150 degrees). The course repeats from the beginning approximately every minute ³. The repetition of the course is called a *round*. Participants perform these operations for 30 minutes.

The data covered in this study were obtained in an online experiment via recruitment on a crowdsourcing website (Lancers.jp) in January 2023. The data from twenty-four participants were the target of the analysis, excluding incomplete data from a total of 50 participants.

Figure 2 shows the performance of the task. The vertical axis shows the offline rate (the percentage of time that the vehicle did not follow the line in the total time of the segment), and the horizontal axis shows the time that the 30-minute task

³Scroll speed is 40 fps, the total course length is 1500 frames.

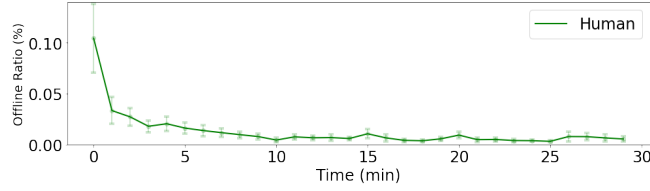


Figure 2: Human data. Error bars indicate a standard error of mean.

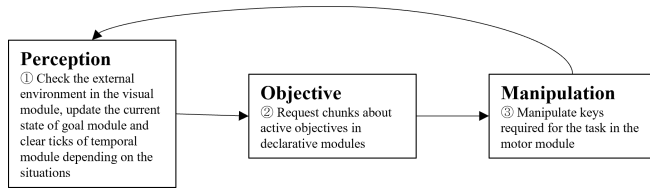


Figure 3: Block diagram showing basic model processing.

was segmented into 1-minute intervals. The graph shows a decreasing tendency of the offline rate with time. In particular, we confirm that the decrease in the offline ratio occurs rapidly at the beginning of the task.

Model

This section introduces a model of the line-following task. The focus of this study is on learning in the motor module. In particular, we aim to express the cognitive mechanisms behind rapid learning as shown in Figur 2. In the following, we describe the structure and process of the model and then describe the mechanism of motor learning.

Module and Process

While following the previous models (Morita et al., 2020; Nagashima et al., 2022), this study focuses on the fitting between human and model performance. The internal state transit of the model is shown in Figure 3. As seen in the figure, the model consists of cyclic behaviors of perceptual and motor processing. Because individual cognitive functions support this process, this section explains the model according to the ACT-R modules. Among several modules implemented in ACT-R, the model uses the visual, motor, goal, and production modules. These modules function as follows.

Visual Module This module simulates interaction with the external environment (a virtually created window) by recognizing symbols needed to perform a task (e.g., the position of a vehicle or the angle of a line). Figure 4 shows the virtual interface for the line following task⁴. Similar to humans, the model manipulates the vehicle based on a positional relationship to the line to be followed. The relationship is indicated by two lines drawn from the top of the vehicle toward the scrolling line.

⁴The courses presented in the figure are different from the ones used in the simulation (same as Figure 1) for explanation purposes.

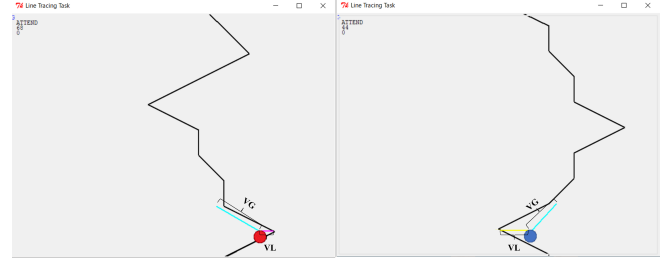


Figure 4: Model interface.

- **VL (Vehicle to Line):** It indicates how well the model follows the current scrolling line. If the length of the VL is less than or equal to the vehicle's radius, the model successfully performs the task. This on/offline status is also indicated as the vehicle's color (red: online, blue: offline). The color of the VL is also changed according to the positional relation between the vehicle and the scrolling line. When the vehicle is positioned to the left of the scrolling line, the VL colors magenta, while when the vehicle is positioned to the right, the VL colors yellow. By recognizing these colors, the model determines which direction the vehicle needs to go next.
- **VG (Vehicle to Goal):** It indicates the direction in which the model will head in the future (subgoal). The subgoals are placed at the next contact points of each 48 pixels pattern. In addition, it is placed at the x-coordinate offset by the radius of the vehicle from the line to be followed so that the shortest path where the line and the vehicle touch can be traced.

Goal Module This module stores the current state of the task to control the flow. In this study, the state includes the lengths of VG and VL, the left-right relationship between the vehicle and the scrolling line.

Motor Module This module simulates the operations required by the task. In the line-following model, key presses are performed to move the vehicle. Nagashima et al. (2022) prepared four operations: *Stop* (release the key), *Left* (press the key assigned to the left), *Right* (press the key assigned to the right), and *Continue* (continue the previous operation). This study adds *Left Punch* (briefly press the key assigned to the left twice) and *Right Punch* (briefly press the key assigned to the right twice) to follow 45 and 135 degrees angle lines.

The selection of these operations is conditioned on the lengths (continuous values) of VL and VG held in the goal module. Table 1 summarizes those conditions for each action selection. The role of each row indicates as follows.

1. *Current Action:* Current action.
2. *Position:* Relative position of the model to the scrolling line.

3. *Goal*: The goal that the model is heading. *Far* represents the subgoal and *Near* represents the scrolling line.
4. *Online*: The distinction of the state that model is following the line.
5. *Tracker*: Conflict resolution criteria using the vehicle coordinates and the motor learning module adjustment values.

Continue is chosen when none of these conditions is satisfied.

Production Module This manipulates the other modules by selecting and applying production rules using chunks held by the other modules. In the current model, the application of this module results in the flow shown in Figure 3.

1. *Perception*: The model sees the state of the external environment in the visual module and it updates the current state.
2. *Objective*: The model recalls chunks about the current goal.
3. *Manipulation*: The model operates a vehicle to trace the line.

The time required for these state transitions is determined by the method defined in ACT-R (e.g., the default production execution time of 50 ms).

Mastering Motor Control

The accuracy of the perceptual-motor loop (Figure 3) is improved by learning through the task. This learning is controlled by a tracker module included in ACT-R 7.27, initially proposed by Anderson et al. (2019). This module adjusts the continuous conditions for selecting motor operations based on positive and negative feedback from the environment and simulated annealing algorithm (Kirkpatrick, Gelatt Jr, & Vecchi, 1983).

In this study, positive feedback is generated when the model goes from the offline state (viewing the blue vehicle) to the online state (viewing the red vehicle). Negative feedback is generated in the opposite pattern. Based on the values of these feedbacks, the parameter θ in Table 1 is adjusted to be optimal to follow the line. If this optimization fails, the model cannot trace the line because the vehicle either crosses the line or stops moving before reaching the line.

The problem with such learning is the time to find optimal combinations of the parameter values because the number of combinations becomes large depending on the size of the parameter set. To reproduce the rapid learning observed in the human experiment, we assume a fixed set of parameters as a motor schema (Schmidt, 1975). In more detail, we suppose that the rapid learning observed in Figure 2 was accomplished by recalling (or rediscovering) the motor schema held in the individual's prior memory. Even participants who perform this task for the first time have experienced playing similar games in the past. Thus, they could recall such motor schema coordinating perceptual and motor processing during the task, rapidly improving their performance.

To examine this hypothesis, in the following simulation, we perform a pseudo-pre-learning phase (schema construction phase) to construct the optimal parameter set for this task in advance. Then, in the phase that reproduces the human experiment (schema application phase), we aim to represent human-level rapid learning in a situation where the pre-constructed parameter set (schema) can be easily discovered.

Simulation

In this simulation, we set two phases: the preliminary schema construction phase and the schema application phase. In the later phase, the discoverability of the constructed schema was manipulated. Furthermore, to strengthen the alignment with human performance, we manipulated the *dat* parameter, which defines the firing time of productions in ACT-R. The default setting of this parameter is 0.05. However, past models of ACT-R have shown delays in reaction time if it uses the default setting. To validate the performance of the base perceptual-motor process in ACT-R, this study examined a setting ($dat = 0.025$) that allows for twice as fast reactions in addition to the default setting.

Procedure

The following procedures were performed for each phase.

Schema Construction We assume that the optimal parameters (a motor schema) are obtained by learning over a long period of time for a sufficiently large parameter space. To apply this assumption, the following procedure is executed.

1. Set the value range of each θ to $min_{Construct} = -64$ and $max_{Construct} = 64$ centered at 0. This range is divided into 128 segments by the control-count parameter of this module.
2. Run a three-hour simulation, compared to 30 minutes for the human experiment.
3. Search for rounds that had an offline ratio of 0 (perfectly follow the line) in the last hour.
4. For each of those rounds, the difference between the maximum and minimum values for each θ is calculated and add them together.
5. Select the round with the lowest total range of θ as the candidate for the best parameter.

In this study, we performed the above procedure 10 times.

Schema Application We assume that the optimal parameters discovered above are acquired by participants prior to the start of the experiment. Thus, the learning process during the experiment is considered to be the process of rediscovering the motor schema. This process of rediscovery is considered to be a search among a limited set of parameters. Based on this assumption, the following procedures are executed in the schema application phase.

1. Set the range of each θ to constants defined by max_{Apply} and min_{Apply} , centered on the optimal value obtained in the

Table 1: Condition-action correspondence.

	Left		Left Punch		Right		Right Punch		Stop					
Current action	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Left	Left	Left	Right	Right	Right
Position	Right	Right	Right	Right	Left	Left	Left	Left	Right	Right	Left	Left	Left	Right
Goal	Far	Near	Far	Near	Far	Near	Far	Near	Far	Near		Far	Near	
Online		False		False		False		False						
Tracker	$vg_x \geq vg_y + \theta_5$ $vg_x > \theta_1$ $vl \geq vg + \theta_4$	$vl > \theta_1$ $vl > \theta_2$ $vl < vg + \theta_4$	$vg_x < vg_y + \theta_5$ $vg_x > \theta_1$ $vl \geq vg + \theta_4$	$vl > \theta_1$ $vl \leq \theta_2$ $vl < vg + \theta_4$	$vg_x \geq vg_y + \theta_5$ $vg_x > \theta_1$ $vl \geq vg + \theta_4$	$vl > \theta_1$ $vl > \theta_2$ $vl < vg + \theta_4$	$vg_x < vg_y + \theta_5$ $vg_x > \theta_1$ $vl \geq vg + \theta_4$	$vl > \theta_1$ $vl \leq \theta_2$ $vl < vg + \theta_4$	$vl \geq vg + \theta_6$ $vg_x \leq \theta_3$	$vl \leq vg + \theta_6$ $vl \leq \theta_3$		$vl \geq vg + \theta_6$ $vg_x \leq \theta_3$	$vl \leq vg + \theta_6$ $vl \leq \theta_3$	

* vg denotes the distance to the goal. vg_x and vg_y denote its xy components. vl denotes the distance to the line. Each θ is a correction value adjusted by the motor learning module.

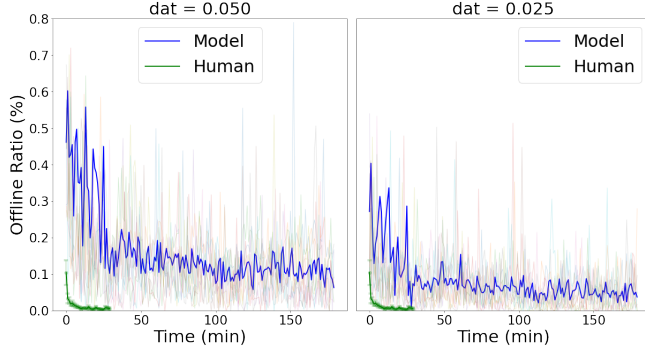


Figure 5: Result of schema construction. The thick lines are averages, and the thin lines are individual cases. ($n = 10$)

schema construction phase. The range is divided into three segments by the control-count parameter.

2. Run a 30 minutes simulation.

The above settings make it easier for the model to discover the constructed optimal parameters as a schema by decreasing the divisions. In this study, max_{Apply} and min_{Apply} were set to the same values, and three conditions were prepared, each setting the parameters to 32, 64, and 128. The other parameters of the motor learning module were set as same as Gianferrara et al. (2021).

Results

Schema Construction Figure 5 shows the offline ratio for three hours, obtained in the process of pre-training with two different dat parameters. The blue and green series represent the model and the human results (a reference) respectively. Although neither the blue line of dat setting did not reach the human performance, there were individual cases where it outperformed humans. Furthermore, the blue line can be seen that as the task progresses, the offline ratio became lower and the performance of the model improved. We can also confirm that the lower dat led to an overall improvement in performance.

For each dat setting, we searched for rounds in which the offline ratio was zero in the last hour and the parameters were less variable. The maximum and minimum values for each θ in the obtained rounds are summarized in table2. The range of values from -64 to 64 was searched for all parameters, but

 Table 2: Ranges of θ after the schema construction.

	$dat = 0.05$		$dat = 0.025$	
	min	max	min	max
θ_1	-35.8	-33.8	2.5	7.6
θ_2	-9.6	-7.6	-33.8	-29.7
θ_3	15.6	24.7	17.6	25.7
θ_4	-48.9	-28.7	21.7	31.7
θ_5	-64.0	-46.9	12.6	14.6
θ_6	63.0	64.0	60.0	64.0

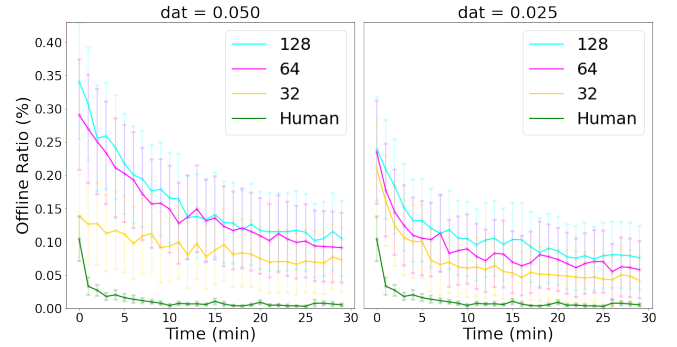


Figure 6: Result of schema application. Error bars in the offline ratio indicate a standard error of mean. ($n = 100$)

the obtained range was different for each parameter. For example, the range of θ_3 is larger than that of θ_2 , etc. In this study, the median value between the maximum and minimum values for any of the parameters was considered the optimal parameter.

Schema Application Figure 6 shows the simulation results of the model with the constructed schema. Each series indicates a different max_{Apply}/min_{Apply} setting. In both dat settings, we can confirm that the narrower the width of the value range leads to better initial performance. It can also be confirmed that except for a 32 in $dat=0.050$ condition, the human characteristic of rapid learning in the early stages of the task was reproduced. Note that a smaller dat setting improves the final performance and is closer to the human performance than the default dat setting.

Table 3 shows the model-human fitting. It can be seen that for each dat setting, the correlation is higher in the schema application phase than in the schema construction

Table 3: Pearson correlation coefficients between human and model results.

	$dat = 0.05$		$dat = 0.025$	
base	0.34	$(10^0 \times 0.12)$	0.33	$(10^{-1} \times 0.36)$
128	0.78	$(10^{-1} \times 0.25)$	0.85	$(10^{-2} \times 0.99)$
64	0.75	$(10^{-1} \times 0.20)$	0.90	$(10^{-2} \times 0.65)$
32	0.69	$(10^{-2} \times 0.64)$	0.90	$(10^{-2} \times 0.37)$

* The numbers in parentheses indicate Root Mean Squared Error (RMSE). The *base* is for the first 30 minutes during schema construction. The others correspond to the series in Figure 6. The bold numbers indicate the best fit.

phase (base). The RMSE is also shown to be higher for the schema construction phase (base) on the schema application phase. These results support the hypothesis that rapid learning in humans can be explained as the rediscovery of a constructed schema.

The best fit between the model and human was obtained when *dat* was accelerated from the ACT-R default to twice the speed. This result may indicate room for improvement in modeling in this study or the limitations of the default production cycle settings in ACT-R.

Conclusion

This study aimed to explain human rapid motor learning using the motor learning module of ACT-R. The proposal of this study is that humans achieve rapid learning on a task by reconstructing a pre-learned motor schema during the task. To examine this hypothesis, we conducted simulations with schema construction and schema application phases based on the motor schema theory. The results showed that the schema application phase was more consistent with human learning performance than the schema construction phase.

The significance of this study is the proposal of modeling human learning using parameter optimization, which usually requires a long learning time. This method follows the motor learning transfer proposed in the previous study (Anderson et al., 2019) in terms of leveraging past related experience. We also consider that this method is extendable to end-to-end deep learning and reinforcement learning frameworks that have been the dominant artificial technology in recent years. These technologies usually take a vast amount of learning trials. Our method aims to reproduce a learning rate equivalent to that of humans by using prior learning. Such a method can be regarded as an application of the recent learning algorithm (Wang, Yao, Kwok, & Ni, 2020) from a small number of examples in large-scale models to the domain of cognitive models.

In addition, we would claim that the advantage of our model in requiring no external programs as was used in the models of driving (Salvucci, 2006) or the past line-following task (Morita et al., 2020). Therefore, the method of this study can be viewed as an attempt to create a unified theory of hu-

man cognition during perceptual-motor tasks.

This approach needs to be progressed further in the future. Although in the present study, the prior learning involved in the construction and application of the schema had a certain effect in terms of performance improvement, it was not sufficient in terms of achieving human performance. One factor that could be considered is the possibility that the schema construction procedures of this study did not find the optimal parameters, as shown in Table 2. It is possible that more optimal parameters could be found by taking a larger number of model runs over a longer period of time. In addition, the loop in the execution of the task in Figure 3 needs to be improved. Performance would be improved by skipping the component of task cycle.

References

- Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe*. New York: Oxford University Press.
- Anderson, J. R., Betts, S., Bothell, D., Hope, R., & Lebiere, C. (2019). Learning rapid and precise skills. *Psychological Review*, 126(5), 727-760.
- Byrne, M. D. (2000). The ACT-R/PM project. In *Simulating Human Agents: Papers from the 2000 AAAI Fall Symposium*. AAAI Press (pp. 1-3).
- Fitts, P. M. (1964). Perceptual-motor skill learning. In *Categories of human learning* (pp. 243-285). Elsevier.
- Gianferrara, P. G., Betts, S., & Anderson, J. R. (2021, 10). Cognitive & motor skill transfer across speeds: A video game study. *PLOS ONE*, 16(10), 1-31.
- Kieras, D. E., & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction*, 12(4), 391-438.
- Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.
- Kotseruba, I., & Tsotsos, J. K. (2020). 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*, 53(1), 17-94.
- Laird, J. E., Lebiere, C., & Rosenbloom, P. S. (2017). A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. *AI Magazine*, 38(4), 13-26.
- Morita, J., Miwa, K., Maehigashi, A., Terai, H., Kojima, K., & Ritter, F. E. (2020). Cognitive modeling of automation adaptation in a time critical task. *Frontiers in Psychology*, 2149.
- Nagashima, K., Nishikawa, J., Yoneda, R., Morita, J., & Terada, T. (2022). Modeling optimal arousal by integrating basic cognitive components. In *Proceedings of the 20th international conference on cognitive modeling* (pp. 196-202).
- Ritter, F., Tehranchi, F., & Oury, J. (2019). ACT-R: A cognitive architecture for modeling cognition. *Wiley Interdisci-*

- plinary Reviews: Cognitive Science*, 10(3).
- Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Human Factors*, 48(2), 362–380.
- Schmidt, R. A. (1975). A schema theory of discrete motor skill learning. *Psychological Review*, 82, 225–260.
- Suwa, M. (2008). A cognitive model of acquiring embodied expertise through meta-cognitive verbalization. *Information and Media Technologies*, 3(2), 399–408.
- Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3), 1–34. doi: 10.1145/3386252

The CoFI Reader: A Continuous Flow of Information Approach to Modeling Reading

Bruno Nicenboim (b.nicenboim@tilburguniversity.edu)

Department of Cognitive Science & Artificial Intelligence, Tilburg University

Abstract

I present a novel cognitive model of reading based on a continuous flow of information approach, the CoFI reader, where partial information from different levels of representation is continuously being made available to next levels. In an example application, I implement the model in a hierarchical Bayesian framework and fit it to self-paced reading times data: a reading task where one word is presented at a time and the presentation time is controlled by the experimental subject. The results show that the model provides a reasonable fit to word-level reading times, and can account for two previously observed findings: (i) reading times are much shorter than the minimum time required for all cognitive processes that should take place, and (ii) the processing difficulty of a word affects the reading times of subsequent words (i.e., spillover or lag effects). Computational models have explained these findings through parafoveal preview, that is, the partial processing of upcoming words during reading before they are directly fixated by the eyes. The CoFI reader model provides an explanation for these findings that is relevant for natural reading, but also, crucially, for self-paced reading, where parafoveal preview is not possible.

Keywords: continuous flow of information; Bayesian hierarchical modeling; reading times; self-paced reading

The classic perspective on human information processing posits that cognition is composed of a series of discrete, non-overlapping cognitive processes, a notion introduced by Donders (1868–1969) through his pioneering work on mental chronometry (a view modernized later by, among others, Sternberg, 1969). Despite the development of alternatives (e.g., McClelland, 1979), non-overlapping discrete processes (whether they are serial, parallel, or a combination of the two) have been the overwhelming and sometimes implicit choice in human information processing and cognition (Townsend & Wenger, 2021). This is also the case in the study of human sentence processing, where verbal and computational models of reading overwhelmingly assume discrete cognitive stages. In contrast to this “Dondersian” approach, I propose here the foundations of a computational model of reading, where the processing of different stages overlaps because information cascades from lower levels of representation (e.g., spatial information) to higher ones (e.g., semantics/syntax).

Reading times as a window to cognitive processes

Reading times are widely used to investigate human sentence processing in psycholinguistics, as they are meant to provide insight into cognitive processing difficulty. For example, consider the following sentences:

- (1) It’s raining, I’ll take the umbrella before I leave the house.
- (2) It’s raining, I’ll take the hat before I leave the house.

Say that we have some measure of the predictability of “umbrella” and “hat” as a continuation of “It’s raining, I’ll take the”, where “umbrella” is more predictable than “hat”. Predictability can be measured with a cloze task or using neural network models (as in Frank, 2013). Longer reading times at an unpredictable word (e.g., “hat”) in comparison with a more predictable one (e.g., “umbrella”) (controlling for the effect of length) has been taken as evidence for increased processing difficulty for readers (e.g., Ehrlich & Rayner, 1981).

There are two methods commonly used to measure reading times: Eye-tracking-while-reading and self-paced reading. In the eye-tracking-while-reading paradigm, eye movements are recorded while subjects read a text “normally” (usually using a head mounted device to prevent head movements). This method allows researchers to determine the duration of the first fixation on a word, the sum of fixations entering from the left, the sum of all fixations, and other metrics. As in everyday reading, short words are skipped and words in sentences are not necessarily read in sequential order. Furthermore, while readers fixate their gaze on a word, they obtain *some* information about the upcoming word(s) through parafoveal preview (how much information is obtained in this preview is contested, e.g., Vasilev & Angele, 2017). In contrast, in the self-paced reading paradigm, words are presented one by one, with the subject pressing a button to reveal each new word. The difference in times between button presses is used to determine reading times. This method, while admittedly less naturalistic, simplifies modeling, as eye movements are not relevant and there is no parafoveal preview. Crucially, however, all findings in psycholinguistics are consistent across the two methodologies.

There are two issues that complicate a straightforward interpretation of reading times of word n as an index of processing difficulty for the same word and suggest a looser coupling between processing and reading times than the one typically assumed in psycholinguistics (a phenomenon that has been recognized in cognitive psychology for some time: e.g., Bouma & De Voogd, 1974; Kliegl, Nuthmann, & Engbert, 2006): (i) The times spent at each word (whether these are reaction times in self-paced reading or fixation durations in eye-tracking methodology) are too short to be affected by all

the cognitive processes required for successful reading (Reichle & Reingold, 2013), *if* the assumed processes would occur serially and in a non-overlapping way. (ii) Spillover effects (Mitchell, 1984) or lag effects (Kliegl et al., 2006), that is the observation that characteristics of a given word affect subsequent words, are ubiquitous in both eye-tracking and self-paced reading paradigms.

(i) Reading times are very short considering all the cognitive processes involved. Computational models of reading have attempted to solve this paradox by assuming (i) that much of the processing occurs before a reader fixates on a word through parafoveal preview during the previous fixation, and (ii) that only a minimal amount of linguistic processing is required to initiate a saccade or press a button. For example, according to the E-Z Reader (Reichle, Warren, & McConnell, 2009) a signal to initiate a saccade is sent immediately after a familiarity check stage, which is unaffected by most linguistic processing. According to SWIFT (Engbert, Nuthmann, Richter, & Kliegl, 2005), lexical processing can inhibit a timer that triggers the saccades, but the model does not account for post-lexical high-level linguistic processing (there is, however, a recent attempt to integrate memory process to the model: Rabe, Paape, Vasishth, & Engbert, 2021).

(ii) Spillover/lag effects are ubiquitous. A widespread view in psycholinguistics is that spillover effects result because some aspects of processing join a buffer and are dealt with later (Mitchell, 1984). However, given that spillover effects happen at each word in reading, it is unclear then how much can be processed at each word. Computational models of reading such as SWIFT and E-Z reader explain the spillover/lag effects mostly through parafoveal preview: If a reader fixates on a challenging word n , they are likely to perceive “less” of word $n + 1$ with their parafoveal preview, and, thus, when they fixate on word $n + 1$ they would need to process more in comparison with a situation where word n is easier and allows for “more” parafoveal preview on word $n + 1$. However, SWIFT provides an additional explanation that does not depend on parafoveal preview: When the reader fixates on a difficult word, they inhibit the progress of the timer that triggers the saccades. Depending on whether a saccade program is already running, inhibition can either affect the saccade timer for word n , or for the saccade for the word $n + 1$. This means, however, that it cannot affect the saccade program of word $n + 2$ and later words.

A proposal for a computational framework to model (self-paced) reading data: CoFI reader

We introduce here the CoFI reader model, which can account for the seemingly fast reading times and explain spillover effects without the need of parafoveal preview. Whereas in principle it could be extended to “normal” reading, the focus of this specific model is to account for self-paced reading data. This means that the effect of parafoveal preview and the mechanism directing the target for the eye move-

ments are ignored. CoFI reader assumes that parallelism while reading is present to a large extent (similarly to Engbert et al., 2005; Reilly & Radach, 2006; Snell, van Leipsig, Grainger, & Meeter, 2018; Trukenbrod & Engbert, 2014), and that there is a signal to control the motor system (here to press a button and be presented with the next word). This signal is controlled by a stochastic timer and can be inhibited by processing load (similarly to Engbert et al., 2005; Trukenbrod & Engbert, 2014). The CoFI reader model differs from other reading models by assuming that information cascades from lower levels of representation (e.g., spatial information) to higher ones (e.g., syntactic/semantic information), and its implementation assumes a continuous flow of information from different processes (Ashby, 1982; Coles, Gratton, Bashore, Eriksen, & Donchin, 1985; McClelland, 1979; Townsend & Wenger, 2021; also compatible with a dynamic systems-approach, e.g., Spivey, 2008). This is depicted in Figure 1.

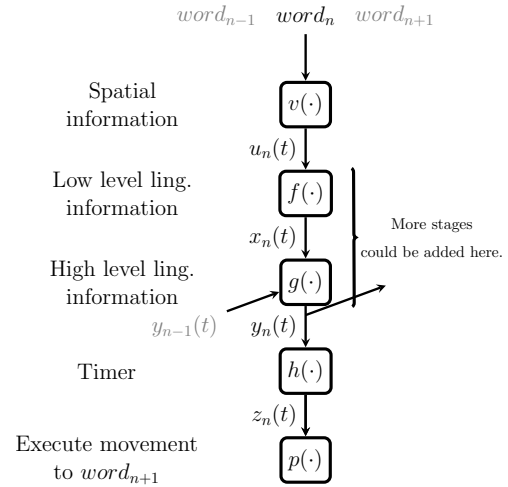


Figure 1: Schematic of the CoFI model.

The first three stages, v , f , g , process information which is represented with a real value between 0 and 1. The first stage involves the processing of visual information. To simplify the model, $v(t)$ is assumed to be a step function for word n , $u_n(t) = 1$ if $t > t_0$, with $t_0 = 50$ ms after the presentation of word n ; otherwise $u_n(t) = 0$. I assume that the visual information is accessible even if the word is no longer in the field of vision (as shown, among others, by Rayner, Inhoff, Morrison, Slowiaczek, & Bertera, 1981). As time elapses, the visual information of word n should become unavailable. As a simplifying assumption, the reader is assumed to have access to visual information of words up to $n - 3$ when viewing word n . The spatial information feeds into the low level linguistic information processing stage:

$$f(u, t, r) = u_n(t) \cdot \text{sigmoid}((t - t_{n,0}) \cdot r_{n,f} \cdot u_n(t)) = x_n(t) \quad (1)$$

The sigmoid function starts by yielding an output of zero and, as it acquires information from the preceding stage, its

output gradually grows in time until it reaches a plateau (of output one). The sigmoid function is $1 - e^{-(x/\lambda)^k}$ with $\lambda = 5$ and $k = 2$. The growing rate of the information is determined by r . The output of the entire function is zero when u is zero, that is before $t_{n,0}$ ($t_{n,0} : t_0$ after word n is presented).

The high-level linguistic information stage is similar to the previous one with the difference that it depends not only on the output of the previous stage, $x_n(t)$, but also on the output of the previous word analogous stage $y_{n-1}(t)$. The intuition behind this is that to process high-level linguistic features such as syntax, the reader needs to have (some extent of) knowledge about the current word and also the context.

$$g(x, t, r) = x_n(t) \cdot \text{sigmoid}((t - t_{n,0}) \cdot r_{n,g} \cdot x_n(t) \cdot y_{n-1}(t)) = y_n(t) \quad (2)$$

As a first implementation, the current model has only two stages for linguistic processing. The last processing stage output feeds a timer:

$$h(t) = y_n(t) \cdot (t - t_{n,0}) \cdot r_h = z_n(t) \quad (3)$$

When the output of the timer reaches a threshold $c_n(t)$, a signal is sent to the motor system. This threshold is not a function of the characteristics of the current word, and it is assumed to be affected by the difficulty of the task, and characteristics of the reader such as motivation, tiredness, etc (as in other accumulator models). Finally, t_{press} time after the signal is sent, $p(t)$ is set to one meaning that a button is pressed and word $n + 1$ is displayed. The output of the different functions are shown in Figure 2 for a single word. Figure 3 shows the way in which the model accounts for spillover effects.

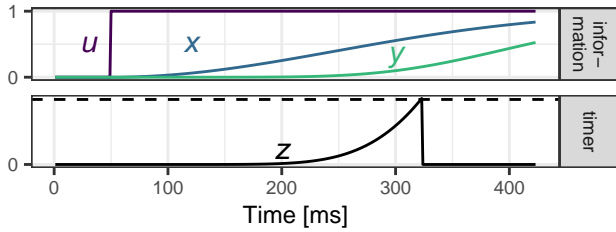


Figure 2: The upper panel shows the accumulation of information of the different stages for a single word. The lower panel shows the timer reaching its threshold triggering the motor system. Notice that while the figure ends with the pressing of a button, the accumulation of information for this word would continue.

Example application: Self-paced reading data of Frank, Fernandez Monsalve, Thompson, & Vigliocco (2013)

In the following example application, I focus on the effect of word frequency as a representative characteristic of a word that might affect a low-level linguistic processing stage, and surprisal (to what extent a word is unpredictable given a context) as a characteristic of a word and context that might affect

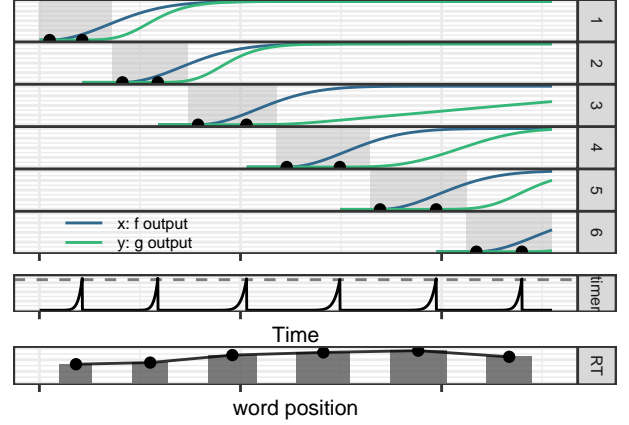


Figure 3: A toy simulation depicting the processes involved in understanding a six-word sentence, where each word shares similar features except for the word in position 3. The gray boxes illustrate the time period when each word is displayed. For each word, the first black dot indicates that the visual information of the given word is available, the second black dot indicates the moment when the timer reaches the threshold. In this toy example, the complexity of *word*₃ results in a lower $r_{3,g}$. As for all the words the g stage is barely processed when the timer is triggered (i.e., y is low), the change in g_3 is most relevant for the timers of *word*₄ and *word*₅, where the effect of g_4 and g_5 (influenced by g_3) is more pronounced.

a higher-level linguistic processing stage. In addition, I exemplify how the output of a model could determine the value of the parameters that control the growing rate of the sigmoid functions, r , by using the output of a log-normal regression.

Data The CoFI reader model was fit to a small subset of data from the Frank et al. (2013) study, which collected word self-paced reading times from independent (and disconnected) sentences taken from amateur novels written in British English spelling. The data were collected from 114 subjects, but non-native English speakers and those with low comprehension scores were excluded. Because the current implementation of the model cannot handle outliers, only 6 subjects whose first 100 sentences contained reading times between 200 and 3000 ms were used for the analysis.

Modeling The CoFI model has a tractable likelihood and is implemented as a Bayesian hierarchical model. For each subject i and word n , the rate of accumulation of information in the low- and high-level linguistic information processing stages is modeled as a function of r with a hierarchical structure by subject, where r is sampled from a log-normal model:

$$\begin{aligned} r_{n,f} &\sim \text{LogNormal}(\alpha + \text{freq}_n \cdot (\beta_f + \nu_{\beta,f}), \sigma_r) \\ r_{n,g} &\sim \text{LogNormal}(\alpha + \text{surp}_n \cdot (\beta_g + \nu_{\beta,g}), \sigma_r) \end{aligned} \quad (4)$$

freq_n represents the scaled Zipf value of the frequency of word n derived using SUBTLEX-UK (Van Heuven, Mandera, Keuleers, & Brysbaert, 2014), surp_n is a scaled measure of

Table 1: Priors for the parameters. Distributions followed by $_{+}$ indicate support for only positive values. Distributions followed by $_{T[0, \min(RT)_i]}$ indicate support between 0 and the minimum reading time for each subject (in seconds).

Parameter	Prior
α	$Normal(3, 0.5)$
$\beta_{f,g}$	$Normal(0, 0.1)$
σ_r	$Normal_{+}(3, 2)$
α_c	$Normal(0, 1)$
β_c	$Normal(0, 1)$
σ_c	$Normal_{+}(0.5, 0.5)$
lc_1	$Normal(-2, 1)$
σ	$Normal_{+}(3, 2)$
$t_{press, 1..6}$	$Normal_{T[0, \min(RT)_i]}(0.1, 0.05)$

surprisal, formally $-\log P(w_n | w_1 \dots w_{n-1})$ derived using a recurrent neural network in Frank (2013). The parameters α , β_f , and β_g represent, respectively, a baseline rate for both stages when a word is present, the effect of frequency, and the effect of surprisal. The parameters v_β are the by-subject adjustments.

The rate of the timer is stochastic with a fixed mean following a log-normal distribution as shown in Eq. (5).

$$r_h \sim \text{LogNormal}(0, \sigma) \quad (5)$$

The threshold of the timer, c , follows a hierarchical log-normal first order autoregressive model, AR(1):

$$c_n \sim \text{LogNormal}(\alpha_c + v_{\alpha,c} + \log(c_{n-1}) \cdot (\beta_c + v_{\beta,c}), \sigma_c) \quad (6)$$

where $n \neq 1$, and $c_1 = e^{lc_1 + v_{lc_1}}$. This means that changes in the threshold are much slower than changes due to processing of information (as in the model of visual search ICAT, Trukenbrod & Engbert, 2014).

All the by-subject adjustments to the parameters are drawn from uncorrelated normal distributions as shown in Eq. (7).

$$v \sim \mathcal{N}(\mathbf{0}, \Sigma); \rho = 0 \quad (7)$$

Finally, the minimum time needed for pressing a button is estimated individually for each subject as $t_{press,i}$. Priors for all the parameters can be found in Table 1.

Next, I derive the observational model to obtain the likelihood of the model. When the threshold is reached, the following holds:

$$h(t) = y_n(t) \cdot (t - t_{n,0}) \cdot r_h = z_n(t) = c_n \quad (8)$$

$$r_h = \frac{c_n}{y_n(t) \cdot (t - t_{n,0})}$$

A draw of t is accepted if r_h is such that solving the equation yields a root in a narrow interval (smaller than ϵ) around

the observed t^* . This, in turn, means that r_h falls in a narrow interval around r_h^* (the value of r_h that implies that $t = t^*$):

$$|t - t^*| < \epsilon \iff |r_h - r_h^*| < \left| \frac{\partial r_h}{\partial t} \right| \epsilon \quad (9)$$

$$P\left(|r_h - r_h^*| < \left| \frac{\partial r_h}{\partial t} \right| \epsilon\right) \approx 2\epsilon \cdot p(r_h^*)$$

where $p(r_h)$ is the probability density function for the prior distribution of r_h and ϵ is a constant (which will not matter for the posterior distribution). Then,

$$\mathcal{L}(t | \Theta, n) \approx \left| \frac{c_n \frac{\partial}{\partial t} \frac{1}{y_n(t) \cdot (t - t_{n,0})}}{\partial t} \right| \cdot p(r_h) \quad (10)$$

where Θ represents all the parameters of the model, and $p(r_h)$ is the PDF of $\text{LogNormal}(0, \sigma)$.

Results I fit the models using Stan with the `cmdstanr` package (Gabry, Češnovar, & Johnson, 2022; Stan Development Team, 2023) in R (R Core Team, 2022) and verified that the model converged. Figure 4 shows the successful recovery of the model parameters using a non-hierarchical version of the current model.

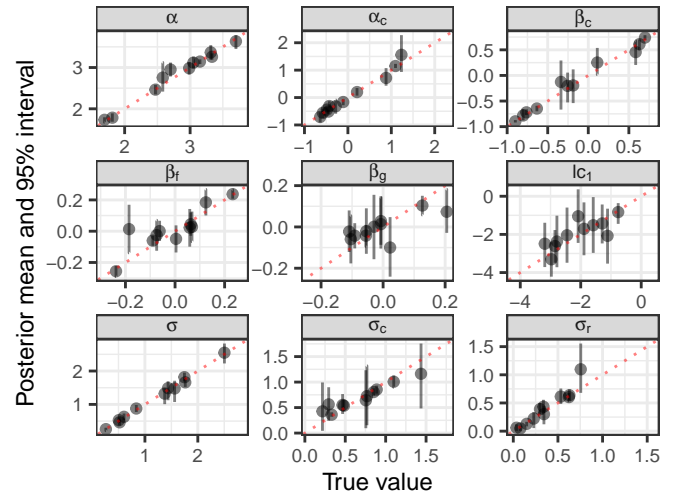


Figure 4: Parameter recovery based on a non-hierarchical version of the model through 20 simulations; only the 10 models that converged were kept. The x-axis shows the true values, which were sampled from priors, while the y-axis shows the posterior mean and 95% credible intervals.

The posterior distributions of the parameters are summarized in Figure 5. Given that the population-level parameter β_c and the by-subject estimates ($\beta_{c,i} = \beta_c + v_{c,i}$) are below 1, it is possible to conclude that the threshold is weakly stationary. The parameter that control the growth of f , β_f (i.e., the low level-linguistic stage), is inconclusive regarding the effect of frequency. This is probably because frequency is highly (anti) correlated with surprisal ($\rho = -0.74$). In contrast, the parameter that controls the growth of g , β_g (the high level-linguistic stage), has a negative sign, indicating (as expected)

that a more surprising word leads to a slower accumulation of information at this stage.

Descriptive adequacy of the model Posterior predictive checking is employed to assess the extent to which the model accurately describes the observed data. This involves verifying that the observed data appears plausible when compared to simulated data from the posterior predictive distribution. The simulated datasets are generated by drawing 50 samples from the posterior and then (i) solving for t using the bisection method $c_n - h(t) = 0$; (ii) given that t is the time from the presentation of word 1 up to the moment when the timer n reaches the threshold (and before a button is pressed, which takes t_{press}), it follows that $RT_n = t - \sum_{j=1}^{n-1} t_j + t_{press}$.

Figure 6 reproduces the general trends of reading times at the level of individual words by subject. It is evident that the model overestimates reading times in certain regions. This could be because the model fails to account for the inter-subject variability in the presence of long reading times. Despite the exclusion of extreme outliers in the data, the presence of infrequent long reading times may disproportionately influence the model's predictions. Figure 7 shows that the simulated dataset partially replicates the pattern of surprisal effects and the spillovers of surprisal. Specifically, the simulated effects for surprisal and the second spillover of surprisal align well with the observed effects, but the model underestimates the first spillover.

Conclusion and future directions

I introduced the CoFI reader model, a novel framework that models reading as a continuous flow of information allowing information to cascade from lower to higher levels of representation, with the partial output of each processing stage being available to the next one. The model uses a stochastic timer that depends on the processing of the current displayed word and previous words to predict reading times.

By fitting the model to self-paced reading data using a hierarchical Bayesian method and using word frequency and surprisal values as predictors, I found that the model provides a reasonable fit to word-level reading time trends. Furthermore this framework provides a formal account of the loose coupling between cognitive difficulty and reading times and provides a novel explanation to two previously observed findings without the need of parafoveal preview (which is unavailable in self-paced reading): (i) that reading times are much shorter than the minimum time required for all the cognitive processes that should take place, and (ii) the presence of spillover or lag effects, that is that processing difficulty of a word affects the reading times of subsequent words.

The proposed framework offers a basis for further investigation by extending the model: (i) One clear limitation of the current model is that it cannot handle outliers, this could be solved by incorporating a mixture process that takes into account lapses of attention. (ii) In the current model, the rate of information accumulation was assumed to be the output of a log-normal regression. It would be possible, however, to embed

more complex models to investigate other phenomena from sentence processing. (iii) The model could be extended by adding other processing stages such as discourse or reanalysis stages. (iv) The model could readily incorporate how individual differences influence various aspects of reading. This could be achieved by integrating measures of individual differences, which might impact different processing stages or the threshold. (v) While the current version of the model only handles self-paced reading data, it would be possible to extend the model by including insights from models of oculomotor control to fit natural reading data.

Acknowledgements

I am grateful to Pavel Logačev for insightful discussions on modeling reading data. Special thanks go to Giacomo Spigler for his assistance in developing the model. His help with the equations underpinning the model has been invaluable. I am also very grateful to Niko Huurre and Michael Betancourt, who were extremely generous with their time on the Stan Forums <https://discourse.mc-stan.org/>.

References

- Ashby, F. G. (1982). Deriving exact predictions from the cascade model. *Psychological Review*, 89, 599–607. <http://doi.org/10.1037/0033-295X.89.5.599>
- Bouma, H., & De Voogd, A. (1974). On the control of eye saccades in reading. *Vision Research*, 14(4), 273–284. [http://doi.org/10.1016/0042-6989\(74\)90077-7](http://doi.org/10.1016/0042-6989(74)90077-7)
- Coles, M. G. H., Gratton, G., Bashore, T. R., Eriksen, C. W., & Donchin, E. (1985). A psychophysiological investigation of the continuous flow model of human information processing. *Journal of Experimental Psychology: Human Perception and Performance*, 11, 529–553. <http://doi.org/10.1037/0096-1523.11.5.529>
- Donders, F. C. (1868–1969). On the speed of mental processes. *Acta Psychologica*, 30, 412–431. [http://doi.org/10.1016/0001-6918\(69\)90065-1](http://doi.org/10.1016/0001-6918(69)90065-1)
- Ehrlich, S. F., & Rayner, K. (1981). Contextual effects on word perception and eye movements during reading. *Journal of Verbal Learning and Verbal Behavior*, 20(6), 641–655. [http://doi.org/10.1016/S0022-5371\(81\)90220-6](http://doi.org/10.1016/S0022-5371(81)90220-6)
- Engbert, R., Nuthmann, A., Richter, E. M., & Kliegl, R. (2005). SWIFT: A Dynamical Model of Saccade Generation During Reading. *Psychological Review*, 112(4), 777–813. <http://doi.org/10.1037/0033-295X.112.4.777>
- Frank, S. L. (2013). Uncertainty Reduction as a Measure of Cognitive Load in Sentence Comprehension. *Topics in Cognitive Science*, 5(3), 475–494. <http://doi.org/10.1111/tops.12025>
- Frank, S. L., Fernandez Monsalve, I., Thompson, R. L., & Vigliocco, G. (2013). Reading time data for evaluating broad-coverage models of English sentence processing. *Behavior Research Methods*, 45(4), 1182–1190. <http://doi.org/10.3758/s13428-012-0313-y>
- Gabry, J., Češnovar, R., & Johnson, A. (2022). *Cmdstanr: R interface to 'CmdStan'*.

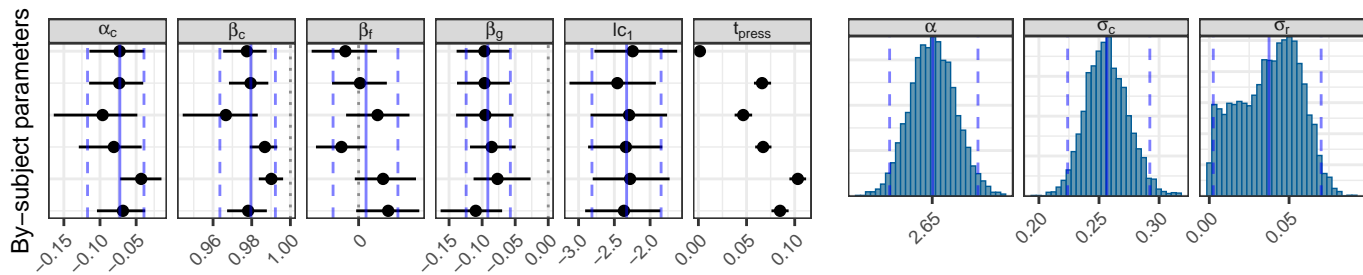


Figure 5: Mean and 95% CI as blue vertical lines for the population-level parameter and as dots and horizontal lines for the by-subject adjusted parameters. $t_{press,i}$ are by-subject parameters with no population-level counterpart, and that α , σ_c and σ_r have no by-subject adjusted counterparts.

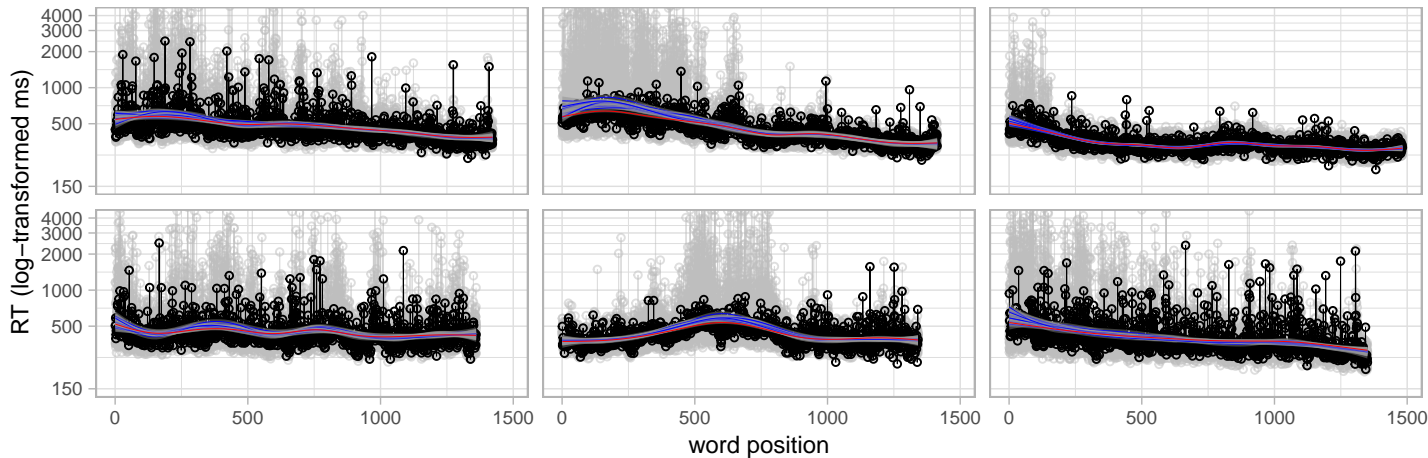


Figure 6: Each panel shows the ordered reading times for each subject. Observed data is shown in black with smoothed conditional means in red, data from 14 datasets generated from the posterior predictive distributions are shown in grey with each smoothed conditional means curve in blue.

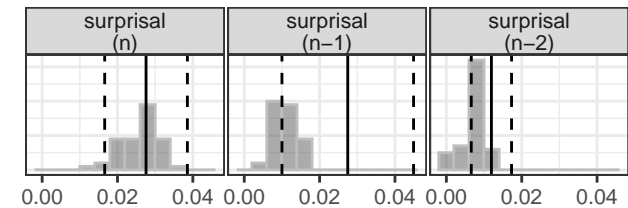


Figure 7: This figure displays the distribution of 50 simulated effects of surprisal from the current word, the previous word, and two words prior, at the current word position. That is the effect of surprisal and two spillovers. The same effects (with their standard errors) estimated from the observed data with their standard error are represented by black lines. Estimation was done with a linear mixed model fitted to the log-transformed reading times.

Kliegl, R., Nuthmann, A., & Engbert, R. (2006). Tracking the mind during reading: The influence of past, present, and future words on fixation durations. *Journal of Experimental Psychology: General*, 135(1), 12. <http://doi.org/10.1037/0096-3445.135.1.12>

McClelland, J. (1979). On the time relations of men-

tal processes: An examination of systems of processes in cascade. *Psychological Review*, 86, 287–330. <http://doi.org/10.1037/0033-295X.86.4.287>

Mitchell, D. C. (1984). An evaluation of subject-paced reading tasks and other methods for investigating immediate processes in reading. *New Methods in Reading Comprehension Research*, 69–89.

R Core Team. (2022). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from <https://www.R-project.org/>

Rabe, M., Paape, D., Vasishth, S., & Engbert, R. (2021). Dynamical cognitive modeling of syntactic processing and eye movement control in reading. *PsyArXiv*. <http://doi.org/10.31234/osf.io/w89zt>

Rayner, K., Inhoff, A. W., Morrison, R. E., Slowiaczek, M. L., & Bertera, J. H. (1981). Masking of foveal and parafoveal vision during eye fixations in reading. *Journal of Experimental Psychology: Human Perception and Performance*, 7(1), 167. <http://doi.org/10.1037/0096-1523.7.1.167>

Reichle, E. D., & Reingold, E. (2013). Neurophysiological

- constraints on the eye-mind link. *Frontiers in Human Neuroscience*, 7. <http://doi.org/10.3389/fnhum.2013.00361>
- Reichle, E. D., Warren, T., & McConnell, K. (2009). Using E-Z Reader to model the effects of higher level language processing on eye movements during reading. *Psychonomic Bulletin & Review*, 16(1), 1–21. <http://doi.org/10.3758/PBR.16.1.1>
- Reilly, R. G., & Radach, R. (2006). Some empirical tests of an interactive activation model of eye movement control in reading. *Cognitive Systems Research*, 7(1), 34–55. <http://doi.org/10.1016/j.cogsys.2005.07.006>
- Snell, J., van Leipsig, S., Grainger, J., & Meeter, M. (2018). OBI-reader: A model of word recognition and eye movements in text reading. *Psychological Review*, 125(6), 969–984. <http://doi.org/10.1037/rev0000119>
- Spivey, M. (2008). *The continuity of mind*. Oxford University Press.
- Stan Development Team. (2023). *Stan modeling language users guide and reference manual, version 2.31*.
- Sternberg, S. (1969). The discovery of processing stages: Extensions of Donders' method. *Acta Psychologica*, 30, 276–315. [http://doi.org/10.1016/0001-6918\(69\)90055-9](http://doi.org/10.1016/0001-6918(69)90055-9)
- Townsend, J. T., & Wenger, M. J. (2021). A beginning quantitative taxonomy of cognitive activation systems and application to continuous flow processes. *Attention, Perception, & Psychophysics*, 83(2), 748–762. <http://doi.org/10.3758/s13414-020-02180-2>
- Trukenbrod, H. A., & Engbert, R. (2014). ICAT: A computational model for the adaptive control of fixation durations. *Psychonomic Bulletin & Review*, 21(4), 907–934. <http://doi.org/10.3758/s13423-013-0575-0>
- Van Heuven, W. J., Mandera, P., Keuleers, E., & Brysbaert, M. (2014). SUBTLEX-UK: A new and improved word frequency database for British English. *Quarterly Journal of Experimental Psychology*, 67(6), 1176–1190. <http://doi.org/10.1080/17470218.2013.850521>
- Vasilev, M. R., & Angele, B. (2017). Parafoveal preview effects from word N+1 and word N+2 during reading: A critical review and Bayesian meta-analysis. *Psychonomic Bulletin & Review*, 24, 666–689. <http://doi.org/10.3758/s13423-016-1147-x>

Improving Visuomotor Control of a Cognitive Architecture

Grace Roessling (pro.graceroessling@gmail.com)

Cognitive Science Department, Rensselaer Polytechnic Institute, Troy, NY, USA

Tim Halverson (th Alverson@aptima.com)

Aptima, Inc., Fairborn, OH, USA

Christopher Myers (christopher.myers.29@us.af.mil)

Air Force Research Laboratory, Wright-Patterson AFB, OH, USA

Abstract

Symbolic/hybrid computational cognitive architectures, including the ACT-R framework, are adept at capturing a wide variety of human cognitive processes and behaviors including problem-solving, memory, and language. However, such cognitive architectures do not capture visuomotor behaviors that tightly couple perceptual and motor processes – such as manual tracking. In this study, we aimed to improve the cognitive fidelity of manual tracking behavior within the ACT-R framework by implementing the position control model (PCM) – a continuous, linear control model that effectively captures human tracking behavior (Powers, 1978). We integrated PCM within a MATB task model developed within the ACT-R framework, to examine if the integrated ACT-R/PCM model showed improvement in capturing human tracking performance relative to the Standard ACT-R model. Results indicate that the ACT-R/PCM Integrated model showed improved performance in capturing certain aspects of human tracking behavior, in comparison to the Standard ACT-R model.

Keywords: ACT-R; manual tracking; visuomotor behavior; Fitt's law; perceptual control theory; linear control model

Introduction

Few symbolic/hybrid computational cognitive theories expressed as architectures incorporate the tight coupling of perceptual and motor systems. Years of research in visual-motor control have demonstrated that their relationship can be mathematically described. We seek to improve visual-motor control within a commonly used cognitive architecture, ACT-R, to improve tracking, visual and motor pursuit of a target, etc. In the following sections, we will identify the key components to this project: the task, the problem, and the solution.

Task

We are interested in improving the cognitive fidelity of symbolic computational cognitive architectures, specifically in the context of human tracking behavior. To this end, we start with ACT-R. To successfully track a target, one must perceive the spatial discrepancy between the tracking item (i.e., the *cursor*) and the item to be tracked (i.e., the *target*), and produce control adjustments that minimize the distance between the cursor and the target through manual inputs to an end-effector (e.g., joystick, mouse; Powers, 1973). Compensatory tracking requires the subject to control an end-effector to maintain alignment between a cursor and a target as the cursor is randomly displaced over time (Poulton, 1952a; Cherkinoff et al., 1955).

As human tracking behavior is exhibited across many everyday tasks that humans complete (e.g., maintaining lane

position while driving down a road), there is a long tradition of research dedicated to computationally modeling human tracking behavior (Craik, 1948; McRuer & Jex, 1967; Navas & Stark, 1968; Powers, 1973). Compensatory tracking requires motor behaviors that rapidly respond to visual information to reduce control error (i.e., the distance between current cursor position and desired cursor position), thus necessitating a tight coupling between perception and action.

Problem

Theoretical frameworks that are focused on developing high cognitive fidelity models of visuomotor behavior, specifically ecological psychology and perceptual control theory, demonstrate that there is a tight coupling between human perception and motor capacities (Gibson, 1986; Powers, 1978; Warren, 2006). However, symbolic computational cognitive architectures, including ACT-R, are limited in their ability to represent tight couplings between perceptual and motor processes due to structural constraints. Currently, the ACT-R architecture does not facilitate communication directly between perceptual and manual modules—rather, information exchange is mediated via the production system. Thus, ACT-R does not use direct communication between motor and perceptual cognitive processes to execute tracking behavior. Instead, ACT-R requires sharing information over a procedural bottleneck (i.e., production system).

One can examine and implement the control solutions developed by the ecological psychology framework or perceptual control theory framework to implement the tight coupling between visual and motor control. Such an implementation will result in (1) a more accurate representation of how cognitive processes interact to produce visuomotor behavior, and (2) the execution of continuous, smooth motor control that resembles human motor behavior.

Solution

To improve the cognitive fidelity of human tracking behavior within the ACT-R architecture, we implemented the *position control model* – a linear control model that effectively captures human tracking behavior (Powers, 1973). The position control model uses a simple and effective control heuristic that minimizes the distance between the cursor and the fixed target location (i.e., center of the screen). We integrated the position control model within the ACT-R framework and (1) compared the model to human tracking data, and (2) investigated if the combination of both models improved ACT-R's tracking performance.

Background

ACT-R Cognitive Architecture

ACT-R is an integrated theory of human cognition and action (Anderson, 2009). The ACT-R cognitive architecture is a computational instantiation of that theory. Most critical for the current research is that ACT-R includes mechanisms for visual perceptual and manual motor processes. Relevant existing ACT-R capacities are introduced and covered in the following subsections.

Visual System The visual theory instantiated in ACT-R is a straightforward interpretation of Feature Integration Theory (Treisman & Gelade, 1980). Most central to this work is that attending to visual stimuli takes time, on the order of 135 ms to find and integrate visual stimuli into the visual buffer. This integration of visual information can be performed in parallel with motor movements.

Motor System The manual motor theory is closely related to that instantiated in the EPIC cognitive architecture (Meyer & Kieras, 1997). Most central to this work is that manual motor movements are characterized by Fitts' law (Fitts, 1954), predicting the timing of ballistic pointing movements based on the distance from start to end point and target width. End points may be retrieved from memory or guided by information in the visual buffer.

Production System All of the cognitive, perceptual, and motor processes represented in ACT-R are coordinated through the production system, acting as a serial bottleneck. The production system matches conditions specified in procedural memory, represented as if-then statements of task knowledge and strategy, to the state of the perceived environment that exists in the various buffers (e.g., visual and declarative) to produce action, such as locations for pointing behavior. All interactions between the visual and motor processes are mediated by this central cognitive production system.

Tracking in Cognitive Architectures

A variety of methods have been used to model tracking tasks in cognitive architectures, including Fitts' law related methods for tracking tasks and control theory related methods for driving. By far the most common method for modeling tracking in cognitive architectures, Fitts' law related methods tend to use ballistic mouse-like cursor plying as surrogates for joystick deflection (e.g., Ballas et al., 1999; Balint, Reynolds, Blaha, & Halverson, 2017). These models generally use one or more productions to find discrepancies between target and cursor, decide to move the joystick, and execute a motor movement with *timing* described by Fitts' law.

While restricted to models of driving behavior, control theory-like methods have been used in cognitive architectures (Salvucci, 2006) and cognitive constraint model

frameworks (Brumby, Salvucci, & Howes, 2009). These models adjust steering movements with the motor movement *extents* described by control law-like equations.

A relatively new method for modeling tracking in computational cognitive models is the discrete movement model for cursory tracking that uses Fitts' law to determine the time course of non-ballistic, incremental movements in tracking (Zhang & Hornof, 2012). This method of modeling is perhaps most similar to the method presented here, although the theory and implementation are substantially different. Like the control law models, the movements are incremental instead of ballistic. Unlike the control law models, the details of the model lie in the timing rather than the extent of movements.

ACT-R's structure hinders direct communication between perceptual and motor modules – limiting its ability to capture visuomotor tasks that exhibit a tight coupling between perception and action including human tracking behavior. How can we further develop the ACT-R framework to better capture the rapid and adaptive behavior of humans during compensatory tracking tasks? To start, we can examine the sensorimotor literature that is dedicated to developing theories and models of human tracking behavior.

Visual-Motor Control Model

One of the predominant theoretical frameworks that aim to capture human manual tracking behavior is perceptual control theory (PCT). PCT provides control heuristics that couple visual information to motor behavior – producing smooth and continuous motor behavior (Powers 1973). PCT applies concepts and methods from negative feedback control, and proposes that biological control systems nullify effects of unpredictable internal/external perturbations by controlling perceptual input (rather than motor output). More specifically, biological control systems modify their behavior to receive perceptual input that matches referent goal states. For example, a goal state would be to move the joystick in such a way that the cursor overlaps with the target location. PCT models are simple and effective control heuristics formulated as linear control models that reduce control error over time.

Formulation of Linear Control Models A general formulation of linear control models is expressed as

$$\dot{C}(t) = -K \cdot P(t - \tau_d) \quad (1)$$

specifying the rate of change of a motor command executed by a biological control system (Markkula et al., 2017). P specifies control error – a function that determines the difference between the goal state of a perceptual input and the current state of a perceptual input. The term $P(t - \tau_d)$ represents the control error given perceptual, control decision, and motor delays. The gain parameter, K , scales and inverts the sign of control error to specify the rate of change of a given control adjustment.

In the context of manual tracking within the PCT framework, control error or P is considered to be the distance in pixels between the cursor location and target location. \dot{C} specifies the rate of change in cursor position to reduce control error P to zero. The control adjustment determines a new visual location for the cursor that brings it closer to the target location. Gain K and delay τ_d are parameters that are specific to individuals, and studies show that these parameter values are internally consistent (Parker et al., 2017).

Applications Linear control models, such as PCT, are effective at capturing human motor control behavior, particularly manual tracking behavior. There are numerous studies that have rigorously analyzed and validated the PCT architecture with human tracking data (see review by Parker et al., 2020). Previous studies show that PCT models are able to act against disturbances, capture individual specificity, and simulate complex tracking behavior (Bourbon & Powers, 1999; Marken & Powers, 1989; Parker et al., 2017).

Approach

Task Environment

We gathered data to validate the model using the NASA MATB-II (from now on, just “MATB”) tracking task (Santiago-Espada, Myer, Latorella, & Comstock, 2011). The MATB can be configured to include four subtasks: tracking, system monitoring, communications, and resource management. We configured the MATB to include tracking only for this validation data.

The MATB tracking task is a slightly unique variation of tracking tasks. The goal of the task is to keep the center of a cursor with a 25 pixel radius within a 75 pixel square centered within a 300 pixel square tracking window. The location of the cursor is updated every 100 ms to include a random perturbation and input from the joystick. The direction of the random perturbation is selected pseudo-randomly from eight directions, the four cardinal and four intercardinal directions. The extent of the perturbation is 3 pixels in each dimension of the selected direction; the intercardinal directions would change both the x and y location, whereas the cardinal directions would change either the x or y location. The extent is configurable from 1 to 3 pixels, but set to 3 for this validation.

The input from the joystick is discretized in much the same way as the random perturbation. The direction of joystick deflection is grouped into one of the same eight directions. The extent of the input is scaled to an integer value with a maximum determined by the joystick sensitivity configuration. For this validation, we varied the sensitivity between sessions for each participant to be either low (6 pixel max) or high (18 pixel max). The location of the center of the cursor was always limited to the edges of the tracking window.

Human Tracking Data

Data were collected for validation purposes only. The tracking data exported from MATB are RMSE from the center of the tracking window in one second bins, the narrowest bin boundary allowed by MATB. Data were collected from two conditions: low and high joystick sensitivity (described above).

Three collaborators very familiar with the MATB performed the tracking task for eleven minutes in each condition, always performing the low sensitivity condition first. Data were collected with a Logitech Extreme 3D Pro joystick and 27” IPS LED monitor running at 1080p (60Hz) attached to a Windows 10 system.

Position Control Model

The position control model we implemented is adapted from PCT (Powers, 1973; Powers, 2008). The functions in the PCT architecture are the input function, comparator function, and output function. These functions contain parameter values that can be specific to individuals: delay, gain, damping constant, and reference value. To start, the model uses the input function to detect the distance between the target position C_T and the actual cursor position C :

$$D(t) = C(t) - C_T(t) \quad (2)$$

where D specifies control error, which is then applied to the comparator function:

$$P(t) = R - D(t) \quad (3)$$

where $P(t)$ is the error signal that compares the perceptual signal D to the reference signal R , the desired position of the cursor. If R is equal to zero (i.e., if the desired position of the cursor is at the target location), then D and P are equivalent. However, if R is a nonzero value (i.e., the desired position of the cursor is not exactly on the target location due to bias or intentional imprecision of the subject), then D and P are not equivalent.

Lastly, the output function specifies the joystick position, which determines cursor position:

$$\dot{C}(t) = K_o \cdot P(t - \tau_d) - K_d \cdot C(t - 1) \cdot \Delta t \quad (4)$$

$$C(t) = C(t - 1) + \dot{C}(t) \quad (5)$$

In Equation 4, gain K_o is multiplied to control error P given a perceptual delay of τ_d to calculate the cursor translation. The damping constant K_d is multiplied to the previous output $C(t-1)$ to reduce the effects of the previous output in the calculation of the current output. t is the time increment between each control adjustment. Lastly, Equation 5

Table 1: Low and High Sensitivity Mean and SD of RMSE

	Low Sensitivity Mean (Error %), SD (Error %)	High Sensitivity Mean (Error %), SD (Error %)
Human	27.3, 12.84	31.7, 15.5
PCM	26(4.9), 13.2(3.4)	33.2(4.9), 16.9(8.9)
ACT-R/PCM	28(2.7), 11.7(9)	35.5(12), 13.7(11.6)
Std ACT-R	27.1(0.7), 12.2(5)	27.7(12.5), 15.7(1.4)

expresses the output function C , which is the summation of the previous cursor position $C(t-1)$ and the rate of change of the cursor position \dot{C} .

In summary, PCM is a one-dimensional, continuous control model that contains four mutable parameters: reference value R , damping constant K_d , delay τ_d , and gain K_o . Note that, in our implementation, we extended the model to two-dimensions (controlling both the x and y axis of the cursor). We tuned the parameters for these values with guidance from Parker et al. (2017), with the aim of reproducing similar mean RMSE to that of human subjects completing the tracking task.

Model Implementation

Python Implementation of PCM To test and verify our implementation of the position control model before integrating it with ACT-R, we wrote a program in Python that simulates the model as it completes the manual tracking task that human subjects completed. Analogous to the MATLAB tracking task, the simulation randomly perturbs the cursor location over time. After the cursor is displaced, the position control model is executed to determine the new location of the cursor. The PCM model was validated with human tracking data and analytically compared to other tracking models -- specifically, the Standard ACT-R model and Integrated ACT-R/PCM model.

ACT-R/PCM Integration Once the position control model was tested and verified, we then integrated it within a MATLAB task model developed within the ACT-R framework (version 7.14). As previously discussed, the ACT-R framework does not facilitate communication directly between perceptual and motor modules. To couple the manual and visual modalities within the model, we developed a new Tracking module that receives the visual location chunks of the cursor and target and then executes the manual movements based on underlying position control model, thus bypassing the production system bottleneck.

The Tracking module implements three new motor commands: “set-target”, “set-cursor”, and “track”. The first

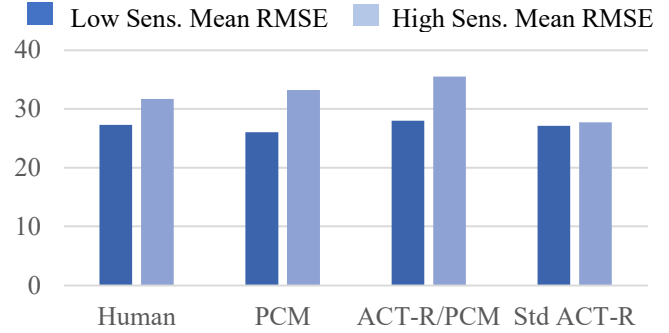


Figure 1: Bar graph comparing Mean RMSE across models.

two take the visual locations of the respective objects as arguments. The “track” command can either use previously specified target and cursor locations or provide new visual locations as arguments. When this command is executed, the position control model specified earlier determines the joystick.

The ACT-R/PCM model contains four productions. First, the model finds the target (center of the tracking area). Second, the model attends the target and sets the tracking target. Third, the model finds the cursor (tracking reticle). Finally, the model attends the cursor and starts tracking the cursor.

Differences between Implementations When applying the position control model to the ACT-R framework, we made a couple of alterations to the PCM model for the sake of compatibility. A difference between assumptions in PCM and ACT-R is that the output function in PCM directly specifies the new visual location of the cursor over time. On the other hand, the ACT-R architecture interacts with (simulated) computer input devices (e.g., a joystick) which have their own dynamics (e.g., gain) separate from the human controller. Thus, in our ACT-R/PCM integration, we modified the output function in PCM to accommodate feeding its values into a polar coordinate system that determines the joystick deflection, which then influences the new velocity of the cursor within the simulated device.

Standard ACT-R Model An ACT-R model using the original joystick manual motor command was developed in the ACT-R framework (version 7.14) for comparison with the ACT-R/PCM model. There are only two productions in the model. The first finds the visual location of the tracking reticle when the visual and motor modules are free, and the second moves visual attention to the reticle in parallel with a manual motor movement toward the center of the tracking window. The extent of the joystick movement was scaled linearly so that the maximum joystick deflection would result when the current error (distance between the reticle and center) was at its maximum (reticle in a corner of the tracking window).

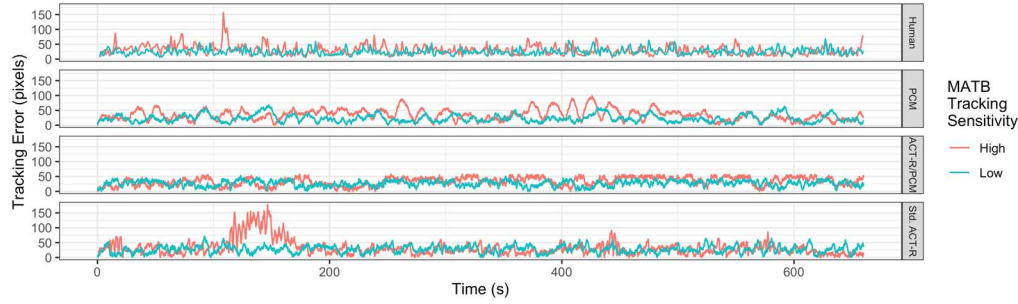


Figure 2 : RMSE over time for one of the three human participants and each model. Red lines show RMSE in the high sensitivity condition, and the green lines show RMSE for the low condition.

Results

How does the ACT-R/PCM model compare in tracking performance relative to its counterparts -- PCM and Standard ACT-R? To address this question, we ran simulations with the ACT-R/PCM, PCM, and Standard ACT-R models. Then, we compared RMSE values between human data and each of the models.

There are two conditions that the model was tested under: low sensitivity and high sensitivity. As discussed in the Task Environment section, the low and high sensitivity correspond to the gain of the simulated joystick. In the human data, there is a trend in mean RMSE between the low and high conditions, in which the mean and variability of RMSE is higher in the high sensitivity condition compared to the low sensitivity condition. We were interested in observing if the ACT-R/PCM model can capture similar trends.

To examine how effectively the ACT-R/PCM model captures human tracking behavior relative to the PCM model and Standard ACT-R model, we measured: (1) mean and standard deviation of RMSE across low and high sensitivity conditions, and (2) RMSE over time. Model fits were evaluated using the sum of the mean RMSE error and mean standard deviation (SD) error between the human and model data. Note: the trials we used to compare these models have different sequences of randomized perturbations for the cursor. The following sections will describe the parameter values that produced RMSE values that most closely matched the human tracking data for each of the three models.

PCM Fitting

The damping constant K_d , delay τ_d , and gain K_o parameters were each varied by sampling ten evenly spaced values within the respective ranges to find a good fit: 0.01 to 0.06, 0.0 to 10ms, and 1.0 to 5.0. The reference signal parameter R was not varied and maintained a value of 0 (i.e. the desired position of the cursor is at the target location). The model was run 30 times for each parameter value.

For the low and high sensitivity conditions, delay τ_d and damping constant K_d were held constant and showed a goodness of fit at 8.33ms and 0.06. A K_o parameter value of 1.22 produced an acceptable fit of the low sensitivity

condition with a 4.9% error in mean and a 3.4% error in standard deviation, as seen in Table 1.

ACT-R/PCM Fitting

The gain PCM K_o parameter was varied from 4 to 8.0 in increments of 0.25 to find a good fit. All other PCM parameters were left at the best fitting values for the standalone PCM model ($\tau_d = 8.33$; $K_d = 0.06$). All ACT-R parameters were left at default. The model was run 30 times for each parameter value.

A K_o parameter value of 7.25 produced a good fit of the low sensitivity condition with a 2.7% error in mean and a 9.0% in standard deviation. A K_o parameter value of 5.25 produced an acceptable fit of the high sensitivity condition with a 12% error in mean and an 11.6% error in standard deviation, as seen in Table 1.

Standard ACT-R Model Fitting

Except for the cursor Fitts' coefficient parameter (:cursor-fitts-coeff), all parameters were left at default. For all parameter values, the model was ran 30 times. The Fitts' coefficient parameter was systematically varied between the default value (0.1) and 0.2 in increments of 0.05. A parameter value of 0.145 produced a very good fit to the low sensitivity condition with a 0.7% error in mean and a 5.1% error in standard deviation. A parameter value of 0.155 produced the best fit to the high sensitivity condition with a 12.5% error in mean and a 1.4% error in standard deviation (Table 1).

The bottom panel in Figure 2 shows a randomly selected trial from these fits. As can be seen, the poor fit of the high sensitivity condition is a result of underpredicting the mean error while predicting variance in the error that is more sporadic than observed. To predict a greater mean error in the high sensitivity condition required a substantially higher Fitts' coefficient parameter, which then results in variance in the data that is too incredibly high.

Discussion

Comparative Analysis

In this study, we are interested in comparatively analyzing the tracking performance of the ACT-R/PCM model relative to the PCM and Standard ACT-R models. We looked at how each of the three models compared in terms of (1) capturing mean RMSE and standard deviation of RMSE across the low and high sensitivity conditions, and (2) qualitative differences in trends of RMSE over time across the low and high sensitivity conditions. The following sections will describe our findings across these analyses.

Trends of Mean and Standard Deviation of RMSE

We captured human-like tracking behavior for the PCM and ACT-R/PCM models by setting consistent delay τ_d and damping constant K_d values at 8.33ms and 0.06 across both models, as well as across both the low and high sensitivity conditions. Only gain K_o was varied between the low and high conditions and between models – revealing that gain K_o was higher in the high sensitivity condition compared to the low sensitivity condition.

We propose that the gain parameter values were different between low and high sensitivity conditions because subjects had to adapt to the differences in gain between the two conditions. Additionally, we propose that the gain parameters were different between the PCM and ACT-R/PCM models, as the ACT-R/PCM model contained two gain parameters that affected the output: gain K_o for the PCM model, and another gain for the simulated joystick. Thus, the difference in gain dynamics between the two models may have been the cause to differences in gain K_o fits.

As shown in Table 1, all three models captured the trend of a lower mean and standard deviation of RMSE in the low sensitivity condition vs. the high sensitivity condition. The standard ACT-R model produced values with the lowest error percentage. However, the Standard ACT-R model did not capture the differences in mean RMSE between the low and high conditions as well as the PCM or ACT-R/PCM models, as the Standard ACT-R mean RMSE values between conditions were more consistent (Figure 1). Although the Standard ACT-R model best fits the human data, it does not capture behavioral differences caused by joystick sensitivity as well as the PCM or ACT-R/PCM models.

Trends of RMSE over Time

When observing RMSE over time for the human subjects, there is higher variance in the high sensitivity condition compared to the low sensitivity condition – as shown by the taller red spikes that intermittently appear across a given trial (Figure 2). We propose that, as the high sensitivity condition increases the gain of the joystick for participants, there is less stability in human tracking movements compared to the low sensitivity condition. When examining the RMSE over time for the three tracking models, we find that the Standard ACT-R tracking model also exhibits higher variance for the high

sensitivity condition, showing intermittent spikes as well. However, these spikes show significantly higher error and persist for longer than the spikes shown in human data. When examining the PCM model, it shows higher variance and a higher frequency in spikes in the high sensitivity condition – however, these spikes are much more regular and closer in resemblance to the human data. The ACT-R/PCM model also shows higher variance for the high sensitivity condition; however, it does not appear to produce spikes as high in error as the human data.

We propose that these qualitative differences in variance across these models may be due to the different capabilities between the models. While fitting PCM model to the human data, we found that the damping constant K_d allowed the PCM model to produce human-like variance in the tracking movements. Without K_d , the PCM model would go through phases where it would accumulate error over time and destabilize, producing increasingly larger control adjustments in response to the growth in error. The damping constant prevents destabilization by scaling down the control adjustments to smaller distances. The Standard ACT-R model does not contain a damping constant; thus, if error starts to rapidly accumulate, the model produces control adjustments that are less stable and more persistent than human control adjustments.

Limitations

Due to limitations in the human tracking data, we were unable to conduct analyses that would allow us to examine certain aspects of human tracking behavior. Specifically, the time resolution of the data was relatively coarse, at 100ms per second. If humans made control adjustments between these 100ms increments, we were not able to capture that with our current dataset. Additionally, the only tracking data we acquired was RMSE over time. We did not have access to the location of the cursor over time, thus were not able to make distinctions in error along the x-axis or y-axis. In future work, we would like to acquire human tracking data that contains cursor position information, as well as a higher time resolution.

Conclusion

The aim of this study was to examine if integrating the position control model (PCM) within the ACT-R framework would increase not only the cognitive fidelity of the model's tracking behavior, but also its performance in capturing human-like tracking behavior movements. When comparing RMSE between human tracking data and the PCM, ACT-R/PCM, and Standard ACT-R models, results indicated that the Standard ACT-R model produced the best overall fit to human data. However, the PCM model performed the most successfully in (1) capturing mean differences in RMSE between low and high conditions, and (2) producing human-like variance in RMSE over time. In future work, we would like to simulate these models in a multitasking context, as the MATB is primarily used to investigate multitasking behavior.

Acknowledgements

This research was supported by Air Force Research Laboratory's Airmen System Directorate's Repperger Internship Program through the Oak Ridge Institute for Science and Education.

References

- Anderson, J. R. (2009). *How can the human mind occur in the physical universe?* Oxford University Press.
- Balint, J. T., Reynolds, B., Blaha, L. M., & Halverson, T. (2017). Visualizing eye movements in formal cognitive models. In *Mathematics and Visualization*.
- Ballas, J., Kieras, D., Meyer, D., Stroup, J., & Brock, D. (1999). How is Tracking Affected by Actions on Another Task? *Proceedings of the International Symposium on Aviation Psychology*. International Symposium on Aviation Psychology, Columbus, OH.
- Bourbon, W. T., & Powers, W. T. (1999). Models and their worlds. *International Journal of Human-Computer Studies*, 50(6), 445-461.
- Brumby, D. P., Salvucci, D. D., & Howes, A. (2009, April). Focus on driving: How cognitive constraints shape the adaptation of strategy when dialing while driving. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 1629-1638).
- Chernikoff, R., Brimingham, H. P., & Taylor, F. V. (1955). A comparison of pursuit and compensatory tracking under conditions of aiding and no aiding. *Journal of Experimental Psychology*, 49(1), 55.
- Craik, K. J. (1948). Theory of the human operator in control systems. II. Man as an element in a control system. *British journal of psychology*, 38(3), 142.
- Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6), 381-391.
- Gibson, J. J. (1986). *The ecological approach to visual perception: classic edition*. Psychology press
- Marken, R. S., & Powers, W. T. (1989). Levels of intention in behavior. In *Advances in Psychology* (Vol. 62, pp. 409-430). North-Holland.
- Markkula, G., Boer, E., Romano, R., & Merat, N. (2018). Sustained sensorimotor control as intermittent decisions about prediction errors: Computational framework and application to ground vehicle steering. *Biological cybernetics*, 112, 181-207.
- McRuer, D. T., & Jex, H. R. (1967). A review of quasi-linear pilot models. *IEEE transactions on human factors in electronics*, (3), 231-249.
- Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic Mechanisms. *Psychological Review*, 104(1), 3-65.
- Navas, F., & Stark, L. (1968). Sampling or intermittency in hand control system dynamics. *Biophysical Journal*, 8(2), 252-302.
- Parker, M. G., Tyson, S. F., Weightman, A. P., Abbott, B., Emsley, R., & Mansell, W. (2017). Perceptual control models of pursuit manual tracking demonstrate individual specificity and parameter consistency. *Attention, Perception, & Psychophysics*, 79, 2523-2537.
- Parker, M. G., Willett, A. B., Tyson, S. F., Weightman, A. P., & Mansell, W. (2020). A systematic evaluation of the evidence for perceptual control theory in tracking studies. *Neuroscience & Biobehavioral Reviews*, 112, 616-633.
- Poulton, E. C. (1952a). Perceptual anticipation in tracking with two-pointer and one-pointer displays. *British Journal of Psychology*, 43(3), 222.
- Powers, W. (1973). Perceptual control theory.
- Powers, W. T. (1978). Quantitative analysis of purposive systems: Some spadework at the foundations of scientific psychology. *Psychological Review*, 85(5), 417.
- Powers, W. T. (2008). Living control systems III: The fact of control.
- Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Human Factors*, 48(2), 362-380.
- Santiago-Espada, Y., Myer, R. R., Latorella, K. A., & Comstock, J. R., Jr. (2011). *The Multi-Attribute Task Battery II (MATB-II) Software for Human Performance and Workload Research: A User's Guide* (No. TM-2011-217164). NASA.
- Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, 12(1), 97-136.
- Warren, W. H. (2006). The dynamics of perception and action. *Psychological review*, 113(2), 358.
- Zhang, Y., & Hornof, A. J. (2012). A discrete movement model for cursor tracking validated in the context of a dual-task experiment. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 1000-1004.

A Cognitive Model of a Temporal Binding Task

Laura Saad (laura.saad@rutgers.edu)

Department of Psychology, Rutgers University. New Brunswick, NJ

Alexander R. Hough (alexander.hough.1@us.af.mil)

Leslie M. Blaha (leslie.blaha@us.af.mil)

Air Force Research Laboratory, Wright-Patterson AFB

Christian Lebiere (cl@cmu.edu)

Department of Psychology, Carnegie Mellon University. Pittsburgh, PA

Abstract

Temporal binding (TB) is the subjective compression between a voluntary action and its associated outcome. It is regarded as an implicit measure of the sense of agency; however, an underlying mechanism has yet to be agreed upon. Previous research suggests memory as an alternative explanation for TB in two publicly available datasets. We test this idea by implementing a model within the ACT-R cognitive architecture and leveraging its existing memory and time perception mechanisms to simulate participants from these datasets. Our simulations provide evidence to suggest that memory and time perception mechanisms can explain the pattern of results. Implications for temporal binding and the sense of agency are discussed.

Keywords: Temporal Binding; ACT-R; Cognitive Models; Sense of Agency; Time Perception

Introduction

Temporal binding (TB) is assumed to be an implicit measure of the sense of agency. TB is defined as the perceived subjective compression of time between a voluntary action and its associated outcome (Haggard, Clark, & Kalogeras, 2002). In the seminal study by Haggard et al. (2002), participants were asked to press a button at a time of their choosing; a few hundred milliseconds later, there was an audible tone. Participants were then asked to estimate the timing of their button press and the tone. The key finding was that when the action is voluntary—as opposed to involuntary—participants subjectively estimated that their button press occurred later than it objectively did. Furthermore, participants also subjectively estimated that the tone occurred earlier than it objectively did. This compression, or underestimation, of the subjective time interval between the action and its outcome is what is known as *temporal binding*. Importantly, the opposite effect, or a repulsion of the subjective time interval, occurred for involuntary actions (e.g., finger twitch produced by transcranial magnetic stimulation of the motor cortex) and their outcomes. This difference in the pattern of results between these two conditions led to the conclusion that TB is an implicit marker for the sense of agency.

There is some theoretical debate in the literature over whether it is the presence of a voluntary action, and therefore intentionality (Haggard, 2005), or the perceived causality between events (Hoerl et al., 2020) that is necessary to elicit TB. One reconciliatory explanation for TB comes from cue integration theory (Ernst & Banks, 2002). This theory suggests the motor system optimally combines cues from different sources to reduce the overall variability of estimates.

Cues are weighted by their reliability such that information from more reliable cues is more heavily weighted in the integration process. There has been one successful formal implementation of a Bayesian cue integration model in the context of TB (Legaspi & Toyoizumi, 2019); it remains unclear if this model can be applied to all timing estimation methods.

Though most TB tasks involve free recall, the role of memory has been largely underexplored in this literature. Recently, a memory process was proposed as a potential explanation for TB (Saad, Musolino, & Hemmer, 2022). In this paper, a regression pattern was revealed by re-plotting participant estimates from two publicly available datasets (Weller, Schwarz, Kunde, & Pfister, 2020) as the difference between the subjective responses and objective values (i.e., bias). Regression here refers to the bias in estimations such that participants, when making estimations, select a value closer to the mean of intervals observed in the task. This regression pattern replicated across conditions regardless of the agency manipulation. Saad et al. (2022) then successfully simulated participant estimates using a Bayesian rational memory model. This provided the first evidence that a memory mechanism could account for estimations at the aggregate level in a TB task.

Relatedly, the role of time perception in eliciting the TB effect has also been understudied in this literature. During encoding, participants perceive the timing of or the intervals between events. One mechanism that has been proposed to explain this is a pacemaker-accumulator process, where a pacemaker produces pulses at some rate, and these pulses are counted in an accumulator. The perceived length of the interval between two events is a function of how many pulses are in the accumulator; more pulses correspond to a longer perceived duration. This mechanism makes a prediction that a shortening of a perceived time interval (i.e., the compression characteristic of the TB effect) is a result of a slower pulse rate leading to fewer pulses in the accumulator.

Fereday, Buehner, and Rushton (2019) empirically investigated whether internal clock slowing is a viable explanatory mechanism for TB. In two experiments, the authors incorporated a temporal discrimination task where participants compared durations of causal (button press and a flash) and non-causal trials (two flashes) to a reference duration (black square presented on screen) and were asked to report which interval length was longer. The authors calculated point of subjective equality (PSE) values across conditions. The PSE

value represents the duration of the comparison interval (i.e., causal or non-causal) that is perceived as the same as the reference interval 50% of the time. The prediction is such that lower values of PSE correspond to more compression or underestimation (i.e., binding). The authors reported evidence to support this prediction in both experiments.

Although the pacemaker-accumulator process makes explicit predictions which can be tested empirically, a formal model of this process has never been implemented. Importantly, though much of the TB literature is focused on developing a theory of agency, there has not yet been an investigation into how memory and time perception processes may work together to influence or explain the temporal binding effect. We aim to do just this. We hypothesize the regression pattern in the human data results from a memory mechanism where participants estimate time intervals according to a pacemaker-accumulator process and then use estimates from previous trials during recall. We develop a cognitive model to test this hypothesis; the model specifies mechanisms for memory and time perception and is capable of simulating human performance in one condition in a TB task. We then explore how estimating different parameters related to memory mechanisms in the model affect simulation results at the aggregate and individual level.

We focus first on simulating one trial-type, action trials, because they represent the most frequently used trial-types in this literature. Additionally, our initial aim was to establish a cognitive model as a viable means for simulating human behavior in these tasks. These results lay the groundwork for future simulations using the same cognitive model to simulate the passive, comparison trial-type and therefore the entire temporal binding effect. We discuss this and other ideas for future work in the final section of the paper.

Method

Data Set

We model the publicly available data from experiment 3A in Weller et al. (2020). Code for all analysis, figures, and supplementary material included in this paper are also publicly available (<https://osf.io/6bkjp/>). A detailed description of the experimental method and procedure can be found in the original Weller et al. (2020) paper.

We briefly describe the procedure for experiment 3A. The experiment included three trial-types: action, non-action, and baseline. At the beginning of each trial, participants were asked to choose between an action and a non-action which would each produce distinct outcomes. These trials were called operant trials. During non-action trials, participants chose not to act and a default outcome would occur; in the action trials, participants acted by pressing a button at a timing of their choosing to change the default outcome. In the baseline trials, there was no initial decision necessary, and participants passively watched two events unfold: a progress bar filled which ended in a “click” sound; then a ball launched in a pre-specified direction.

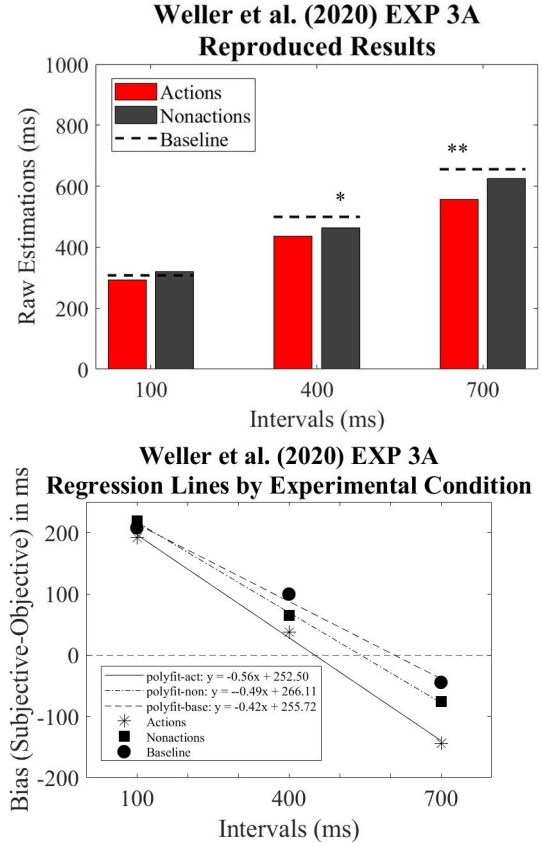


Figure 1: Reproduction of graphs from Weller et al. (2020) depicting mean raw estimations across three trial-types (top panel). Re-plotting these estimates as bias (bottom panel) reveals a consistent regression pattern across intervals. Data plotted here are from 27 participants. * $p < 0.05$, ** $p < 0.01$

At the end of each trial, regardless of which type, participants were asked to recall and report their estimate of the interval between two events (i.e., either the keypress and ball launch in action trials, or the clicking sound and ball launch in non-action and baseline trials) in milliseconds using a slider on-screen. Three different time intervals were used between events: 100ms, 400ms, and 700ms. The presentation of these intervals was randomized across the different blocks of trials.

At the beginning of the experiment, participants completed a series of 20 practice trials, 10 baseline and 10 operant. Practice trials included time intervals between 100ms and 1000ms in steps of 100ms. Participants received feedback about the accuracy of their estimation at the end of each trial. Data was not collected or analyzed for practice trials. During the main experiment, no feedback was given.

Weller et al. (2020) reported statistical results comparing TB values across trial-type and delay ($N=27$), and reported two significant results for experiment 3A: more binding (or more compression) for actions compared to baseline at the 700ms interval and for non-actions compared to baseline at the 400ms interval. No other comparisons were significant. From these results, Weller et al. (2020) concluded that “temporal binding ha[d] [also] emerged for non-actions” (p. 8).

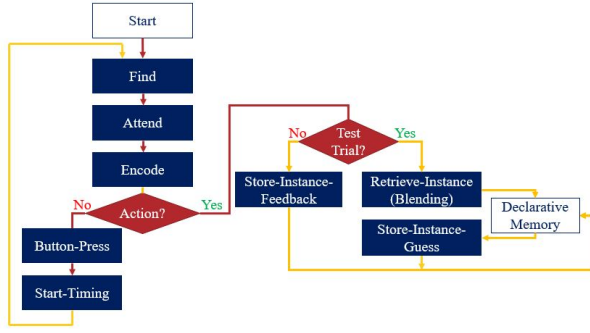


Figure 2: Visualization of TBM model processes.

Figure 1 reproduces the original visualization of Weller et al.’s results (top) and the regression effect (bottom) when participants’ raw estimates are re-plotted as bias, or the difference between the average estimates and the actual length of each interval. When the estimates are re-plotted as bias, a clear regression pattern is revealed on all three trial-types.

The Temporal Binding Memory (TBM) Model

We developed a cognitive model called the Temporal Binding Memory (i.e., TBM) model based on the action trials from the Weller et al. (2020) TB task. The model was implemented in ACT-R, which is a hybrid cognitive architecture used to understand and simulate human cognition. ACT-R contains a set of modules which perform distinct cognitive functions and communicate via requests relayed through limited-capacity buffers. The TBM model used the imaginal, goal, vision, motor, procedural, declarative memory, and temporal modules.

The imaginal module holds the problem representation, and the goal module represents the model’s current task focus. The vision module represents a visual attention system containing both a “what” and a “where” subsystem. The motor module represents two hands on a virtual keyboard. The procedural module, or production system, is a pattern matching system which constantly searches for productions matching the current state of the buffers using conditional statements (i.e., if-then rules). Only one production can be executed at a time. When a production is executed, or “fired”, the state of the system changes, progressing the model through a task. Knowledge is represented in the form of chunks in declarative memory, which each have an activation value corresponding to the recency and frequency of the chunk with some noise. Chunks are retrieved via the retrieval buffer in the declarative memory module which searches through declarative memory to find a chunk with the highest activation value to satisfy the current request.

The temporal module, created to represent subjective time estimation between two events (Taatgen, Van Rijn, & Anderson, 2007), models time perception as a pacemaker-accumulator process. The pacemaker generates pulses, and the accumulator counts them. Tick lengths are noisy and increase in duration as time progresses, which means the tem-

poral module is more accurate for shorter compared to longer time intervals. Tick lengths are based on the following equations for the n^{th} tick.

$$t_0 = \text{start} + \epsilon_1 \quad (1)$$

$$t_n = a * t_{n-1} + \epsilon_2 \quad (2)$$

The length of the first tick, t_0 , is controlled by the *start* parameter (default = .011) with some noise. The *a* parameter (default = 1.1) affects the length of subsequent ticks. Noise is added to tick lengths using the *act-r-noise* command, and the *s* values for each are according to the following equations:

$$\epsilon_1, s = b * 5 * \text{start} \quad (3)$$

$$\epsilon_2, s = b * a * t_{n-1} \quad (4)$$

The *b* parameter is set to 0.015. The model recalls the intervals on each trial by accessing the current pulse value in the temporal buffer and reporting the tick count.

ACT-R models are cognitive models that specify mechanisms at the algorithmic level (Marr, 1982) and therefore require simulation of both the task and cognitive processes. As this is the first cognitive model of a TB task (to our knowledge), we aimed to replicate the major components of the experimental design (i.e., interval lengths, outcome modality, presence of feedback, and practice trials). However, we simplified stimuli and simulated action trials first, as these are the type of trial in which TB has been reported most frequently.

Figure 2 displays the steps the model completed to estimate time in our modified task. Each rectangle represents a separate production. Letters were used as cues for the beginning and end of the interval that was timed by the model. During both practice and test trials, the presentation of the first stimulus, “A”, initiated the proceeding course of events. First, the model looped through a standard find-attend-encode loop by which the visual system located and then encoded the visual information on the virtual screen. After encoding, the model pressed the “A” key on the virtual keyboard, initiating the timing process by making a temporal buffer request to start timing in ticks. This triggered the presentation of another visual stimulus, “Z”. The same find-attend-encode procedure was completed in response to this second visual stimulus before proceeding to the store-instance-feedback production. The tick count was stopped once the letter “Z” was perceived.

During each run, the model completed two different types of trials: 20 practice and 150 test. The key factor differentiating practice and test trials is that during practice trials, the model received feedback in the form of a real-time millisecond interval value which was then paired with the tick count from the temporal buffer and stored in declarative memory. During test, no feedback was given, and the tick count value on each trial was paired with a guess. This guess was informed by the chunks in memory that were formed during practice. We will now describe each trial type in detail.

Practice trials heavily influenced model performance as they generated chunks that were later retrieved by the model

and were the basis for the model's responses during test trials. During practice trials, feedback was provided after estimations to mimic the participants' experience in the original study (Weller et al., 2020). This feedback was used to build chunks which contained two slots: tick count (from the temporal buffer) and real-time delay which stored the actual delay given as feedback. Each chunk was encoded into declarative memory, and the entire process was repeated. The length of the interval between the action and outcome was manipulated depending on the trial type. During practice trials, we randomly presented twice each the intervals from 100ms to 1000ms in steps of 100ms, for a total of 20 practice trials.

During test trials, the process was similar except only three intervals were used (100, 400, and 700ms), and there was no feedback provided. The blending mechanism (Lebiere, 1999) was used to retrieve chunks from declarative memory. The blending mechanism computes a weighted average over chunks in memory learned during practice such that chunks with a higher likelihood of retrieval, determined by activation, carry more weight. The ACT-R activation equation,

$$A_i = B_i + S_i + P_i + \epsilon_i \quad (5)$$

includes a: 1) base level term, B_i , for recency and frequency of use, 2) spreading term, S_i , for context effects, 3) partial matching term, P_i , for degree of match with retrieval cues, and 4) noise term, ϵ_i , for noise in memory. The blending mechanism uses the equation,

$$V = \operatorname{argmin}_V \sum_i P_i (1 - \operatorname{sim}(V, V_i))^2 \quad (6)$$

to produce a value that minimizes the sum of all squared dissimilarities, $((1 - \operatorname{sim}(V, V_i))^2)$, of each chunk, i , between the consensus value V and the chunk value V_i , and weights it by its probability of retrieval,

$$P_i = \frac{e^{A_i/t}}{\sum_j e^{A_j/t}}. \quad (7)$$

The probability of retrieval is a function of the activation for a chunk, $e^{A_i/t}$, normalized by the activation of all retrieved chunks, $\sum_j e^{A_j/t}$.

During recall, the current tick count at the time the second stimulus appeared was used as a retrieval cue to find a match in declarative memory in the retrieve-instance-feedback production. We defined a linear similarity function for ticks in the temporal buffer which impacted how chunks were weighted during this retrieval process and how the blending average was computed. As ticks in the temporal buffer operate according to a log-scale (to reflect the scalar property of time estimations), we thought it appropriate to define ticks as having a relationship such that tick values were most similar to themselves with linearly decreasing similarity. At the end of each trial, a new chunk was created pairing the blended real-time value (i.e., the guess) and the tick count. This process was repeated over 150 trials. The accumulation

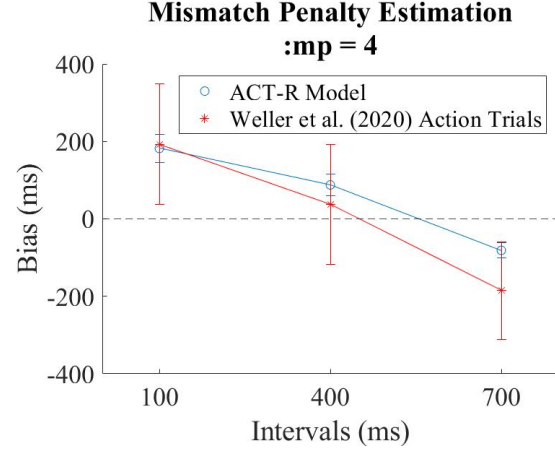


Figure 3: Best fitting model from analysis testing mismatch penalty (:mp) parameter values.

of chunks during the experimental trials resulted in a gradual regression pattern in model estimates.

Before comparing model and human performance at the aggregate level, we completed a series of simulations to estimate the mismatch penalty (:mp) parameter as there is no default. The :mp parameter specifies the penalty, (P_i), in the activation equation and calibrates the degree of regression to the mean in the model. Lower values of :mp correspond to a wider range of chunks taken into account during retrieval. We tested values of :mp from 1 to 5 in steps of 0.5, keeping all other parameter at their default values, and we simulated 10 model runs per parameter value for a total of 90 model runs.

Results

The results from the :mp estimation are shown in Figure 3, where the best fitting model and human performance at the aggregate (across trials and individuals) are plotted together.

To determine the model with the best quantitative fit, we computed the root mean squared error (RMSE) for the difference between each model's simulated estimates compared to the human estimates. We also computed a Pearson correlation between the human and model estimates across the three interval lengths ($r = 0.99$, $p = 0.04$). The model with :mp = 4 produced the lowest $RMSE = 67.02$.

Though the adjustment of the :mp parameter improved the fit at the aggregate level, there was still a substantial difference in variance between the human data and model fit. This difference can be seen in Figure 4 which plots individual participant and model run data at the trial level using the parameter settings from the first analysis. When comparing across timing intervals (colored dots), it is clear that the variability in the model is substantially less compared to the human data. These results indicate that the model was not capturing individual level behavior.

To investigate this, we explored one parameter that represents a plausible way to account for the individual variability: blending temperature (:tmp). The :tmp parameter controls

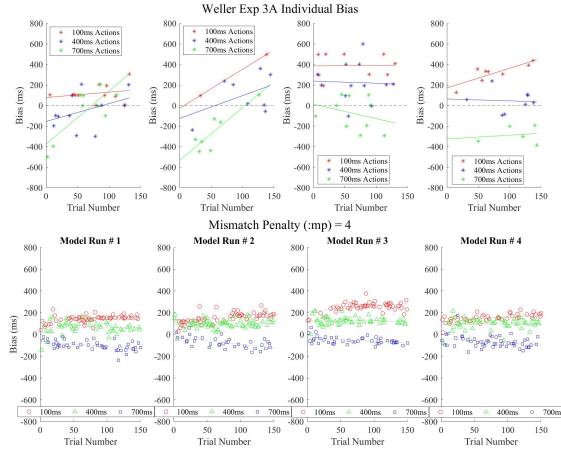


Figure 4: Individual variability and linear fits across human (top) and model simulation (bottom) data. Data points represent an estimate on a given trial across the interval lengths.

the preference to blend chunks in memory. Higher values of :tmp correspond to more blending over chunks and, therefore, more regression to the mean of chunks. Lower values of :tmp correspond to a retrieval process closer to the best match, or “winner-take-all”. We tested 30 values of :tmp sampled from a normal distribution ($\mu = 0.5$, $\sigma = 0.1$). For this analysis, we kept all other parameters at default, except for :mp = 4.

Figure 5 depicts the results from this analysis. We computed the RMSE to evaluate quantitative fit of the model simulation to the human data. Here, the model with the lowest average difference ($RMSE = 62.29$) used a :tmp value of 0.453. It produced only a marginal improvement from the first analysis ($RMSE = 67.02$) which did not set the :tmp parameter and only adjusted :mp to 4. We computed a correlation between the human data and model fit across the three interval lengths ($r = 0.99$, $p < 0.001$) which represented a good quantitative fit at the aggregate level. However, this value of :tmp did not improve fit to the individuals. As this was the aim of this analysis, these results indicate adjusting this parameter in future simulations may not be necessary.¹

Discussion

Here we have developed and implemented the first cognitive model of a TB task. Using core components from the ACT-R architecture and default settings for all but one parameter (mismatch penalty), our model was able to simulate human time interval estimates in action trials from a TB dataset

¹We conducted an additional analysis to investigate how varying the amount of noise added between ticks, via the :time-noise parameter in the temporal buffer (b , in Equations 3 and 4), influenced variance in timing estimates. We assessed values of 0.005 to 0.1 in steps of 0.015 keeping all other parameters at default except :mp = 4. The best fitting value was :time-noise = 0.03, which is slightly higher than the default value 0.015. The minimum $RMSE = 51.11$ improved model performance at aggregate, but not substantially enough at the individual level to warrant adjusting the default setting. This represents an interesting area of investigation for future work. See <https://osf.io/6bkjp/> for complete details.

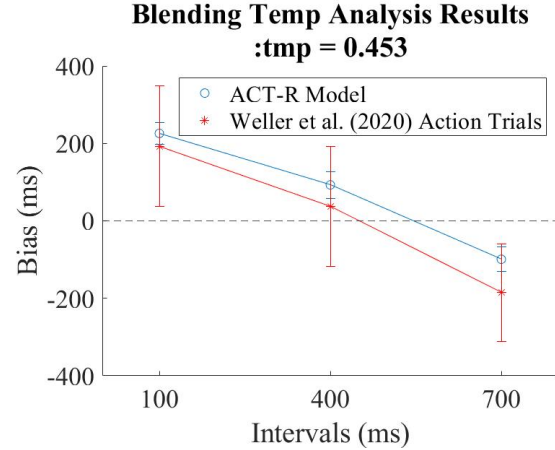


Figure 5: Best fitting model from the blending temperature (:tmp) parameter analysis.

(Weller et al., 2020).

After defining a similarity function to specify how chunks were weighted during retrieval, our first aim was to estimate the appropriate value for the :mp parameter. Using the minimum RMSE value as the primary metric for quantitative fit, the simulations suggested only adjusting the :mp parameter to 4. This is relatively high in the range of values tested. Conceptually, this means that to simulate participant estimations in this task, the model needed a higher penalty against chunks in memory which limited the number of chunks that are averaged in memory during a single retrieval to those that were a close match to the current retrieval request.

In our second analysis, we evaluated sources of individual variability using another parameter affecting memory mechanisms, the :tmp parameter. The best fitting value provided a marginal improvement of model fit at the aggregate across three time intervals but did not improve variability comparable to human performance. Future work could allow variation in the :tmp parameter to represent individual differences in humans regarding whether they use more (i.e., blending) or fewer (i.e., single best match) previous instances in memory to inform current estimates of time intervals.

In the TBM model, parameters affecting memory mechanisms within ACT-R were the primary influence on model performance at the aggregate level. This provided some additional evidence to suggest that memory mechanisms are capable of capturing the patterns in human data from a TB task. Surprisingly, our investigation of the timing mechanism (:time-noise) did not appear to affect model performance at the individual level as much as expected (see Footnote 1 for more details). Increasing values of the :time-noise parameter increased individual level variability but worsened the fit at the aggregate level, indicating a trade-off. It may be the case that conducting an analysis similar to the one we suggest for blending temperature (i.e., altering :time-noise parameter value to fit individuals) may represent a feasible way forward

in investigating sources of noise at the individual level.

It is also possible other aspects of the temporal buffer may affect model performance in this task. As mentioned in the introduction, according to Fereday et al. (2019) one might expect a slower clock rate when comparing action trials to passive ones. In the ACT-R architecture, the :time-mult parameter, which controls a multiplier constant applied to each tick, could be adjusted to formally investigate this hypothesis. Specifically, one might test a range of values higher and lower than the default value (1.1) to investigate whether this can capture any observed differences between conditions.

It is important to note that so far the majority of the analyses we describe here have been conducted at the aggregate level. However, there is evidence to suggest that the TB effect is not consistently present at the individual level, and currently this variability is not captured by this model. There are some reasons why this might be the case. For instance, one can interpret the aggregate model fit to represent one participant completing the experiment 10 times without variation. The observed variability in the human data then would potentially reflect an aggregation over different models using different parameter values. Future work should aim to investigate sources of variability across individuals and possible explanatory mechanisms.

Future work might also investigate the sources of individual variability in estimations. Currently the information in declarative memory is created in the same way across individual model runs (i.e., assuming no prior experience before beginning the task). This is due to the fact that we did not have access to practice trial data. Seeding each model run with actual practice trial data may better simulate the individual variability in the human data. In lieu of practice data, it may be useful to simulate a large number of practice trials at sub-second interval lengths to build a more realistic declarative memory store. Additionally, it may be useful to incorporate individual differences in the initial guesses the model makes during the task (e.g., Cranford et al. (2021)). Participants' initial guesses may be based on environmental priors (i.e., we expect our button press to lead to an outcome after a very short interval, typically less than 100ms in length). These expectations may vary across participants, which may lead to some of the variability present at the individual level.

As mentioned in the introduction, due to the preliminary nature of this work, our analysis did not include a simulation of the comparison trial-type that is used to determine the presence of the TB effect. In experiment 3A Weller et al. (2020), a baseline trial-type wherein participants passively observed two events, was the comparison of interest for the action trials. Baseline trials can be simulated by removing the initial voluntary action in our current task, so that the model passively observes and times the interval length between the presentation of the two visual stimuli (e.g., the letters "A" and "Z"). The estimations in this trial-type can then be compared to the action trial-type that we have developed here to determine whether the TBM model can account for the entire TB

effect (i.e., more compression in the action trial-type compared to the baseline trial-type). We suggest first simulating both trial-types using the same core components and parameter values to evaluate whether additional specifications are necessary to produce the TB effect.

In conclusion, we have successfully developed and implemented the first cognitive model of a temporal binding task. An ACT-R model, using declarative memory and time perception mechanisms, provided a good qualitative fit to human data. These results, while still preliminary, add to the growing evidence that memory mechanisms can account for results from temporal binding studies. Future work should evaluate whether specifying an agency mechanism is necessary to account for the temporal binding effect.

Acknowledgments

Part of this work was completed while L.S. was an ORISE intern at AFRL, supported by the 711th Human Performance Wing. C.L. was supported by the Center of Excellence for Trusted Human-Autonomy Teaming (AFRL/AFOSR award FA9550-18-1-0251). The views expressed are those of the authors and do not reflect the official guidance or position of the United States Government, the Department of Defense, or the United States Air Force. Distribution A. Approved for public release; distribution unlimited (AFRL-2023-1412).

References

- Cranford, E. A., Gonzalez, C., Aggarwal, P., Tambe, M., Cooney, S., & Lebiere, C. (2021). Towards a cognitive theory of cyber deception. *Cognitive Science*, 45(7), e13013.
- Ernst, M. O., & Banks, M. S. (2002). Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415(6870), 429–433.
- Fereday, R., Buehner, M. J., & Rushton, S. K. (2019). The role of time perception in temporal binding: Impaired temporal resolution in causal sequences. *Cognition*, 193, 104005.
- Haggard, P. (2005). Conscious intention and motor cognition. *Trends in Cognitive Sciences*, 9(6), 290–295.
- Haggard, P., Clark, S., & Kalogeras, J. (2002). Voluntary action and conscious awareness. *Nature Neuroscience*, 5(4), 382–385.
- Hoerl, C., Lorimer, S., McCormack, T., Lagnado, D. A., Blakey, E., Tecwyn, E. C., & Buehner, M. J. (2020). Temporal binding, causation, and agency: Developing a new theoretical framework. *Cognitive Science*, 44(5), e12843.
- Lebiere, C. (1999). Blending: An ACT-R mechanism for aggregate retrievals. In *Proceedings of the Sixth Annual ACT-R Workshop*. George Mason University, Fairfax, VA, USA.
- Legaspi, R., & Toyoizumi, T. (2019). A Bayesian psychophysics model of sense of agency. *Nature Communications*, 10(1), 4250.
- Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. MIT press.

- Saad, L., Musolino, J., & Hemmer, P. (2022). Bayesian rational memory model simulates temporal binding effect. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 44).
- Taatgen, N. A., Van Rijn, H., & Anderson, J. (2007). An integrated theory of prospective time interval estimation: the role of cognition, attention, and learning. *Psychological Review*, 114(3), 577.
- Weller, L., Schwarz, K. A., Kunde, W., & Pfister, R. (2020). Something from nothing: Agency for deliberate nonactions. *Cognition*, 196, 104136.

Moral Judgments as the Combination of Distributed Language Representation and Memory Activation Mechanism

Kenya Sasaki, Jumpei Nishikawa, Kazuma Nagashima, Junya Morita
Shizuoka University, 3-5-1 Johoku, Naka-ku, Hamamatsu-shi, Shizuoka-ken
Sasaki.kenya.21@shizuoka.ac.jp

Keywords: moral judgments, dual-process theory, ACT-R

Introduction

In recent years, research on autonomously operating machines, such as self-driving cars, has progressively increased. When the social implications of such technologies are discussed, moral judgment by computers often becomes an issue. As autonomy increases to a high degree in the future, an in-depth understanding of human morality will become necessary in thinking systems. Even though individual human beings value morality, conflicts may arise. Therefore, it is necessary to examine how humans perceive complex society through morality and how humans make decisions based on morality.

This research examines the mechanism of human moral judgment based on System 1 and System 2 (Kahneman, 2012) as described by the dual-process theory (Evans and Stanovich, 2013). The Trolley Problem (Thomson, 1984), one of the famous thought experiments in moral judgment, is used as a task, and a model for making judgments based on memories by giving Japanese sentences of the problem is developed based on distributed language models and the cognitive architecture ACT-R (Anderson, 2007).

We provide a discussion of dual-process theory and moral judgments as a background to this research, and present previous cognitive computational modeling work on this theory. We then describe previous research using ACT-R and connect it to the model proposed in this research.

Related Work

Dual-process theory explains that humans have two different modes of thinking: fast and slow thinking. Fast thinking is referred to as System 1 and slow thinking as System 2, with the former being unconscious, intuitive, and impulsive based on heuristics and the latter being deliberative which is more conscious and burdensome to process. It is said that System 1 is the main processor, with System 2 intervening when necessary to make more complex decisions (Greene, 2015). System 1 and System 2 are not clearly distinct; rather, a spectrum exists between the two modes, with System 1 requiring less "effort" and System 2 requiring more (Conway-Smith and West, 2015).

Studies of analogical reasoning also have found reasoning based on superficial similarities called availability heuristics. In the past, there have been several studies on decision-making by analogy for issues where political judgment is at stake (Spellman and Holyoak, 1992). There is also a model

for reasoning about social problems by retrieval of similar cases (Blanchette and Dunbar, 2001).

These analogical models of political judgment do not consider the dynamic process of switching between System 1 and System 2. The definitions of superficial and structural similarity are also based on hand-coding, which makes the generality of the models questionable. To overcome these concerns, an integrative model that includes a chronological process is needed. An integrative model of cognition is achieved through the concept of cognitive architecture, of which ACT-R is a representative example.

In an example study on availability heuristics using ACT-R, Schooler and Hertwig (2005) developed the model to judge the size of a city's population. However, their model does not explain the switching between System 1 and System 2 in moral judgments. Therefore, in this research, we use ACT-R to construct a model that addresses moral decisions considering the dynamic changes of System 1 and System 2.

Case Representation

As in the related research of Schooler and Hertwig, a news corpus was used for the cases to be considered in the model. Our research uses news articles from the Livedoor News Corpus as the model's pre-stored cases in declarative memory. This corpus consists of several news sites having varied characteristics. We believe that these can be used to model the cultural background of individuals making moral judgments.

In coding the cases, sentiment and similarity to the problem statement were given based on the availability heuristics. Sentiment and similarity were each calculated using a method with distributed language models.

For the sentimental component, we used the Natural Language API by Google Cloud. The Natural Language API combines score, which indicates the positive and negative sentiments recognized from the sentence, and magnitude, which indicates the amount of sentimental content, to represent the sentimental elements of the entire sentence.

Sentence BERT (Reimers and Gurevych, 2019) was used to calculate the similarity between problem sentences and Livedoor news article titles.

Prototype Modeling

The design of the prototype model constructed in this research is such that System 2 intervenes in the decision-making process by System 1 as shown in Figure 1. The model repeatedly reads the given problem sentences in sequence, and for each sentence, it retrieves a memorized news article

that the sentence currently being read reminds it of. This retrieval process is governed by the chunk activation mechanism of ACT-R. The news articles are stored in the declarative memory module as chunks, and each chunk holds an activation. In ACT-R, the activation of a chunk is calculated as the summation of the relevance to the current situation (similarity) and the utilities estimated from experience (frequency of using the chunk). Thus, each time the model reads a sentence, it retrieves a story similar to the trolley problem, and the retrieved story is gradually converged until the model reaches the last sentence.

When the last sentence is read, it makes decisions according to the acquired memory sentiment: if POSITIVE, it chooses to push the target; if NEGATIVE or NEUTRAL, it chooses not to push the target, and if MIXED, it chooses to rethink.

We consider that the above process is generally consistent with the known characteristic of human availability heuristics. According to Kahneman (2012), the thought process based on System 1 is determined by easily recalled memory and substantially influenced by emotion. In contrast, we consider there is room for intervention by System 2 when the retrieved sentiment is MIXED at the last sentence. At that time, the model decides to read the sentences again. The number of steps in decision-making increases by rethinking, which can be explained as taking more effort.

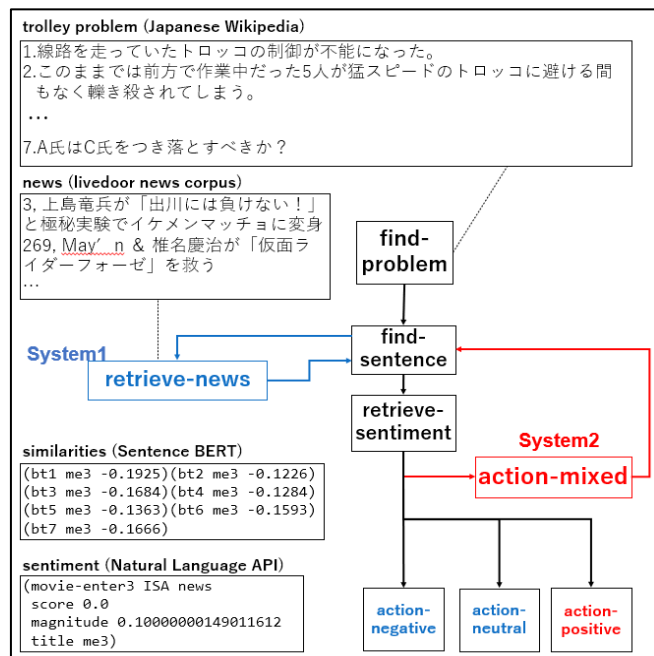


Figure 1. Moral judgement model based on two systems (System 1: blue line, System 2: red line). Japanese text of “Fat Man,” example articles from livedoor news corpus, and the retrieved chunks during the process are presented.

Results

Simulations were performed on nine models with pre-stored cases from multiple news sites using “Switch” and “Fat Man” as different expressions of the trolley problem. Greene (2015) pointed out that those expressions lead to

different decision outcomes in human experiments. For both expressions, simulations were run 1000 times. From the obtained result, we found that the sentiments of pre-stored news noticeably influenced decision-making. A crucial difference between the two expressions was also found when the news articles were collected from a site named MOVIE-ENTER (push choice in “Switch”: 0.463, push choice in “Fat Man”: 0.404). However, the effect of System 2 intervention could not be clearly observed.

Conclusion

In this research, we created a prototype model of the availability heuristic in moral judgments using distributed language models and ACT-R. Simulation of two simple trolley problem cases as tasks showed that the decision was strongly influenced by the sentiment of pre-stored cases in declarative memory. As the switch from System 1 to System 2 in the prototype model was based on sentiment, a redesign to a more deliberative version of System 2 is necessary to match the tendency of human moral judgments.

References

- Anderson J. (2007). How can the human mind occur in the physical universe?, Oxford University Press
- Blanchette I. and Dunbar K. (2001). Analogy use in naturalistic settings: The influence of audience, emotion, and goals, *Memory & Cognition*, 29(5), 730-735.
- Conway-Smith, B. and West, R. L. (2015). Clarifying System 1 & 2 through the Common Model of Cognition, *Proceedings of the 20th International Conference on Cognitive Modeling*, 40-45
- Evans, J. S. B., and Stanovich, K. E. (2013). Dual-process theories of higher cognition: Advancing the debate, *Perspectives on psychological science*, 8(3), 223-241.
- Greene J. D., Translated by Takeda M. (2015). *Moral Tribes: Emotion, Reason, and the Gap Between Us and Them*. Iwanami Shoten.
- Google, “Cloud Natural Language API”, Google Cloud, <https://cloud.google.com/natural-language/docs/reference/rest>
- Kahneman D., (2012). *Thinking, Fast and Slow*. Penguin
- Reimers N. and Gurevych I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.
- RONDHUIT. (2012). Livedoor news corpus. [Data set]. <https://www.rondhuit.com/download.html#ldcc>
- Schooler L. J. and Hertwig R. (2005). How forgetting aids heuristic inference. *Psychological review*, 112(3), 610.
- Spellman B. A. and Holyoak K. J. (1992). If Saddam is Hitler then who is George Bush? Analogical mapping between systems of social roles. *Journal of personality and social psychology*, 62(6), 913.
- Thomson J. J. (1984). The trolley problem. *Yale LJ*, 94, 1395.

Generating Body Images from Distributed Word Representation

Kosuke Sasaki (sasaki.kosuke.19@shizuoka.ac.jp),
Jumpei Nishikawa (nishikawa.jumpei.16@shizuoka.ac.jp),
Junya Morita (j-morita@inf.shizuoka.ac.jp)

Department of Informatics, Graduate School of Integrated Science and Technology, Shizuoka University,
3-5-1 Johoku, Naka-ku, Hamamatsu-shi, Shizuoka-ken, 432-8011 Japan

Abstract

This study proposes a method of generating body gestures from distributed representations of words. In the method, the size image for words is computed based on the index whose poles correspond to “small” and “large” word images. In addition, the size image of the words is physically implemented as robot gestures. The proposed methods were evaluated by two online surveys. Summarizing the results, the authors claim the potential of developing artifacts exchanging qualitative and quantitative aspects of word representations.

Keywords: Distributed representation of words; Gestures; Robotics

Introduction

Communications are mediated by symbolic or quantitative representations. Symbolic interfaces use discrete representations such as language and icons, while quantitative interfaces use physical quantities such as speech and movement. The two media are processed complementarily in human-human / human-machine communications.

The mechanism of exchange for these representations has been repeatedly discussed in the field of cognitive science. According to the reference frames theory proposed by Hawkins (2021), a continuous space exists behind each concept that stores knowledge and generates behavior, and this knowledge is recovered through language. Similar discussions are also made by Tversky (2019) claiming human language and thoughts that originally come from bodily experience made in a continuous time and space. In her discussion, the meanings of words are essentially embedded in our living physical world. Other similar discussions are also found in literature in the field of cognitive linguistics (e.g., Pinker, 2007).

A computational model of word meaning is needed to implement those mechanism. In natural language processing research, statistical analyses (Bag of Words, co-occurrence frequency, or principal component analysis from word vectors) have been applied to corpora derived from human language operations to capture the semantic relations between words. More recently, vector representations (distributed word representations) collapsed into the middle layer of a neural network have become the mainstream method for understanding words’ quantitative meanings (Bengio, Ducharme, & Vincent, 2000).

Based on the above background, this study proposes a method for generating body images using a conversion mech-

anism between discrete symbols and quantitative images. This method extracts the spatial knowledge structure underneath words from distributed representations constructed on a neural network. According to the aforementioned theoretical background, it can be assumed that there exist various quantitative images such as “size” and “speed” in the space where words are positioned (Grand, Blank, Pereira, & Fedorenko, 2018). In this study, we focus on “size images” to obtain quantitative representations of words related to their size; then, we generate iconic gestures (body images) for a robot by converting the “size image” into body actions.

This research aims to test the following hypothesis.

1. The spatial knowledge structure contained in the distributed word representation contains a “size image.”
2. Body image generated by the “size image” can recover human judgments.

As background leading to the purpose of this study, the next section reviews research on the computational method by extracting word meaning and gesture generation. Following this, we describe the proposed methods for generating body images of the concepts. The method contains two parts, each of which corresponds to the above two hypotheses; these two parts were verified by two experiments. The final section presents a summary of this study and future prospects.

Related Studies

The meaning of a word or concept can be modeled by several approaches. One approach is to write down the meanings of concepts circulating in society manually. Large-scale databases such as WordNet (Miller, 1995) and ConceptNet (Speer, Chin, & Havasi, 2017) have been developed so far. These databases define the normative knowledge structure in society.

Meanwhile, in recent years, there have been many approaches to capture the meaning of concepts statistically based on daily language use. Distributed representation of words (Bengio et al., 2000) considers a word as a point embedded in a vector space. In this framework, a word’s meaning is considered to be the relationship (distance or similarity) between words in the vector space. The underlying idea here is the distributional hypothesis that “words which are similar

in meaning occur in similar contexts.” (Rubenstein & Goodenough, 1965; Sahlgren, 2008)

Attempts have been made to extract words’ quantitative images by using distributed word representations. For example, Utsumi (2020) worked to reveal the internal knowledge embedded in distributed word representations. Using models of word distributions, he classified words into attributes and compared the results with word classifications obtained from human data. The results suggest that the vector space of distributed word representation captures important aspects of human knowledge. In particular, it was shown that abstract concepts are more deeply embedded in word distributions than in words with physical or bodily meanings associated with animate concepts.

In contrast, Grand et al. (2018) proposed a method for extracting context-dependent relations using distributed word representations. Context-dependent relations mean that, for example, the word “dog” has several different semantic features such as “size,” “intelligence,” and “danger,” depending on the context. This study shows that by projecting word vectors onto axes representing characteristics such as “size,” “intelligence,” and “danger,” it is possible to recover human judgments about categories and characteristics of various objects. Thus, it is suggested that human quantitative images of words are embedded in word-distributed representations. In other words, the quantitative meanings of concepts that humans have physically acquired are inherent in distributed word representations created from our daily language use.

The remaining question is about how such quantitative meanings work in communication settings. In this regard, many psychologists claim that bodily gestures represent humans’ internal thoughts (Goldin-Meadow, 1999; Kita, Alibali, & Chu, 2017). Other researchers have developed a method to make robots perform gestures in accordance with the emotions of the content of speech ((Lourens, Van Berkel, & Barakova, 2010). More recently, several studies are trying to generate co-speech gestures by end-to-end learning (Yoon et al., 2019; Liu et al., 2022).

However, we have not found any studies that link the quantitative meanings of concepts acquired using distributed word representations to the generation of iconic gestures (body images). Therefore, we propose a method for generating body images by acquiring the quantitative meaning of “size” concepts from the vector space of word distributions. Furthermore, by collecting human evaluations of generated body images, we verify the consistency with human perceptions.

Proposed method

Our proposed method includes the steps shown in Figure 1; the framework of each step is as follows.

Generating “size image”

To generate a “size image” according to the hypothesis in the first section, it is necessary to define an axis (index) of “size” in the vector space. Below, we show the process of extracting

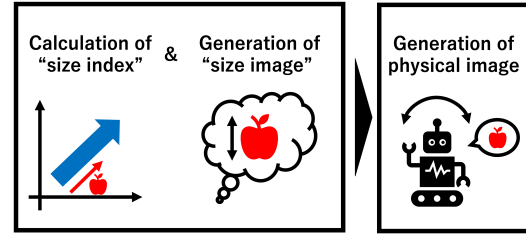


Figure 1: Flow of proposed method

a “size index” from the multidimensional vector space and then extracting a “size image” for an arbitrary word.

1. Composition of “size index”

The method for constructing word “size index” from the multidimensional vector space of distributed word representations follows Grand et al. (2018)’s study. In this method, the size index is taken to be a value on an axis consisting of “large” and “small” poles. To define the poles, we need to extract the coordinates of “large” and “small” in the distributed representation. However, in addition to their size-related meanings, these two words have extra meanings that derive from their adjectival status. To exclude such meaning unrelated to “size,” Grand et al. (2018) defined a set of synonyms that have the same role as “large” and “small” in the distributed word representation. Then, the polar coordinates are determined by computing the mean vector of these synonyms.

However, the above method proposed by Grand et al. (2018) has limitations in the arbitrariness of selecting the synonyms; they didn’t provide criteria of selecting the synonym. Our method overcomes this by automatically determining the set of synonyms for “large” and “small” with reference to an existing thesaurus. Words are usually polysemous and have multiple meanings. In a thesaurus, a set of synonyms for a word is defined as a synset for each of their meanings. From these synsets, we seek the combination of those that maximize the distance between the “large” and “small” words obtained from the human survey in order to determine the polar coordinates consistent with human perception.

In this step, we consider categories based on word abstraction. According to Tversky (2019) and others, the meaning of a concept is originally composed of the human movement. However, as shown by Utsumi (2020), physical quantities are not expected to be strongly embedded in distributed word representations composed of socially published documents. Therefore, the scale of the “size index” has the possibility to be changed by the categories the word belongs.

2. Composition of “size image”

The cosine similarity of the input word vectors is calculated from the “size index”, and the value is used as the

“size image” of the input word. In our proposed method, the larger this value is, the larger the word is assumed.

Physicalizing body images

In this step, the body image (iconic gesture) is generated according to the “size image” calculated for each word. The parameter scaling and the procedure are shown below.

1. Setting large and small postures

The “size index” is mapped to the postures composed of body parts. For this purpose, we determine the body posture corresponding to the smallest and largest words recognized by humans. Using this posture as a reference (the image of the smallest word is 0 and the image of the largest word is 1), the “size image” of each word can be positioned in the range of 0 to 1.

2. Calculation of parameters at each joint

The above scaling is applied to the joint angles of each joint that constitutes the posture.

3. Generating the body image

A gesture is generated based on the values obtained by step 2. This gesture is assumed to be made simultaneously with the utterance of the word.

Experiment 1: Generation of “size image”

We test the first hypothesis presented in the first section by generating “size images” using the proposed method. A questionnaire survey was conducted to extract human perceptions of word size, and to generate a “size index”.

Method

Models In this study, we used the Japanese Wikipedia Entity Vector (Suzuki, Matsuda, Sekine, Okada, & Inui, 2016), which we call JWikiEntVec in this paper, as the distributed representation model. This model is built by word2vec (Mikolov, Chen, Corrado, & Dean, 2013). We consider that word2vec is more suitable for this research than more recently developed models such as BERT (Devlin, Chang, Lee, & Toutanova, 2018) because it is a faster method that can be applied in real-time settings.

In addition, we used the Japanese WordNet (Bond, Isahara, Uchimoto, Kuribayashi, & Kanzaki, 2009) for synonym selection. This thesaurus contains 28 synsets for “large” and 14 synsets for “small”. Words not included in JWikiEntVec and synsets with no synonyms were excluded from the later analysis. As a result, we obtained 23 synsets for “large” and 13 synsets for “small.” The “size index” was calculated for the combinations of these synsets ($23 \times 13 = 299$).

Survey To define a “size index” consistent with human perception, we conducted an online survey to obtain the set of “large” and “small” words recognized by humans. One hundred respondents were recruited from Lancers, a Japanese crowdsourcing site (reward: 55 yen). The participants were

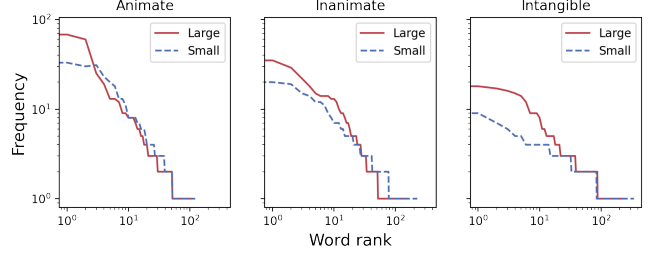


Figure 2: Rank frequency of words obtained by the survey

Table 1: Responses obtained for each question (Top five words)

(a) Animate			
Large		Small	
Word	Frequency	Word	Frequency
Elephant	85	Ant	74
Whale	68	Daphnia	33
Giraffe	60	Mosquito	31
Bear	25	Tick	30
Hippopotamus	19	Fleas	23
(b) Inanimate object			
Large		Small	
Word	Frequency	Word	Frequency
Tokyo Sky Tree	45	Sand	26
Mt. Fuji	35	Beads	20
Tokyo Tower	29	Needle	26
Everest	22	Microchip	15
Pyramid	18	Screw	14
(c) Intangible concept			
Large		Small	
Word	Frequency	Word	Frequency
Space	36	Mind	13
Love	18	Jealousy	9
Dream	17	Envy	7
Mind	16	Vanity	6
Sea	15	Point	5

*Words are translated from Japanese

asked to write down five “large” and “small” words for animate, inanimate, and intangible concepts. Thus, each participant provided totally 30 words (five words \times two sizes [small vs. large] \times three categories [animate vs. inanimate vs. intangible]) in this survey.

Results

Overview of responses The total number of words obtained from this survey was 937, of which 828 were included in JWikiEntVec. Of these words, 62, 117, and 188 were large animate, inanimate, and intangible concepts, respectively while 83, 150, and 228 were small animate, inanimate, and intangible concepts, respectively. Figure 2 shows the rank frequencies of these words as a two-tailed logarithmic graph. Table 1 shows the top five words for each combination of size and categories.

Extraction of “size index” From the 299 “size index” (corresponding to all combinations of synsets), we selected the

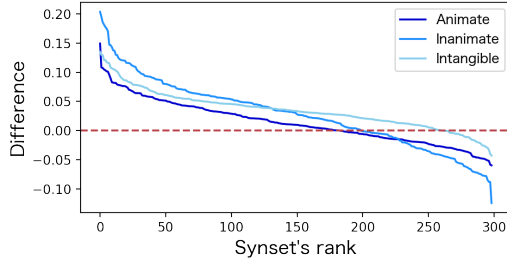


Figure 3: Distribution of “size image” difference

Table 2: Top and bottom combinations of synsets ordered by the “size image” difference

Rank	“large” synset	“small” synset
1	larger-than-life	flyspeck
2	outstanding	flyspeck
3	outstanding	softend
297	sizable	immature
298	large	immature
299	sizable	small

index that is most consistent with human image. In this selection, the average value of the “size image” of “large” words and the average value of the “size image” of “small” words were calculated for each category, and the difference between them were obtained. Figure 3 shows the differences calculated for the 299 indicators. The horizontal axis of this figure corresponds to the combination of synsets ordered by rank. From this figure, it can be seen that the differences in the “size image” of some of the combinations are large for all categories. Overall, there are many combinations in which the difference of “size image” is larger than 0 (the red dotted line), indicating that the size index is calculated for more than half of the synset combinations is consistent with human image.

Table 2 shows the top three and the bottom three combinations when the rankings of the three categories are summed. From those lists, we can find several instances of “outstanding” for “large” and “flyspeck” for “small” in the top combinations. Contrary in the bottom combination, “sizable” for “large” and “immature” for “small” were appeared several times. From those, we can see the tendencies in which the synsets with abstract meanings tended to be placed higher, while those with physical meanings tended to be placed lower.

Table 3 shows synonyms included in the top synsets. Among them, the words “tiny (ti-kko-i)” and “very small (go-ku-ti-i-sa-i)” were unlearned in JWikiEntVec, so they were excluded from the calculation of the “size index” in this study.

Validation of the “size image” We examine the “size image” calculated by the “size index” using “larger-than-life” and “flyspeck” extracted by the above analysis. Figure 4 shows the average “size image” of the words in each category obtained from the survey. From the figure, it can be seen that

Table 3: Word list for “larger-than-life” and “flyspeck”. Words in parentheses indicate original Japanese words in Hepburn romanization

word	“large”	“small”
synset	larger-than-life	flyspeck
meaning	very impressive	very small
Synonym1	magnificent (so-da-i)	tiny (ti-ppo-ke)
Synonym2	large scale (da-i-ki-bo)	minute (b-i-syo)
Synonym3		tiny (ti-kko-i)
Synonym4		very small (go-ku-ti-i-sa-i)

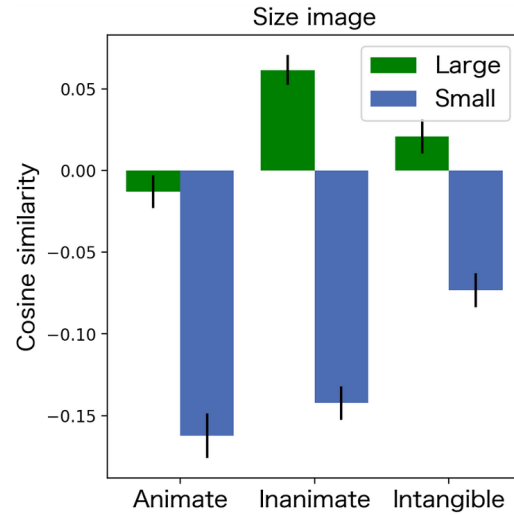


Figure 4: Mean value of the cosine similarity (size image) between the word group and the “size index” obtained from the questionnaire (error bars are standard errors)

the “size image” of the “large” words exceeds the “size image” of the “small” words in all categories. To confirm this impression, we conducted a two-way [categories (animate vs. inanimate vs. intangible) size (large vs. small)] analysis of variance (ANOVA) with “size image (cosine similarity)” as the dependent variable. The results showed the significant interaction between the factors ($F(2, 824) = 8.84, p < .001$) with a simple main effect of the size for animate ($F(1, 824) = 65.60, p < .001$), inanimate ($F(1, 824) = 122.14, p < .001$) and intangible ($F(1, 824) = 26.06, p < .001$).

The above results confirm that the size image of “large” words is larger than that of “small” words in all categories. In other words, the size index calculated using “larger-than-life” and “flyspeck” can discriminate the size of words consistent with human perception.

Discussion

From the above, we were able to define an axis in the vector space of the distributed word representation that distinguishes between large and small words that are recognized by humans. In other words, the first hypothesis of this study, that a quantitative image of “size” is embedded in the vector space of the distributed word representation, is supported.

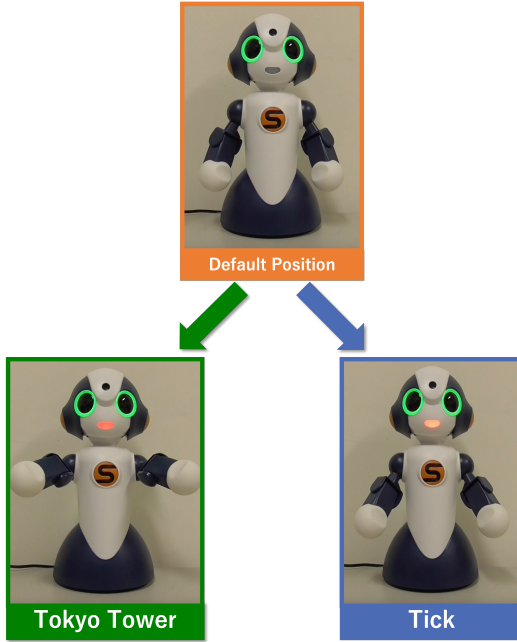


Figure 5: Examples of Sota gestures (left: “Tokyo Tower”, right: “tick”)

However, this result is not surprising, since it only defines the axis that maximizes the difference between large and small words obtained from the questionnaire. An interesting result is that synsets with abstract meanings were observed at the top of Figure 3, while synsets with psychical meanings were observed at the bottom. This result supports the discussion that the vector space of word-distributed representations is biased toward abstract meanings (Utsumi, 2020). This bias also seems to be consistent with another trend in Figure 4 showing larger size images in intangible (abstract) words compared with animate (physical) words.

Experiment 2: Physicalization of “body image”

A body image (iconic gesture) of the robot is generated in line with the “size image” generated in the previous section and evaluated. Through this examination, the second hypothesis presented in the first section is verified.

Method

Equipment and Materials For the physicalization of the body image (embodiment), we use Sota, a small communication robot manufactured by Vstone¹. Sota’s body movements are controlled by nine joints (one torso, three necks, two shoulders, and two arms joint). By controlling the angle and speed of these joints, Sota can generate various gestures. In addition, Sota has a speech function and can speak any words while simultaneously displaying gestures.

In this study, the parameters of the arm and shoulder joints were instantiated by “size image” to generate gestures. Sota’s

Table 4: Maximum and minimum values of parameters for “size image” and each joint

(a) Proposed embodiment		
	Maximum	Minimum
Size image	-0.490	0.310
Shoulder angle	-70	27
Arm angle	20	-25
(b) Reversal embodiment		
	Maximum	Minimum
Size image	-0.490	0.310
Shoulder angle	27	-70
Arm angle	-25	20

default posture is the one shown in the upper image in Figure 5, with the shoulders down and the arms slightly bent. From this state, the parameters of the arm and shoulder joints are changed to generate gestures that correspond to the size of the word. When the “size image” of the word is large, the shoulders are raised and the arms are extended, and when it is small, neither the arm nor the shoulders move much.

The bottom images of Figure 5 shows examples of the generated body images. The lower-left image shows the gesture when the user says “Tokyo Tower” (large inanimate concept), and the lower-right image shows the gesture when the user says “tick” (small animate concept).

Table 4 (a) shows the maximum and minimum values of the “size image” calculated for all words obtained from the survey and the matched parameters of the shoulder and arm joints. These values were used to scale the “size image” in the range 0 to 1.

Design and measures A body image was generated for the 30 words in Table 1, and generated gestures for each word were recorded as movies (about 4 s videos). In this experiment, the participants were asked to rate the naturalness of the correspondence between the robot’s movements and the words it spoke on a 5-point scale (1 = not at all natural; 5 = very natural). The “naturalness” here assume a situation where the user is interacting with the robot. Specifically, the user is asked to rate whether or not they feel that the robot understands the meaning of the words as a human would.

The proposed method was evaluated by comparing the reversal embodiment condition, where the minimum value for each joint is taken when the scaled value is 1, and the maximum value for each joint is taken when the scaled value is 0, as opposed to the scaling in the proposed condition (Table 4 (b)). This results in the arms and shoulders being moved less when the “size image” value is large and the arms being raised and opened wider when the “size image” value is small.

To prevent order effects, three question forms were prepared with randomly sorted 60 videos. The survey participants were equally assigned to these forms in the order in which they accessed them. In between the evaluations of the 60 videos, two dummy questions were asked, in which the

¹<https://www.vstone.co.jp/english/index.html>

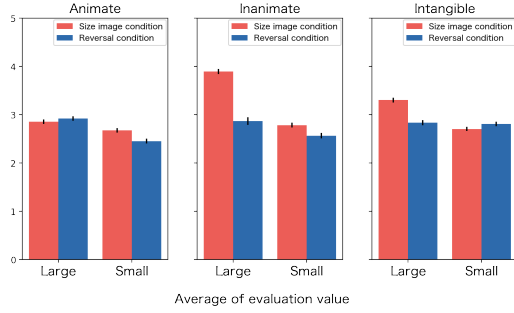


Figure 6: Average rating value per category

participants were asked to answer a specified number verbalized by the robots.

Participants and procedure Two hundred participants (62 females) who were contacted via Lancers evaluated the videos after reading the instructions provided on the request screen (reward: 110 yen). The instructions explained the evaluation procedure, the definition of naturalness, and the obligation to answer dummy questions. After agreeing to the instructions, participants answered 62 questions on Google Forms.

Results

Two participants with multiple responses and two participants with incorrect answers to the dummy questions were deleted. For each of the 196 participants, the six averages of the evaluation values on body images (sizes \times conditions) were calculated (Figure 6). This value was used as the dependent variable in a three-factor [embodiment (proposed vs. reversal) \times size (large vs. small) \times category (animate vs. inanimate vs. intangible)] ANOVA. The analysis showed a main effect of embodiment ($F(1, 195) = 93.31, p < .001$), confirming that the proposed embodiment was evaluated more naturally than the reverse condition overall. However, a second-order interaction ($F(2, 390) = 86.97, p < .001$) was also significant, indicating that factors of size and categories influenced this effect.

To examine the differences in the effects of embodiment between each combination of size and categories, three two-way [embodiment (proposed vs. reversal) \times size (large vs. small)] ANOVAs were conducted for the categories (animate, inanimate, intangible). The results showed significant interactions between the factors in all analyses (animate: $F(1, 195) = 32.01, p < .001$, inanimate: $F(1, 195) = 89.14, p < .001$, intangible: $F(1, 195) = 103.74, p < .001$).

Table 5 illustrates the simple main effects of embodiment (proposed - reversal) for each combination of size and categories, showing that the naturalness of the proposed condition exceeded that of the reversal condition for small animate words, large and small inanimate words, and large intangible words. In particular, the large inanimate and the large intangible words had a medium or large effect size, indicating that the gestures were generated naturally in those conditions.

Table 5: Simple main effects of size image condition and reversal condition

Category	Size	Difference	<i>F</i>	<i>p</i>	<i>d</i>
Animate	Large	-0.067	4.85	0.029	0.103
	Small	0.222	28.19	<.001	0.328
Inanimate	Large	1.031	157.61	<.001	1.107
	Small	0.219	15.80	<.001	0.299
Intangible	Large	0.466	70.16	<.001	0.652
	Small	-0.104	7.13	0.008	0.165

However, the naturalness of the reversal condition exceeded the naturalness of the proposed condition for large animate and small intangible words with small effect sizes.

Discussion

The above results show that the proposed embodiment outperformed the reversal embodiment in the overall and several categories' naturalness evaluation. These results partially support the second hypothesis of this study (that body images generated by "size-images" can recover human judgments). However, contrary to our hypothesis, several reversal embodiment conditions may have been evaluated as more natural than the corresponding proposed conditions. This inconsistency may be due to the aforementioned bias in the categories of meanings embedded in the word-distributed representations. In particular, unbalanced distributions of "size images" presented in Figure 4 seem to explain the inconsistent result in the large animate and small intangible conditions.

Conclusion

In this study, we proposed a method for generating body images related to "size". Through two experiments, we partially confirmed that it is possible to generate "size images" consistent with human perception from distributed word representations, and that the body images generated from the "size images" can recover human size images.

One limitation of our results is that we were unable to generate body images consistent with human perception in some categories. According to Utsumi (2020), distributed representations constructed from corpus has a bias toward high granularity in abstract concepts rather than embodied or physical concepts. Accommodating this property will require improving the normalization method by distinguishing categories of word abstractness.

The other future work is setting other baselines. In this study, the reversal condition was only set as a baseline. In the future, it will be necessary to examine the effects compared for conditions in each step of the method proposed in this study by setting a contrast condition. For example, the body image generated by the "size index" using the combination of lower-level synsets in Figure 4 can be set as a baseline condition.

In addition to the above issues, in the future, we will work on the generation of body images using not only "size" but also "sharpness," "speed," and various other index. We believe that this will contribute to both the understanding of our

cognition and the development of “an advanced human interface that realizes smooth interaction with humans, equipped with a conversion mechanism between human symbols and quantities.”

References

- Bengio, Y., Ducharme, R., & Vincent, P. (2000). A neural probabilistic language model. *Advances in neural information processing systems*, 13.
- Bond, F., Isahara, H., Uchimoto, K., Kuribayashi, T., & Kanzaki, K. (2009). Extending the Japanese WordNet. In *Proc. 15th annual meeting of the association for natural language processing* (pp. 80–83).
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Goldin-Meadow, S. (1999). The role of gesture in communication and thinking. *Trends in Cognitive Sciences*, 3(11), 419–429.
- Grand, G., Blank, I. A., Pereira, F., & Fedorenko, E. (2018). Semantic projection: recovering human knowledge of multiple, distinct object features from word embeddings. *arXiv preprint arXiv:1802.01241*.
- Hawkins, J. (2021). *A thousand brains: A new theory of intelligence*. Basic Books.
- Kita, S., Alibali, M. W., & Chu, M. (2017). How do gestures influence thinking and speaking? the gesture-for-conceptualization hypothesis. *Psychological review*, 124(3), 245.
- Liu, X., Wu, Q., Zhou, H., Xu, Y., Qian, R., Lin, X., ... Zhou, B. (2022, June). Learning hierarchical cross-modal association for co-speech gesture generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (cvpr)* (p. 10462–10472).
- Lourens, T., Van Berckel, R., & Barakova, E. (2010). Communicating emotions and mental states to robots in a real time parallel framework using laban movement analysis. *Robotics and Autonomous Systems*, 58(12), 1256–1265.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11), 39–41.
- Pinker, S. (2007). *The stuff of thought: Language as a window into human nature*. Penguin.
- Rubenstein, H., & Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, 8(10), 627–633.
- Sahlgren, M. (2008). The distributional hypothesis. *Italian Journal of Linguistics*, 20, 33–53.
- Speer, R., Chin, J., & Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI Conference on Artificial Intelligence* (pp. 4444–4451).
- Suzuki, M., Matsuda, K., Sekine, S., Okada, N., & Inui, K. (2016). Wikipedia kiji ni taisuru kakucyo koyu hyogen label no tajyu fuyo (in japanese). In *Proceedings of the twenty-second annual meeting of the association for natural language processing* (pp. 797–800).
- Tversky, B. (2019). *Mind in motion: How action shapes thought*. Hachette UK.
- Utsumi, A. (2020). Exploring what is encoded in distributional word vectors: A neurobiologically motivated analysis. *Cognitive Science*, 44(6), e12844.
- Yoon, Y., Ko, W.-R., Jang, M., Lee, J., Kim, J., & Lee, G. (2019). Robots learn social skills: End-to-end learning of co-speech gesture generation for humanoid robots. In *2019 international conference on robotics and automation (icra)* (pp. 4303–4309).

Relative Attention Across Features Predicts That Common Features Increase Geometric Similarity

Florian I. Seitz (florian.seitz@unibas.ch)

Department of Psychology, Missionsstrasse 62A
Basel, Switzerland

Abstract

The human mind relies on similarity to organize the world around it. A geometric approach to similarity, which assumes that two objects' similarity decreases with the sum of their feature value differences, has been particularly influential. Yet, geometric similarities are claimed to ignore common features, which is inconsistent with human similarity judgments that increase the more common features the objects share (the *common features effect*). This paper shows that a relative attention mechanism, as it is implemented in current cognitive models based on geometric similarities, can naturally predict the common features effect by weighting each feature value difference with the share of attention allocated to the feature. Additional common features draw away attention from the already present features, which entails that the objects' differences with respect to already present features receive less weight, resulting in a higher similarity. The ability of the geometric similarity theory with relative attention to predict the common features effect is illustrated for data from Gati and Tversky (1984) and for data from a new pairwise similarity judgment experiment.

Keywords: Similarity; Attention; Common features; Computational modeling

Introduction

The human mind organizes the world around it based on similarity. People can group objects by similarity because objects with similar features are likely to be similar in other variables of interest as well (Goldstone & Son, 2012). For instance, people can use the similarity of different animals in terms of color, shape, and other features to classify them into species or judge their toxicity. In other words, similarity is fundamental for inferences such as categorizations (Kruschke, 2008; Nosofsky, 1986) and judgments (Albrecht, Hoffmann, Pleskac, Rieskamp, & von Helversen, 2020; Juslin, Jones, Olsson, & Winman, 2003), and understanding psychological similarity provides direct access to understanding the cognitive system. This article counters a criticism of a widespread psychological similarity theory—namely, that common features do not affect the geometric similarity between objects—by showing that current geometric similarity models can consider common features due to a relative attention mechanism.

With decades of psychological research conducted, many theories of how people compute similarity have emerged (e.g., Tversky, 1977; for an overview, see Goldstone & Son, 2012). Particularly influential is the *geometric similarity theory*, which assumes that people compute the similarity between objects as a function of their distance in a perceptual space, in which each dimension corresponds to an object feature (Nosofsky, 1986; Shepard, 1987). Specifically, the geometric similarity theory represents each object as a point in

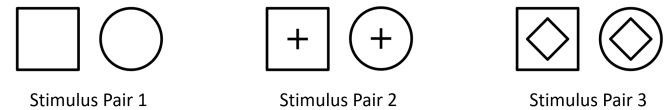


Figure 1: *Common Features*. Shown are three stimulus pairs differing in their outer shape. In stimulus pairs 2 and 3, there is an additional common feature (a cross or a diamond), that might increase the similarity of the two stimuli.

the perceptual space, with the object's feature values corresponding to the point's coordinates. Objects with small distances in the perceptual space are assumed to be similar; objects with large distances are assumed to be dissimilar. In line with this idea, past research has found that people's similarity judgments get smaller with larger feature value differences between objects (Gati & Tversky, 1984; Navarro & Lee, 2004; Ritov, Gati, & Tversky, 1990). For instance, in Seitz, von Helversen, Albrecht, Rieskamp, and Jarecki (2023) participants rated the pairwise similarity of objects with three multivalued features on a visual slider ranging from "completely different" (coded as 0) to "identical" (coded as 1). Compared to objects with one common and two minimally differing features, participants' similarity ratings decreased by .11 when the objects also differed on the third feature, and by .30 when the objects differed maximally on the two features. This finding was well described by a geometric similarity model (Seitz, von Helversen, et al., 2023), suggesting that similarity can often be approached with geometric distances (see also Lee & Navarro, 2002; Navarro & Lee, 2002b).

Often, objects differ from each other only on some features but share the same value on the remaining features. Figure 1 illustrates this for geometric figures. In each stimulus pair, the figures differ in their outer shape; in pairs 2 and 3, however, there is an additional common feature in the form of a cross or a diamond. Such common features increase the objects' psychological similarity (the *common features effect*; Falkowski, Sidoruk, Olszewska, & Jabłońska, 2021; Ritov et al., 1990; Tversky, 1977; Young & Wasserman, 2002). In turn, geometric similarities are claimed to consider only the differing features between objects, but not the common features (Kruschke, 2008). The underlying intuition is that features that do not differ between two objects do not affect the objects' distance in perceptual space. Subsequently, it has been ar-

gued that geometric similarities cannot account for the common features effect (Verguts, Ameel, & Storms, 2004; Young & Wasserman, 2002), which is one reason for the emergence of other theories of psychological similarity (e.g., Tversky, 1977). In this paper, I will show that geometric similarities can predict the common features effect by means of a relative attention mechanism, that is implemented in many current geometric similarity models (e.g., Nosofsky, 2011).

The Effect of Common and Differing Features

While differing features decrease psychological similarity, common features typically increase it (Gati & Tversky, 1984; Lee & Navarro, 2002; Ritov et al., 1990; Young & Wasserman, 2002). In Gati and Tversky (1984), participants rated the pairwise similarity of verbal stimuli (e.g., descriptions of people) or visual stimuli (e.g., schematic face drawings) varying on up to three binary features. The authors found that an additional common feature increased the similarity ratings, whereas an additional differing feature decreased them. For verbal objects, adding a common feature affected similarity more ($M = .24$ across studies) than adding a differing feature ($M = .09$)¹. For visual objects, in turn, the effect of adding a differing feature ($M = .20$) exceeded the one of adding a common feature ($M = .08$). The result that common features affect similarity more than differing features for verbal objects but less for visual objects was replicated in Ritov et al. (1990), who found that one reason for this discrepancy is the higher cohesiveness of visual relative to verbal stimuli.

Navarro and Lee (2002b, 2004) showed that some features can act as purely common features, in the sense that they increase the similarity of objects that share the feature, but do not affect the similarity of objects that do not share the feature (e.g., the presence of twinhood increases the similarity of two people, while its absence may not decrease their similarity). In contrast, other features can act as purely differing features, decreasing the similarity of objects that do not share the feature but not affecting the similarity of objects that share the feature (e.g., a gender mismatch decreases two people's similarity, while a gender match might leave their similarity unaffected). In line with this idea, the authors found that two schematic faces that shared the feature of being unremarkable had a similarity rating .10 higher than two faces that were not both unremarkable—presumably, because unremarkableness acts as a purely common feature that affects psychological similarity only if it is present in both objects of comparison.

Common and differing features not only determine similarity but may also have an effect on similarity-based cognitive inferences. In Young and Wasserman (2002), participants learned to predict the occurrence of an outcome for objects with one or two features. Among the objects that differed in their outcome occurrence, there were pairs that differed on a single feature, pairs that differed on one feature

and matched on the second feature, and pairs that differed on two features. Participants learned to predict the outcome occurrence best for the pairs with one differing feature (80% accuracy), followed by the pairs with two differing features (63% accuracy) and the pairs with one differing and one common feature (57% accuracy). In other words, an additional common feature between two objects lowered participants' predictive accuracy substantially, probably by increasing the objects' similarity and thereby the difficulty of the prediction task (for contrasting results, in which an additional common feature did not affect learning, see Thorwart, Glautier, & Lachnit, 2010).

Explaining the Common Features Effect With Geometric Similarity

The common features effect was considered evidence against the geometric similarity theory (Tversky, 1977; Verguts et al., 2004; Young & Wasserman, 2002). The geometric similarity between objects depends on the objects' distance in a perceptual space, which increases with larger feature value differences. A common feature between two objects entails a difference of 0 for this feature, which will leave the distance and hence the similarity between the objects unaffected. Accordingly, cognitive models based on geometric similarity (e.g., the generalized context model, Nosofsky, 1986) are in general deemed to consider differing features only but to be insensitive to feature matches (e.g., Kruschke, 2008, Table 9.1).

In the following, I will show how the geometric similarity theory can account for the common features effect. Central to the explanation is the concept of relative attention to the object features, which describes that people distribute their attention across features when computing the similarity between objects (e.g., Nosofsky, 1986). Specifically, during similarity computation each feature value difference is weighted according to the share of attention allocated to the feature. Additional features cannot increase the total amount of attention available but draw attention away from other, already present features. In case the objects match on the additional feature, the additional difference of 0 does not affect the objects' distance nor similarity. However, if people attend to the additional common feature, the preexisting feature value differences are weighted with less attention, reducing the objects' distance in space and increasing their similarity.

Figure 2 illustrates how relative attention can explain the common features effect in a geometric similarity framework. The x-axis and y-axis show the number of common and differing features, respectively, for object pairs with binary features. A common feature is coded as a feature value difference of 0, and a differing feature as a feature value difference of 1. For each combination of the number of common and differing features, the predicted geometric similarity, resulting from a formal model detailed below, is shown. In the left panel, the model predictions are made without relative attention; in the right panel, the model predictions are made by weighting each feature value difference with the share of at-

¹The similarities in Gati and Tversky (1984) range from 1 to 20. In this paper, all similarities are rescaled to lie between 0 and 1 for better comparability.

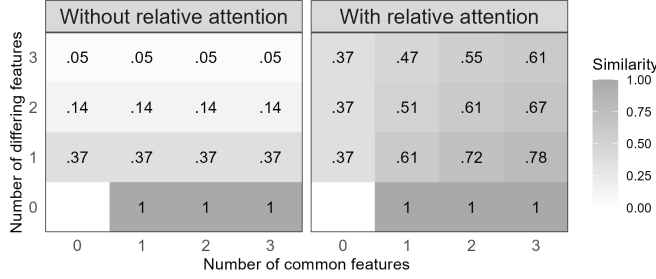


Figure 2: *Effect of common and differing features on geometric similarity.* Each cell shows for an object pair with a specific number of common features n_c and differing features n_d the similarity predicted by a geometric similarity model (see below for a formal description). Without relative attention (left), the common features do not affect similarity. With relative attention (right; in the example, the same share of attention $w_n = 1/N$ was allocated to each feature $n \in \{1, \dots, N\}$, with $N = n_c + n_d$), the similarity increases with the number of common features as soon as there is one differing feature. The predictions stem from Eq. 1 with $c = 1$, $r = 1$, and $p = 1$.

tention allocated to the feature. For the example in Figure 2, the same share of attention is allocated to each feature. In the case of one common and one differing feature, both features are weighted with an attention weight of $1/2$; in the case of two common and one differing features, each feature is weighted with an attention weight of $1/3$. The figure shows that with more common features the predicted geometric similarity remains constant if relative attention is omitted (in line with Kruschke, 2008), but increases when relative attention is included (and when there is at least one differing feature).

The idea of relative attention is not new to geometric similarities. Many formal models of geometric similarity weight the feature value differences between objects by the share of attention allocated to the feature to predict people’s similarity judgments (e.g., Seitz, von Helversen, et al., 2023), numerical judgments (e.g., Albrecht et al., 2020; Hoffmann, von Helversen, & Rieskamp, 2016), and categorizations (e.g., Nosofsky, 1986; Seitz, Jarecki, & Rieskamp, 2023; Smith & Minda, 1998). The attention weights may reflect the features’ importance for an inference task, where more attention is devoted to features that are more task-relevant (Nosofsky, 1986, 2011).

Formal Framework: Relative Attention in a Geometric Similarity Model

A widespread formalization of the geometric similarity theory, shown to describe people’s similarity-based inferences well (e.g., Nosofsky, 1986; Seitz, von Helversen, et al., 2023), computes the feature value differences between objects and transforms the weighted sum of these differences (the distance) into a psychological similarity by means of a negative exponential function (Shepard’s universal law of generalization; Shepard, 1987). Given two objects I and J

with feature values $I = (i_1, i_2, \dots, i_N)$ and $J = (j_1, j_2, \dots, j_N)$, the objects’ geometric similarity s_{IJ} is defined as

$$s_{IJ} = \exp \left(-c \cdot \left[\sum_{n=1}^N w_n \cdot |i_n - j_n|^r \right]^{\frac{p}{r}} \right), \quad (1)$$

where $|i_n - j_n|$ is the absolute feature value difference on feature n between objects I and J . For binary features, as are typically used to assess the effect of common and differing features on similarity (e.g., Gati & Tversky, 1984), $|i_n - j_n| = 0$ in case of a common feature and $|i_n - j_n| = 1$ in case of a differing feature. The geometric similarity defined in Eq. 1 has four parameters, highlighted in red, that can be estimated from (individual) participants’ data: Of particular importance for this paper is the relative attention w_n to feature n (with $0 \leq w_n \leq 1$ and $\sum_n w_n = 1$), which models how people distribute their attention across features and which can thereby predict that additional common features increase similarity.

The remaining three parameters are the sensitivity c to the two objects’ distance (with $c \geq 0$), the norm r of the distance (with $r \geq 1$, where $r = 1$ produces the city-block distance used for objects with separable features, and $r = 2$ produces the Euclidean distance used for objects with integral features), and the exponent p relating distance to similarity ($p = 1$ produces an exponential relation used for well-discriminable objects, and $p = 2$ produces a Gaussian relation used for highly-confusable objects); for more detailed explanation on these parameters, see Nosofsky (2011).

The geometric similarity model of Eq. 1 implements relative attention by constraining the attention weights w_n to sum up to 1. Mathematically, this constraint is necessary to make the model identifiable. To see why, transform

$$\begin{aligned} s_{IJ} &= \exp \left(-c \cdot \left[\sum_{n=1}^N w_n \cdot |i_n - j_n|^r \right]^{\frac{p}{r}} \right) \\ &= \exp \left(-c \cdot \frac{x}{x} \cdot \left[\sum_{n=1}^N w_n \cdot |i_n - j_n|^r \right]^{\frac{p}{r}} \right) \\ &= \exp \left(-c \cdot x \cdot \left[\sum_{n=1}^N \frac{w_n}{x^{\frac{r}{p}}} \cdot |i_n - j_n|^r \right]^{\frac{p}{r}} \right) \\ &= \exp \left(-c' \cdot \left[\sum_{n=1}^N w'_n \cdot |i_n - j_n|^r \right]^{\frac{p}{r}} \right), \end{aligned}$$

where $c' = c \cdot x$ and $w'_n = w_n / x^{\frac{r}{p}}$ for each $n \in 1, \dots, N$. In other words, different parametrizations of the geometric similarity model can be observationally equivalent to each other. In the case of $r = p$, multiplying c and dividing w_n by the same constant x lead to observationally equivalent model versions. Constraining the attention weights w_n to sum up to 1 is therefore necessary to make the model identifiable, because if $\sum_n w_n = 1$, then $\sum_n w_n / x^{\frac{r}{p}} \neq 1$ for any $x \neq 1$.

Given that studies investigating the common features effect typically use well-discriminable objects with separable features (e.g., Gati & Tversky, 1984), I focus on the version with $r = 1$ and $p = 1$. In this case, the formula reduces to

$$s_{IJ} = \exp \left(-c \cdot \left[\sum_{n=1}^N w_n \cdot |i_n - j_n| \right] \right), \quad (2)$$

with sensitivity c and attention weights w_n as defined above.

An Illustrative Example

This section applies the geometric similarity model formalized above to pairwise similarity judgment data and illustrates how relative attention can predict the common features effect. I used the data reported in Gati and Tversky (1984), Experiment 10, Set B, in which participants judged the similarity between pairs of landscape drawings, differing in up to three features². Feature 1 was the background in the form of hills (denoted as p in Gati & Tversky, 1984) or mountains (q); this feature was substitutive, meaning that one of its values was present in each drawing. Features 2 and 3 were a cloud (x) and a house (y), respectively; both these features were additive, meaning that they could be present or absent in the drawing, independently of each other. Participants rated the similarity of various pairs of drawings, differing in the number of additive features included, on a 20-point Likert scale.

Table 1 shows the aggregate similarity judgments s_{IJ} , rescaled to the range of 0 to 1. On average, participants rated the similarity of two drawings that differed in their substitutive feature and had no additional features (p-q in Table 1) to be .16. Adding a common feature increased participants' similarity judgments to .23 (px-qx and py-qy). Table 1 also shows the predictions of the geometric similarity model defined in Eq. 2, setting $c = 1.78$ and $w_n = .17$ for each additive feature $n \in \{2, 3\}$ that is present in at least one of the drawings³. In other words, in this model setup, each additive feature draws .17 of attention as soon as it appears in one of the drawings, and the remaining attention is allocated to the omnipresent substitutive feature. In the pair of drawings in which no additive features are present (p-q), all attention is allocated to the substitutive feature, $w_1 = 1$, and a similarity of $\hat{s}_{IJ} = \exp(-1.78 \cdot 1) = .17$ results. Adding a common feature draws attention away from the substitutive feature, leading to $w_1 = 1 - .17 = .83$, $w_2 = .17$, and $\hat{s}_{IJ} = \exp(-1.78 \cdot [.83 \cdot 1 + .17 \cdot 0]) = .23$. Just as for the observed similarity judgments, the additional common feature increased the predicted geometric similarity. The remaining rows in Table 1 show that this model also predicts the other available aggregate similarity judgments well, suggesting that participants in Gati and Tversky (1984) might have based their similarity judgments on a geometric framework.

²Only the data aggregated across participants was available.

³These parameter values equal those resulting from fitting the model with log likelihood, assuming that each additive feature that is present in at least one of the two drawings draws a constant amount of attention from the substitutive feature.

Table 1: Applying the weighted geometric similarity model to Gati and Tversky (1984), Experiment 10, Set B.

Stimulus	N	Difference $ i_n - j_n $			Similarity	
		$n = 1$	$n = 2$	$n = 3$	s_{IJ}	\hat{s}_{IJ}
p-q	1	1	-	-	.16	.17
px-qx	2	1	0	-	.23	.23
py-qy	2	1	-	0	.23	.23
px-py	3	0	1	1	.54	.55
p-py	2	0	-	1	.74	.74
p-px	2	0	1	-	.77	.74

Note. The similarities s_{IJ} have been standardized to the range of 0 to 1. s_{IJ} = observed similarities; \hat{s}_{IJ} = predicted similarities; N = number of features present in a stimulus pair.

Experiment

To further test the common features effect with the geometric similarity model, I conducted a pairwise similarity judgment study similar to Gati and Tversky (1984). The data, code, and materials are available at <https://osf.io/yu8vt/>.

Method

Participants Sixty subjects (12 females, $M_{age} = 31.93$ years, $SD_{age} = 10.40$ years, age range: 19-66 years), recruited over Prolific Academic (www.prolific.co), participated in a 15-minutes online study in exchange for a compensation of £2.60. The study was approved by the ethics board of the psychology department of the University of Basel (#025-18-9).

Materials Participants judged the similarity of geometric figures on a visual slider ranging from “very dissimilar” (coded as 0) to “very similar” (coded as 1). The figures varied in their outer shape (a circle or a square), in the presence of a diamond, and in the presence of a cross, resulting in $2^3 = 8$ possible figures (see Figure 1 for visualizations).

Procedure Participants' task was to judge the similarity of pairs of stimuli with up to three features. After familiarizing themselves with the eight stimuli and the visual slider, participants judged all possible $8^2 = 64$ stimulus pairs twice in randomized order. In each trial, they saw two stimuli and entered their similarity judgment on a visual slider below.

Results

Participants' mean similarity judgments, split up for the number of common and differing features of a stimulus pair, can be seen in Figure 3 (left side). The figure shows that the similarity increased with the number of common features whenever there was at least one differing feature, replicating therefore the common features effect (e.g., Gati & Tversky, 1984).

To model the common features effect with the geometric similarity model of Eq. 2 and gain more insight into how additional features affect people's relative attention allocation, computational cognitive modeling was applied. I modeled participants' similarity judgments as being sampled from

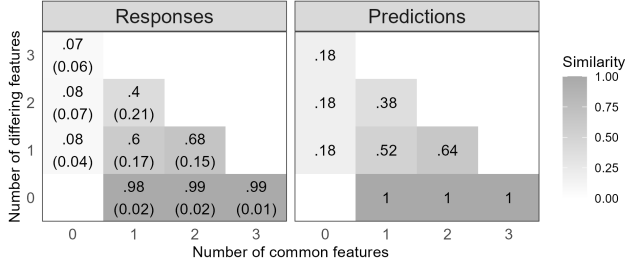


Figure 3: *Experiment, results.* Shown are the mean similarity judgments with in brackets the mean standard deviation across participants (left) as well as the prediction of the geometric similarity model with relative attention in Eq. 2 (right).

a normal distribution in which the mean equaled the predicted similarity as per Eq. 2 and the standard deviation σ was a free parameter. The parameters were estimated with maximum likelihood using individual participants' similarity judgments. Sensitivity c (with $0 \leq c \leq 10$) and σ (with $0 \leq \sigma$) were estimated from all trials; the attention weights w_{outer} (denoting the attention to the feature “outer shape”), $w_{diamond}$, and w_{cross} were estimated separately for trials with a different number of additional features. Specifically, one set of attention weights w_{outer} and $w_{diamond}$ (summing to 1) was estimated from trials in which only the cross was absent from both figures; a second set of attention weights w_{outer} and w_{cross} (also summing to 1) was estimated on trials in which the diamond was absent from both figures; and a final set of attention weights w_{outer} , $w_{diamond}$, and w_{cross} (also summing to 1) was estimated on trials in which no feature was absent⁴.

Table 2 shows the resulting aggregate attention weight estimates⁵. In particular, it can be seen that the attention to the omnipresent feature “outer shape” drops as the additive features “diamond” and “cross” appear in at least one of the two figures. This reallocation of attention away from already present features to new, additional features provides the basis for predicting the common features effect. As can be seen in Figure 3 (right side), the geometric similarity model also predicts that similarity increases with more common features, and it provides an overall good description of subjects' similarity judgments, with a mean log likelihood $M(\ell) = -69.09$ ($Mdn(\ell) = -69.37$ and $SD(\ell) = 26.96$), a root-mean-square error $RMSE = .14$, and a mean absolute error $MAE = .11$.

Importantly, these analyses mainly serve as a first illustration of the ability of the geometric similarity theory to predict the common features effect. Further research comparing the geometric similarity model with other models of psychological similarity (e.g. Tversky's featural model; Navarro & Lee, 2004; Tversky, 1977), for instance by means of out-of-sample model comparisons, will detail to what extent geometric similarities can account for the common features effect.

⁴In case no additive feature is present, $w_{outer} = 1$.

⁵The two remaining mean parameter estimates (with standard deviations in brackets) were $c = 1.81$ (0.45) and $\sigma = .14$ (.03).

Table 2: Parameter estimates for the attention weights.

N (additional feature)	w_{outer}	$w_{diamond}$	w_{cross}
1 (none)	1 (0)	-	-
2 (diamond)	.69 (.11)	.31 (.11)	-
2 (cross)	.76 (.11)	-	.24 (.11)
3 (diamond & cross)	.59 (.14)	.22 (.07)	.19 (.07)

Note. Shown are the mean attention weight estimates aggregated across subjects (with standard deviations in brackets). N = number of features present in a figure pair.

Discussion

This paper counters the claim that geometric similarities, by computing the feature value differences between objects, are insensitive to common features (e.g., Kruschke, 2008). I showed that a formal geometric similarity model predicts the similarity-increasing effect of common features by weighting each feature value difference by the share of attention allocated to the respective feature (e.g., as in Nosofsky, 1986). An additional common feature draws away attention from the already present features, whose differences will then receive less weight, ultimately increasing similarity. I illustrated this idea using data from Gati and Tversky (1984) and new data from a pairwise similarity judgment experiment, in which people judged the similarity of pairs of geometric figures. The geometric similarity model described the aggregate data well, with the exception of overestimating the similarity of objects that share no common features (outer left column in Figure 3).

While this study investigated how common features affect similarity across object pairs, the same principle might also hold within trials. Specifically, with increasing time devoted to an object pair, more features are included in the similarity computation, meaning that similarity may depend on the time course of attention (see also Lamberts, 1995). The same object pair might get more similar with time as further common features are attended to and included in the similarity computation. This could be described by a geometric similarity model that distributes attention to all features included in the similarity computation at a given point in time.

It is important to bear in mind that, although geometric similarities can be sensitive to common features, they represent psychological similarity in particular for continuous features, whereas other approaches such as featural similarities are adapted to process binary features (Lee & Navarro, 2002; Navarro & Lee, 2002a). Furthermore, geometric similarities are associated with other limitations such as assuming a symmetric similarity relation whereas empirical research has shown that the similarity of object i to object j may differ from the similarity of j to i (e.g., Tversky, 1977). Yet, also asymmetric similarity relations can be explained by a geometric approach (e.g., Nosofsky, 1991), and this paper adds to the evidence showing that, in many circumstances, the geometric similarity theory may provide a comprehensive description of psychological similarity.

Acknowledgements

This work was supported by the Swiss National Science Foundation [grant number P0BSP1-195389 to F. I. Seitz].

References

- Albrecht, R., Hoffmann, J. A., Pleskac, T. J., Rieskamp, J., & von Helversen, B. (2020). Competitive retrieval strategy causes multimodal response distributions in multiple-cue judgments. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 46(6), 1064–1090. doi: 10.1037/xlm0000772
- Falkowski, A., Sidoruk, M., Olszewska, J., & Jabłońska, M. (2021). Positive—negative asymmetry in evaluation of natural stimuli: Empirical study in the contrast model of similarity extended to open sets. *The American Journal of Psychology*, 134(1), 1–11. doi: 10.5406/amerjpsyc.134.1.0001
- Gati, I., & Tversky, A. (1984). Weighting common and distinctive features in perceptual and conceptual judgments. *Cognitive Psychology*, 16(3), 341–370. doi: 10.1016/0010-0285(84)90013-6
- Goldstone, R. L., & Son, J. Y. (2012). Similarity. In K. J. Holyoak & M. R. G (Eds.), *Oxford library of psychology. The Oxford handbook of thinking and reasoning* (pp. 155–176). Oxford University Press.
- Hoffmann, J. A., von Helversen, B., & Rieskamp, J. (2016). Similar task features shape judgment and categorization processes. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 42(8), 1193–1217. doi: 10.1037/xlm0000241
- Juslin, P., Jones, S., Olsson, H., & Winman, A. (2003). Cue abstraction and exemplar memory in categorization. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 29(5), 924–941. doi: 10.1037/0278-7393.29.5.924
- Kruschke, J. K. (2008). Models of categorization. In R. Sun (Ed.), *The Cambridge handbook of computational psychology* (pp. 267–301). New York, NY, US: Cambridge University Press. doi: 10.1017/CBO9780511816772
- Lamberts, K. (1995). Categorization under time pressure. *Journal of Experimental Psychology: General*, 124(2), 161–180. doi: 10.1037/0096-3445.124.2.161
- Lee, M. D., & Navarro, D. J. (2002). Extending the ALCOVE model of category learning to featural stimulus domains. *Psychonomic Bulletin & Review*, 9, 43–58. doi: 10.3758/BF03196256
- Navarro, D. J., & Lee, M. D. (2002a). Combining dimensions and features in similarity-based representations. In *Advances in Neural Information Processing Systems* (Vol. 15). doi: 10.31234/osf.io/qejyb
- Navarro, D. J., & Lee, M. D. (2002b). Commonalities and distinctions in featural stimulus representations. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 24). Retrieved from <https://escholarship.org/uc/item/11r2t9mn>
- Navarro, D. J., & Lee, M. D. (2004). Common and distinctive features in stimulus similarity: A modified version of the contrast model. *Psychonomic Bulletin & Review*, 11(6), 961–974. doi: 10.3758/BF03196728
- Nosofsky, R. M. (1986). Attention, similarity, and the identification–categorization relationship. *Journal of Experimental Psychology: General*, 115(1), 39–57. doi: 10.1037/0096-3445.115.1.39
- Nosofsky, R. M. (1991). Stimulus bias, asymmetric similarity, and classification. *Cognitive Psychology*, 23(1), 94–140. doi: 10.1016/0010-0285(91)90004-8
- Nosofsky, R. M. (2011). The generalized context model: An exemplar model of classification. In *Formal approaches in categorization* (pp. 18–39). New York, NY, US: Cambridge University Press. doi: 10.1017/CBO9780511921322
- Ritov, I., Gati, I., & Tversky, A. (1990). Differential weighting of common and distinctive components. *Journal of Experimental Psychology: General*, 119(1), 30–41. doi: 10.1037/0096-3445.119.1.30
- Seitz, F. I., Jarecki, J. B., & Rieskamp, J. (2023). Perceptual similarity mostly ignores within-category feature distributions: Evidence from computational modeling of human categorizations. *PsyArXiv*. doi: 10.31234/osf.io/7v95h
- Seitz, F. I., von Helversen, B., Albrecht, R., Rieskamp, J., & Jarecki, J. B. (2023). Testing three coping strategies for time pressure in categorizations and similarity judgments. *Cognition*, 233, 105358. doi: 10.1016/j.cognition.2022.105358
- Shepard, R. N. (1987). Toward a universal law of generalization for psychological science. *Science*, 237(4820), 1317–1323. doi: 10.1126/science.3629243
- Smith, D. J., & Minda, J. P. (1998). Prototypes in the mist: The early epochs of category learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 24(6), 1411–1436. doi: 10.1037/0278-7393.24.6.1411
- Thorwart, A., Glautier, S., & Lachnit, H. (2010). Convergent results in eyeblink conditioning and contingency learning in humans: Addition of a common cue does not affect feature-negative discriminations. *Biological Psychology*, 85(2), 207–212. doi: 10.1016/j.biopsycho.2010.07.002
- Tversky, A. (1977). Features of similarity. *Psychological Review*, 84(4), 327–352. doi: 10.1037/0033-295X.84.4.327
- Verguts, T., Ameel, E., & Storms, G. (2004). Measures of similarity in models of categorization. *Memory & Cognition*, 32, 379–389. doi: 10.3758/bf03195832
- Young, M. E., & Wasserman, E. A. (2002). Limited attention and cue order consistency affect predictive learning: A test of similarity measures. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 28(3), 484–496. doi: 10.1037/0278-7393.28.3.484

Comparing Model Variants Across Experimental and Naturalistic Data Sets

Florian Sense (florian.sense@infinite-tactics.com)

InfiniteTactics, LLC; Dayton, OH, USA

Michael Collins (michael.collins.74.ctr@us.af.mil)

ORISE at AFRL; Dayton, OH, USA

Michael Krusmark (michael.krusmark.ctr@us.af.mil)

CAE, Inc.; Wright Patterson Air Force Base, Ohio

Tiffany Jastrzembski (tiffany.jastrzembski@us.af.mil)

Air Force Research Laboratory; Dayton, OH, USA

Abstract

Computational models of human memory have largely been developed in laboratory settings, using data from tightly controlled experiments that were designed to test specific assumptions of a small set of models. This approach has resulted in a range of models that explain experimental data very well. Over the last decade, more and more large-scale data sets from outside the laboratory have been made available and researchers have been extending their model comparisons to include such real-life data. We follow this example and conduct a simulation study in which we compare a number of model variants across a range of eight data sets that include both experimental and naturalistic data. Specifically, we test the Predictive Performance Equation (PPE)—a lab-grown model—and its ability to predict performance across the entire range of data sets depending on whether one or both of its crucial components are included in the model. These components were specifically designed to account for spacing effects in learning and are theory-inspired summaries of the entire learning history for a given user-item pair. By replacing these terms with simple lag times (rather than full histories) or a single free parameter, we reduce the PPE’s complexity. The results, broadly speaking, suggest that the full PPE performs best in experimental data but that not much predictive accuracy is lost if the terms are omitted from the model when naturalistic data are concerned. A possible reason is that spacing effects are not very important in real-life data but very important in spacing experiments.

Keywords: Computational models of memory; Simulation study; Learning and forgetting; Large-scale data.

Introduction

Learning and forgetting are core areas of study within the computational modeling community. Snapshot performance measures can serve as proxies of mastery at any given time but fundamentally, both learning and forgetting are dynamic processes that unfold over time. As a consequence, the vast majority of experimental studies on learning and forgetting focus on teasing apart the temporal dynamics for learning and forgetting.

A century worth of experimental work has employed a wide range of tools to illuminate various aspects human memory. Computational cognitive models constitute one such tool. At their core, they formalize theoretical assumptions about how a cognitive process should map onto empirical data. Competing theoretical accounts thus formalized can be pitched against data from a suitably designed experiment to evaluate each model’s relative merit. The primary criterion to

evaluate competing models is traditionally their relative ability to fit data (Yarkoni & Westfall, 2017).

In many applied settings, however, what matters most is a model’s ability to generalize from historical data to new observations. In other words: prediction. In this context, a key question is how learning histories should be used to make predictions: Do we need to consider *all* previous interactions to make good predictions? And if we do, how should those be summarized or weighted (if at all)?

Here, we report findings from a simulation study in which (variants of) computational models of forgetting are compared across eight data sets from both lab experiments and large-scale naturalistic data. Specifically, we will scrutinize the Predictive Performance Equation (PPE) since, by default, it considers the entire learning history and imposes theoretically-grounded constraints on how those histories should be summarized. We compared this full model against variants that relax those assumptions and find that the full PPE generally makes the best predictions. Variants of it that only consider the most recent history perform very well as well, though, particularly in the noisier naturalistic data.

Methods

Data sets

We will evaluate the models of interests across eight data sets (see Table 1). A detailed description of each data set is beyond the scope of the current paper and we refer the interested reader to citations provided below for additional details. The data sets come, broadly speaking, from two domains: experimental studies and naturalistic data sets. The former implement specific manipulations and learners are tested in the laboratory; the latter are data collected ‘in the wild’ and learners have much greater control over *when* they interact with the materials and (usually) also *which* materials they study. The naturalistic data all follow their own algorithms for selecting/suggesting materials to present to students. The three adaptive scheduling algorithms marked with an asterisk (*) in Table 1 use the SlimStampen algorithm to handle the repetitions of items within a learning session (Sense, van der Velde, & van Rijn, 2021). The high-level criterion for a data set to be included here is that multiple users studied multiple items

Label	Scheduling	Instances	Users	Items	Repetitions	Accuracy
cbb21	adaptive*; experimental	62,382	291	60	6.30	0.792
Duolingo16	adaptive/curriculum; naturalistic	12,854,226	115,222	19,279	2.19	0.896
EdNet	user-driven; naturalistic	496,503	410	11,868	3.64	0.707
jla21	adaptive*; naturalistic	468,617	267	1402	9.05	0.892
statscloze15	experimental	58,316	478	144	2.74	0.545
topics16	adaptive*; experimental	98,213	67	150	9.77	0.877
UDRI	experimental	42,780	62	30	23	0.852
WSU	experimental	77,010	72	136	21.0	0.801

Table 1: An overview with summary statistics of the data sets used.

and that timestamps and accuracy information are available for each recorded instance/interaction. In the following, we will provide a number of pertinent details for each data set.

cbb21 Participants in Experiment 1 of this study completed two sessions in which they studied the locations of cities in the US on a map. We used both the MTurk and lab samples. For details, see van der Velde, Sense, Borst, and van Rijn (2021); particularly their Fig. 2 for the experimental design and Fig. 3 for the experimental design.

Duolingo These data were made available as part of the Second Language Acquisition Modeling challenge, the results of which—along with the data—are reported in Settles, Brust, Gustafson, Hagiwara, and Madnani (2018). A month worth of data for a subset of Duolingo users across three languages was made available and the challenge asked researchers to submit predictive models of errors made in hold-out data. We include here the complete data set.

EdNet The EdNet data stem from another community modeling challenge, which is described nicely by Pavlik and Eglington (2021). The data span two years of interactions with the tutoring tool *Santa* and contain massive amounts of nested data. More details are provided in Choi et al. (2019). We only use the basic KT1 data structure that provides timestamps and accuracy for student interactions and we only use a very small subset of users. Specifically, we used data from the 410 users that answered three or more unique questions and whose average mean repetitions were more than three (computed as instances for that user divided by number of questions). This is only 0.05% of the total data but constitutes a disproportionate (but still small) subset of 0.5% of the total data. We chose this small—and probably not representative—subset because for the current comparison, we needed time series of repetitions, which are sparse in these data.

jla21 These data come from a university course in which students had access to the SlimStampen adaptive scheduling tool to rehearse glossary items from the assigned text book. Students chose whether and when to use the tool and which chapter’s content they wanted to rehearse. The algorithm handled within-session repetitions. See Sense, van der Velde, and van Rijn (2021) for details.

statscloze15 This data set was downloaded from the Memphis Data Shop (data set ID = 1465). Some background information is also provided in (Pavlik, Eglington, & Harrell-Williams, 2021). MTurk workers learned about statistical concepts by filling in missing words in sentences. The spacing during learning and to the test were experimentally manipulated. The so-called cloze input format makes this data set distinct from the others.

topics16 Participants in this experiment completed six learning sessions across three days to study different materials. SlimStampen was used for within-session/-material scheduling. Details are documented in Sense, Behrens, Meijer, and van Rijn (2016).

UDRI These data come from a tightly controlled experimental setup that manipulated both the inter-trial and inter-session intervals across three session (all within-subject). We included only the data from the digit-doodle learning task. A complete description is provided by Collins, Sense, Krusmark, and Jastrzembski (2020).

WSU Participants in an experiment conducted in a sleep laboratory at Washington State University learned paired-associates over the course of 11 sessions spread across three days. Three items each were assigned 17 repetition schedules (within-subject) that all included exactly 22 repetitions. See Walsh et al. (2022) for details.

Models

The primary goal is to compare the complete PPE model against variants that have aspects stripped from it. Additionally, we will implement three additional models that will serve as additional comparisons. These additional models are deliberately very simple to provide a lower bound that the PPE (variants) should outperform. Importantly, each model was fit to each user’s data separately.

Benchmark models The control model (**ctrl**) serves as a benchmark of the quality of predictions if the timing information is discarded entirely. It simply computes the mean accuracy of the user in the training data and uses the value as the prediction for all instances of that user in the test data. Next, a two-parameter exponential decay model (**exp**) that can capture memory decay and prior knowledge effects is used. The

	#P	Brief description (see text for details)	NLL	rank
PPE	4	The full PPE using both the model time and stability term.	0.403	1.75
PPEnoMT	4	Using the raw lag time rather than model time but retaining the decay term.	0.406	2.50
PPEnoSt	3	Replacing the decay term with a single free parameter but retaining the model time.	0.403	2.88
PPEnMTnSt	3	Replacing model time with raw lag time <i>and</i> the decay term with a free parameter.	0.408	3.38
pwr	2	A two-parameter power decay model based on raw lag times.	0.430	5.50
exp	2	A two-parameter exponential decay model based on raw lag times.	0.428	5.62
ctrl	1	Uses mean accuracy in the training data as the predicted value in the test data.	0.453	6.38

Table 2: A brief overview of the models compared across the data sets listed in Table 1. Listed are the number of free parameters (#P) for each model along with a brief description and the negative log loss (NLL) values and rank obtained by each model (averaged across all data sets).

exact formulation is $\hat{p} = A + (1 - A) \cdot \exp^{-\alpha \Delta}$, where \hat{p} is the expected performance and Δ is the lag time. An analogous model assuming power-law decay (**pwr**) was implemented as $\hat{p} = A + (1 - A) \cdot \Delta^{-\beta}$. For an excellent discussion of this class of models, see Heathcote, Brown, and Mewhort (2000).

PPE and lesioned variants

The Predictive Performance Equation (PPE) was initially developed to account for established findings in the experimental psychology literature that pertain to the temporal dynamics of human learning and forgetting. Specifically, the power laws of learning and forgetting and the spacing effect. A recent and complete account is provided by Walsh, Gluck, Gunzelmann, Jastrzemski, Krusmark, Myung, et al. (2018). The goal has always been to develop a theory-grounded cognitive model that is applicable in applied settings and PPE has shown great promise there (Oermann, Krusmark, Kardong-Edgren, Jastrzemski, & Gluck, 2022). The primary question here is whether the various components that constitute the PPE are possibly too complex and whether comparable predictive results could be achieved by omitting them. This would both relax the theoretical assumptions made by the model and the computational demands when applying the model. We will briefly outline the model components that make up the full PPE and then present the variants we will compare.

At its core, the PPE features the product of a learning term ($N^{0.1}$, where N is the number of repetitions) and a forgetting term (T^{-d}). The latter accounts for the theoretical assumptions of the model by including two transformations of the time series associated with repetitions of an item by a user. The first transformation is the *model time*, MT , which is the weighted cumulative sum of elapsed time,

$$MT_i = \sum_{j=1}^N w_j \cdot t_j \text{ with } w_j = \frac{t_j^{-x}}{\sum_{j=1}^N t_j^{-x}} \quad (1)$$

with $x = 0.6$ by convention. The model time’s decay rate, d , is defined as $d = b + m \cdot \text{stability}$ and includes two free parameters and the second transformation: the lagged, cumulative mean of the inverse log lag times, Δ :

$$\text{stability} = \text{St} = \frac{1}{N-1} \cdot \sum_{i=1}^{N-1} \frac{1}{\ln(\Delta_i + e)} \quad (2)$$

The stability term accounts for spacing effects because massed training schedules will result in steeper decay. The model time’s motivation encapsulates “that the age of items in memory should be skewed toward the most recent presentations, but that study history should not be entirely discarded” (Walsh, Gluck, Gunzelmann, Jastrzemski, & Krusmark, 2018, p. 13; also see for more background information), which is one of the ideas expressed in ACT-R as well. Notably, both of these terms consider the *entire* training history for a given user-item pair and need to be recomputed every time new observations are added.

Finally, the PPE includes two additional parameters that stem from a logistic transformation, $\frac{1}{1 + \exp(\frac{\tau - M}{s})}$, which maps the unbound ‘activation’, $M = N^{0.1} \cdot T^{-d}$, onto the range $[0, 1]$ such that the model’s output can be interpreted as the probability of a correct answer.

PPE variants The three variants of the PPE we will consider here remove one or both of the transformations from the full PPE—see Table 2. *PPEnoSt*, for example, replaces the decay term, d , with a single free parameter, β , thus omitting the stability term and reducing the number of free parameters by one. The *PPEnMTnSt* includes neither the MT nor the St term. None of these change the learning term and all retain the logistic mapping function.

Procedure

The backbone of the simulation study was a five-fold cross-validation approach. For each data set, instances were randomly assigned to one of five folds. Each fold was designated in turn as the test set and the remaining folds used as training data. Thus, each model was fit to 80% of each data set’s instances five times. All models were fit using gradient descent, minimizing the negative log loss. (Which is why we also use the NLL as the main outcome metric to evaluate the predictions in the Results section below.) The end result of this approach is that each model made a prediction for each instance in each data set.

The first ‘repetition’ as a special case

Given the focus on the lag times in the current work, we omitted the first repetition from the evaluation of the predictions as well as Figure 2 since no lag time exists for those. We argue that predicting performance on the first instance is a separate prediction task than one that relies on learning histories. In most practical settings, it might make the most sense to have two separate models. First, a model that predicts the likelihood that a user will give a correct response on the initial interaction with an item (i.e., before the user has a history of interactions with a particular item). This is particularly important in naturalistic data sets in which learners should be assumed to have varying degree of prior knowledge/exposure to the study material (while experimental studies often choose materials such that no prior knowledge is likely/possible). A relatively simple model that takes into account item difficulty and user ability simultaneously (such as item response theory-based models) would probably be well-suited to this task. Second, a separate model that traces the learning and forgetting process as it unfolds over repeated interactions of a user with the same item can be implemented. This, really, is the domain of the models compared here. Therefore, we will restrict the following analyses to all interactions of an item except the first.

Results

As a first step, we present a high-level comparison of the models across the data sets as part of Table 2. Each model’s negative log loss (NLL) value for the predictions made for each data set were computed and, subsequently, models were ranked within each data set. The table shows both the average rank across the data sets as well as the average NLL value for each model. Ranks were computed because the NLL values vary substantially between data sets (see panel headings in Figure 1), which influences the mean NLL value but not the ranking. As the table shows, the complete PPE performs best overall with the lowest average NLL value and rank. The table also shows that—in the aggregate—all models outperform the *ctrl* model and that all variants of the PPE outperform the simple *exp* and *pwr* models.

A more nuanced and complete summary of the results is given in Figure 1. Since the primary comparison is against the full PPE, the NLL values for each model are shown relative to the PPE’s for each data set. The PPE’s NLL value is noted in each panel’s heading. Next to each bar, the relative change in NLL values from the full PPE model are denoted in percent¹ because a similarly sized bar can correspond to relatively smaller or larger changes depending on the base NLL value. There are a couple of high-level results that can be gleaned from this figure.

First, the two experimental data sets (WSU and UDRI) show the largest differentiation between models (i.e., larger

relative *and* absolute differences between models). This makes sense because these studies were explicitly designed to tease apart the differences between computational models of memory. Hence, the *ctrl* model does particularly poorly on these data sets. The PPE’s advantage over the lesioned variants is most pronounced in the WSU data. This is likely because the study’s core manipulation is a within-subject assignment to 17 different presentation schedules. Since parameters were fit for each user (for all models), a single set of parameters needs to account for the variance in performance of each user across all 17 schedules. The ‘full’ PPE does the best job but the three lesioned variants do not perform much worse.

Conversely, the full PPE does poorly on the statscloze15 data set. The best-performing model is *PPEnoSt*, which does better than the simple *exp* or *pwr* models. This suggests that both learning and forgetting effects are at play in these data (and Figure 2 suggests that the *PPEnoSt* model does particularly well predicting performance up to repetition eight).

For the naturalistic data sets, the differentiation between the models is much less clear and many variants perform essentially equally well. The bars indicating the differences between the full PPE and the variants are barely visible for most of those data sets and the percentages confirm that the relative differences between the models are rather small. This suggests that the lesioned variants perform roughly equally well to the full PPE on those data sets. Interestingly, in the Duolingo16 data, it appears that *all* models perform fairly similarly; the *PPEnMTnSt* model achieves the best results but even the largest relative difference in NLL values is only 2.7%.

One exception to the overall pattern is the statscloze15 data set. It is the only data set for which the PPE is outperformed by all but the *ctrl* model. This data set is conceptually most distinct from the others and it appears that the PPE is not well-suited to account for these data.

To zoom in on the results in more detail, we will look at how the predictive accuracy of the models changes over repetitions. One might expect that the model variants that include cumulative terms do better once a number of repetitions are available for an item. We will omit the *exp* and *pwr* models since they are always outperformed. We will also omit the *ctrl* model since its predictions do not change as a function of either repetition or lag time.

The data sets vary substantially in how often items are repeated (not shown in the figure). In the Duolingo16 data, for example, very few items are repeated five times, let alone 11+ times. In WSU data, on the other hand, *all* items are repeated 22 times by everyone. This points to a general difference in the distribution of items that we observe between experimental and naturalistic data sets. Not only are the temporal dynamics of the repetitions of items tightly controlled in experimental settings (and, almost by definition, not in naturalistic data) but items tend to be repeated more frequently.

Figure 2 shows each model’s average predicted perfor-

¹For example, if the PPE model’s NLL value was 0.500 and a competing model’s NLL value was 0.600, the bar in Figure 1 would reach to 0.1 on the x-axis (the absolute difference) and the percentage next to the bar would indicate “+20%” (the relative difference).

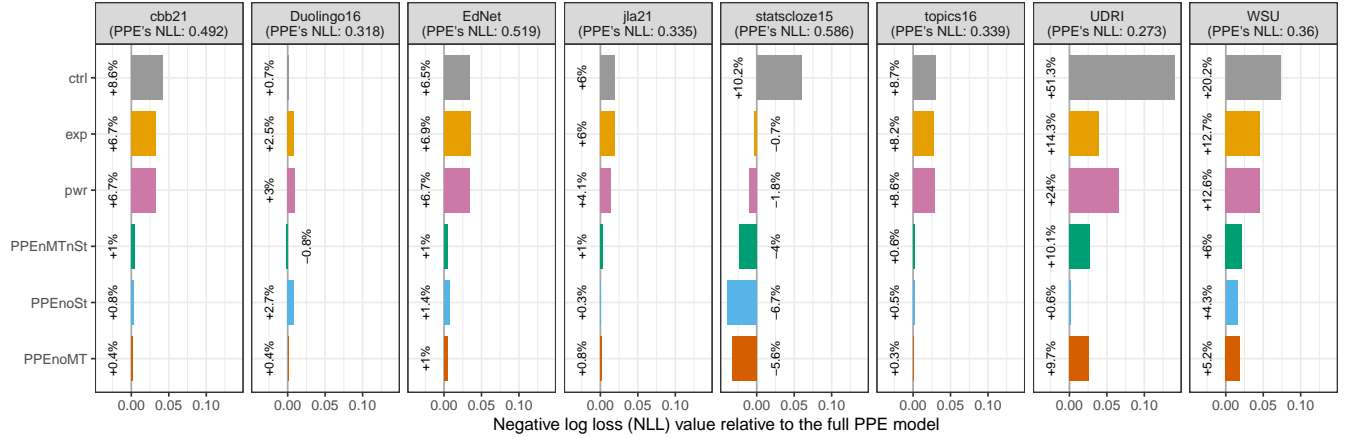


Figure 1: Each model’s NLL values in each data set relative to the PPE. Bars indicate absolute differences in NLL values; percentages denote relative differences in NLL values. Note: Lower NLL values indicate better predictive performance.

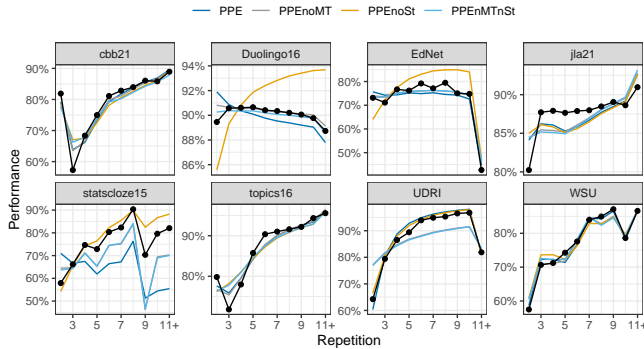


Figure 2: Mean accuracy (black) and each model’s average predicted performance at each repetition. The repetition denoted ‘11+’ includes the eleventh and all subsequent repetitions. Note that the y-axes differ between panels.

mance at each repetition. The most notable pattern that emerges is not that—as we expected—the models differ systematically over repetitions but rather than the models are remarkably similar to each other. In fact, on the SlimStampen data sets in particular, the models are more similar to each other than to the actual data.

Furthermore, the figure reveals that the pattern of performance over repetitions differs substantially between data sets. The EdNet data, for example, exhibits a marked zig-zag pattern, while the Duolingo data shows very little changes in performance over repetitions (note the scale of the y-axis). The latter explains why the *ctrl* model performs quite well in the Duolingo data: always predicting someone’s average accuracy is a good strategy if accuracy is relatively invariant. In the experimental data, on the other hand, it is much clearer that performance starts at a lower level and increases with additional repetitions.

Discussion

The results presented here are by no means simple and clear-cut. Our ambition was to conduct the comparison across a wide range of different data sets, which results in nuanced findings. At a high level, however, we can conclude that the full PPE generally makes the highest quality predictions. A notable exception is the statscloze15 data set, on which it performs poorly. The results also show, however, that while the PPE and its variants outperform the benchmark models, they do not differ from each other very much unless the comparison is made in a data set from an experiment designed to coax out differences between computational models of memory.

While one obvious take-away from these findings is that simpler versions of the PPE might be suitable for noisier, naturalistic data set, we want to acknowledge that it is by no means self-evident that a model predominantly developed on experimental data would fare so well in naturalistic data. In fact, there is no harm in using the maybe-too-complex PPE in these tasks, it appears. At least as long as predictive accuracy is all one cares about. If predictions need to be issued quickly and computing the model time and stability terms poses the main bottleneck, the current results suggest one might get similar predictive performance with a simpler variant. Using a simpler variant might open the door to one exciting area of application in which computational requirements have been prohibitive in earlier explorations: adaptive design optimization (Myung, Cavagnaro, & Pitt, 2013).

Previous work has pointed at model identifiability issues within the PPE’s formulation (Collins, Sense, Krusmark, Fiechter, & Jastrzembki, 2021) and it should be noted that these issues are not resolved by simply replacing the model time or stability term as done here. Along these lines, it is interesting to note that researchers working with the PPE regularly observe very low estimated values of the m (and b) parameters. If the m parameter is near zero, the decay term ($d = b + m \cdot \text{stability}$) collapses to $-b$, which is the same as

the *PPEnoSt* model.

We believe the current study is particularly interesting because the model comparison was conducted across a large range of data sets. This ensures that the conclusions of a comparison is not limited to the specifics of a single data set. Admittedly, it does make the outcomes more difficult to interpret and to extract simple take-home messages. The robustness of the inference is well worth it, though. To facilitate future across-data-set comparison, we are currently working on documenting a large “test harness” of data sets that can be used to this end. We encourage other researchers to contact us with additional data sets that (a) include multiple users interacting with multiple items/stimuli, (b) ideally across multiple sessions, (c) includes timestamps (down to the second), and (d) a performance metric associated with each instance.

Finally, the side-by-side comparison of the models’ performance in both experimental and naturalistic data sets is particularly valuable. Here, for example, we see that the differences between the models were much less pronounced in the naturalistic data sets. This is probably primarily because in those, the exact time course of the practice events is not as important as other factors. And because maybe more factors are under the learner’s control rather than controlled by the experimenter. This certainly seems to be the case in the Duolingo data—see, for example, Table 4 in Settles et al. (2018) and Table 1 in Sense, Wood, et al. (2021), both list feature importance values and show that timing information is not as important as user-, item-, and environment-based features. It is probably safe to assume the same pattern holds across many naturalistic data sets. This poses a challenge to most cognitive models developed in the lab, since they assume that time series are the crucible of learning and forgetting and are not readily extended to take into account additional features (e.g., exercise format). Extending theory-based model’s capabilities to this end would be particularly valuable so we can leverage insights from those models on large-scale data of naturalistic learner behavior.

Acknowledgments

This research is supported by a grant through the Air Force Office of Scientific Research’s Science of Information, Computation, Learning & Fusion Program. We wish to thank Hendrik Doug Riecken for valuable feedback and technical help provided for this work.

Data were wrangled in R (R Core Team, 2020) using *tidyverse* (Wickham et al., 2019); figures were created with *ggplot* (Wickham, 2016).

References

- Choi, Y., Lee, Y., Shin, D., Cho, J., Park, S., Lee, S., ... Jang, Y. (2019). Ednet: A large-scale hierarchical dataset in education. *CoRR*, *abs/1912.03072*. Retrieved from <http://arxiv.org/abs/1912.03072>
- Collins, M. G., Sense, F., Krusmark, M., Fiechter, J., & Jastrzemski, T. S. (2021). Parameter correlations in the predictive performance equation: Implications and solutions. In *Virtual mathpsych/iccm*.
- Collins, M. G., Sense, F., Krusmark, M., & Jastrzemski, T. S. (2020). Improving predictive accuracy of models of learning and retention through bayesian hierarchical modeling: An exploration with the predictive performance equation. In *Cogsci*.
- Heathcote, A., Brown, S., & Mewhort, D. J. (2000). The power law repealed: The case for an exponential law of practice. *Psychonomic bulletin & review*, *7*(2), 185–207.
- Myung, J. I., Cavagnaro, D. R., & Pitt, M. A. (2013). A tutorial on adaptive design optimization. *Journal of mathematical psychology*, *57*(3-4), 53–67.
- Oermann, M. H., Krusmark, M. A., Kardong-Edgren, S., Jastrzemski, T. S., & Gluck, K. A. (2022). Personalized training schedules for retention and sustainment of cardiopulmonary resuscitation skills. *Simulation in Healthcare*, *17*(1), e59.
- Pavlik, P. I., & Eglington, L. G. (2021). Modeling the ednet dataset with logistic regression. *arXiv preprint arXiv:2105.08150*.
- Pavlik, P. I., Eglington, L. G., & Harrell-Williams, L. M. (2021). Logistic knowledge tracing: A constrained framework for learner modeling. *IEEE Transactions on Learning Technologies*.
- R Core Team. (2020). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from <https://www.R-project.org/>
- Sense, F., Behrens, F., Meijer, R. R., & van Rijn, H. (2016). An individual’s rate of forgetting is stable over time but differs across materials. *Topics in cognitive science*, *8*(1), 305–321.
- Sense, F., van der Velde, M., & van Rijn, H. (2021). Predicting university students’ exam performance using a model-based adaptive fact-learning system. *Journal of Learning Analytics*, *8*(3), 155–169.
- Sense, F., Wood, R., Collins, M. G., Fiechter, J., Wood, A., Krusmark, M., ... Myers, C. W. (2021). Cognition-enhanced machine learning for better predictions with limited data. *Topics in Cognitive Science*.
- Settles, B., Brust, C., Gustafson, E., Hagiwara, M., & Madnani, N. (2018). Second language acquisition modeling. In *Proceedings of the thirteenth workshop on innovative use of nlp for building educational applications* (pp. 56–65).
- van der Velde, M., Sense, F., Borst, J., & van Rijn, H. (2021). Alleviating the cold start problem in adaptive learning using data-driven difficulty estimates. *Computational Brain & Behavior*, *4*, 231–249.
- Walsh, M. M., Gluck, K. A., Gunzelmann, G., Jastrzemski, T., & Krusmark, M. (2018). Evaluating the theoretic adequacy and applied potential of computational models of the spacing effect. *Cognitive science*, *42*, 644–691.
- Walsh, M. M., Gluck, K. A., Gunzelmann, G., Jastrzemski, T., Krusmark, M., Myung, J. I., ... Zhou, R. (2018). Mechanisms underlying the spacing effect in learning: A

- comparison of three computational models. *Journal of Experimental Psychology: General*, 147(9), 1325.
- Walsh, M. M., Krusmark, M., Jastrzembski, T., Hansen, D. A., Honn, K. A., & Gunzelmann, G. (2022). Enhancing learning and retention through the distribution of practice repetitions across multiple sessions. *PsyArXiv Preprint: 10.31234/osf.io/dmf4p*.
- Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. Retrieved from <https://ggplot2.tidyverse.org>
- Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., ... Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43), 1686. doi: 10.21105/joss.01686
- Yarkoni, T., & Westfall, J. (2017). Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, 12(6), 1100–1122.

Estimating Individual Property in a Simple Memory Task

Kohei Shimbori, Jumpei Nishikawa, Kazuma Nagashima, Junya Morita

Shizuoka University, 3-5-1 Johoku, Naka-ku, Hamamatsu-shi, Shizuoka-ken
shimbori.kohei.21@shizuoka.ac.jp

Keywords: ACT-R, Internal States, Memory Errors

Introduction

Human memory is the basis of personal identity linking the past to the present and the present to the future. However, in our daily life, we frequently encounter memory errors, which include errors of commission (wrong item recollection) and omission (failures of recollection). It can be assumed that individual properties, such as age, self-confidence, mental and physical conditions, affect the tendency of occurring those errors. In other words, it is possible to infer those properties from the errors expressed in some memory tasks. In this study, the above hypothesis is tested from a cognitive modeling approach. To achieve this goal, we conducted an experiment to collect recall errors in a simple experiment and performed parameter fitting on the experimental data using a model built by the cognitive architecture called Adaptive Control of Thought-Rational (ACT-R: Anderson, 2007).

Number sequences memory task

In the experiment, participants were asked to memorize and report a 10-digit number sequence presented on a monitor. The number sequence was randomly generated and presented for a duration of two seconds. After viewing the sequences, the participants responded by typing the memorized sequences into a text box on the monitor using the keyboard for 20 seconds.

Fifty participants recruited through crowdsourcing completed the task for 30 number sequences. After completing the task, they answered questionnaires asking about personal attributes, their self-confidence in memory ability, and any physical and mental conditions. Their mental condition was assessed using a Japanese version of PANAS (Positive and Negative Affect Schedule: Watson et al., 1988), which was translated by Kawahito et al. (2012).

The obtained responses were evaluated by the similarity with the presented sequence. Among various similarity measures, we adopted the Levenshtein Distance (LD). The value of this distance is zero when the presented and answered sequences are an exact match. The histogram constructed from all participants' responses is shown in Figure 1a.

Model

We used a modified grouped recall model included in the ACT-R Tutorial Unit5 (Bothell, 2022). The model simulates the recollection of a simple sequence of numbers,

1234567890, by grouping adjacent numbers like (123) (456) (7890). Within each group, each number is associated with a position index (first to fourth) within the group.

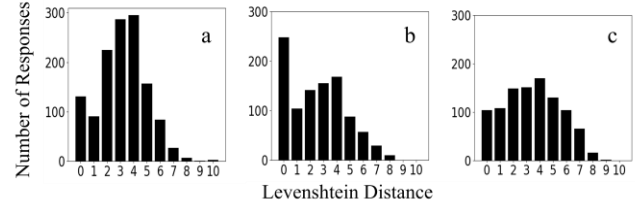


Figure 1. The results of the experiment and simulation

In recalling a number sequence, the model reports the numbers in order, starting from the left of the sequence, shifting groups and position indices. That is, at the beginning of the task, the model attempts to retrieve the “first number” in the “first group” from the declarative memory. Next, the second and third numbers are retrieved and reported. If the memory of the number corresponding to the position cannot be retrieved, the model moves on to the numbers in the “second group.” This process is repeated until the fourth number in the “third group” is retrieved.

In our model, the number i in the declarative memory is assigned an activation value (A_i), defined as $\sum_l P M_{li} + \epsilon_i$, where l and M_{li} indicate the slots of the retrieval request and the similarity between the slot content and the corresponding attribute of the number i , respectively. In this model, the maximum and minimum value of M_{li} are set to 0 and -1, respectively while -0.5 is set for M_{li} of pairs of adjacent groups (first group and second group, second group and third group) and for adjacent positions within a group (first position and second position, second position and third position, third position and fourth position). This similarity is weighted by P (mismatch penalty). As the name suggests, this parameter determines the degree of degradation when the number has low similarity with the retrieval request. Depending on those parameter values and random noise (ϵ), the activation of similar numbers sometimes exceed the activation of the originally requested number.

Summarizing the above, a small P and large ϵ induce a high probability of commission errors, while the error of omission occurs when the activation values of all digits in the memory fall below a set threshold. In ACT-R, the threshold is set by the parameter RT (retrieval threshold).

Simulation of overall data

Figure 1b presents the histogram of LD obtained by 1000 runs using the default parameters in the ACT-R grouped

Table 1 Correlation between individual model and response obtained in the questionnaire.

	HI	P	RT
Age	-0.369*	-0.019	-0.226
Gender	-0.092	0.300*	-0.114
Physical condition	0.069	-0.091	0.030
Education	-0.124	0.113	-0.081
Memory confidence	-0.001	-0.007	0.240
Strong	-0.171	0.192	0.030
Inspired	-0.079	-0.124	-0.100
Active	-0.066	-0.123	-0.048
Enthusiastic	-0.046	0.054	-0.055
Interested	-0.003	0.031	-0.093
Excited	0.032	0.043	-0.036
Proud	-0.023	-0.043	-0.023
Alert	-0.027	0.024	-0.115
Determined	0.038	0.048	-0.201
Attentive	0.021	-0.135	0.059
Positive Score	-0.014	-0.003	-0.054
Afraid	0.286*	-0.036	0.108
Scared	0.188	-0.070	0.023
Upset	0.237	-0.066	-0.002
Ashamed	0.097	-0.076	-0.060
Guilty	0.049	0.023	-0.246
Nervous	0.028	0.115	-0.003
Distressed	0.174	0.185	-0.136
Irritated	0.184	-0.021	-0.308*
Jittery	0.123	-0.147	-0.168
Hostile	0.043	0.036	-0.066
Negative Score	0.186	-0.013	-0.098

model. Comparing the human histogram, the model achieved a better recollection of the presented sequences (distance 0). This is considered to be due to the differences in the coding phase for the number sequence. The model in this study did not include the processes of coding the sequence; rather the sequence was directly coded in the declarative memory. In contrast, participants needed to memorize the sequence from a brief period of two seconds. Despite these differences, there are some similarities between the participants and the model, especially in the middle range of LD.

Following the above default simulation, the grid space consisting of P (0.2 to 2.0 in 0.2 increments) and RT (-3.0 to 1.0 in 0.5 increments) was searched to obtain the best parameter set reproducing the overall human data (1308 responses screened from 30 sequences by 50 participants). The parameter range was determined based on the preliminary simulation with the constraint of execution cost, and the fitting of the model was evaluated by histogram intersection. Figure 1c shows the best fit histograms whose intersection and correlation with the human data were 0.981¹ and 0.952, respectively. The parameter set obtained by this search consisted of 0.8 (P) and -0.5 (RT).

Simulation of individual data

We also performed the same parameter search for individual data. The histograms were constructed from the 30-number sequences reported by each participant. We then searched for the parameters of the model with the largest histogram

intersections for each participant. Table 1 shows Spearman's rank correlation coefficients between the output from this process and the responses obtained from each individual through the questionnaire.

The column titled HI indicates the correlation between the histogram intersections estimated for each individual and the questionnaire items. Thus, significant correlations in this column indicate that the individual property affected the degree of fitting to the model constructed in the range of the parameter set. From the table, we can see several attributes and mental state assessment are significantly correlated with this index. A negative correlation was obtained for HI with age, and a positive correlation was obtained for HI and the mental state "afraid." This suggests that, within the range of the parameter settings in this study, the model fits well with the responses of participants who are younger and those who felt "afraid."

Concerning the estimated parameter, negative correlations were obtained between RT and the "irritated." This can be interpreted as meaning that the more irritated the participants felt during the experiment, the more likely they were to make errors of omission.

Summary

In this study, we examined whether individual properties involved in memory errors can be estimated by fitting a cognitive model. We conducted systematic parameter fitting to the results of experiments in which memory errors were induced, both to the overall data and to individual-level data. Significant correlations were observed between some of the emotional evaluation items and the model parameters. This suggests that the present method can be used to estimate individual properties and internal states from memory errors in a specific task. However, the results obtained in this study have limitations in terms of parameter range. Therefore, we expect that the estimation possibility of the internal state will increase by improving the accuracy of the fitting and by modeling the data in a more individualized manner.

References

- Anderson, J. R. (2007). How can the human mind occur in the physical universe? Oxford University Press.
- Bothell, D. (2022). Unit 5: Activation and Context, ACT-R tutorial (Version 7.27.7).
- Kawahito, J., Otsuka, Y., Kaida, K., and Nakata, A. (2022). Reliability and validity of the Japanese version of 20-item Positive and Negative Affect Schedule. *Hiroshima Psychological Research*, 11, 225-240.
- Watson, D., Clark, L. A., & Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: the PANAS scales. *Journal of personality and social psychology*, 54(6), 1063-1070.

¹ Normalized histogram intersections between experimental data as maximum values

Leveraging Large-Scale Brain Connectivity Data to Explore and Expand the Common Model of Cognition

Sönke Steffen (s.steffen@rug.nl)

Dr. Catherine Sibert (c.l.sibert@rug.nl)

Bernoulli Institute, University of Groningen
Groningen, The Netherlands

Keywords: Common Model of Cognition; CMC; Brain Connectivity; fMRI; Human Connectome Project; HCP

Over the past decades, a vast amount of models and architectures have been developed, looking at the large scale organization of the human brain on different levels of abstraction. In an attempt to synthesize the ideas from some of the most established existing models of cognitive processing, namely ACT-R, SOAR, and Sigma, the Common Model of Cognition (CMC) has been proposed. It identifies five different modules within the brain with discrete functionalities and processing connections between them, modules for Perception, Action, Long-Term Memory, Procedural Memory, as well as Working Memory. These are considered to be essential for cognition across different domains and tasks, representing a generalized model of the structuring and processing of the mind.

Previous work has connected the structure of the CMC to activity in the specific brain regions, helping to validate the model and compare it to other models and architectures, like Hub-and-Spoke Architectures and Hierarchical Architectures. The CMC was found to outperform its alternatives, being a significantly better match for the experimentally gained data. However, the results also suggested that modifications to the original formulation of the CMC would improve its fit. This is not surprising, as the CMC has a rather basic structure, only incorporating high level cognitive components. Other models consist of larger networks of sub-components, representing real human cognition more accurately. It further does not consider many significant aspects of cognitive processing like metacognition or emotional processing in the modularity and organization.

The large scale parcellation currently used to identify signals associated with each cognitive component will not be sufficient in the future, as the model grows in complexity and additional cognitive components are incorporated. Better methods are needed for identifying regions associated with specific cognitive processes and modeling these and its connections in the CMC.

To improve the identification of brain regions we can use meta-analyses of brain data. Tools like Neurosynth synthesize the results of many studies using neuroimaging, allowing to perform connectivity analyses on them. This makes it possible to relate specific brain regions to functions,

as well as investigate the interactions between the different regions, which can be leveraged to inform the CMC about its structure. Due to the large amount of data and the wide variety of domains covered, meta-analyses of brain data are significantly more powerful than single studies. To validate our methods, we can use fMRI brain data from the Human Connectome Project. It provides a wide range of brain activity across multiple tasks allowing us to compare different configurations of the CMC using methods of connectivity analysis.

We propose leveraging the power of connectivity analyses with both large-scale fMRI brain data and meta-analyses of brain data to create expanded and more robust versions of the CMC. The methodology used to research this is defined as follows: First, look at shortcomings of the current CMC structure and create expanded versions with additional components integrated in a plausible way. Then identify and isolate brain activity associated with those components using the proposed combination of meta-analyses and fMRI brain data. Finally, compare the resulting predictions with the current CMC structure.

Novelty Detection, Insect Olfaction, Mismatch Negativity, and the Representation of Probability in the Brain

Terrence C. Stewart (terrence.stewart@nrc-cnrc.gc.ca)

National Research Council Canada, University of Waterloo Collaboration Centre
Waterloo, ON, Canada

P. Michael Furlong (michael.furlong@uwaterloo.ca)

Kathryn Simone (kathryn.simone@uwaterloo.ca)

Madeleine Bartlett (madeleine.bartlett@uwaterloo.ca)

Jeff Orchard (jorchard@uwaterloo.ca)

Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada

Abstract

We present a unified model of how groups of neurons can represent and learn probability distributions using a biologically plausible online learning rule. We first present this in the context of insect olfaction, where we map our model onto a well-known biological circuit with a single output neuron that represents whether the current stimulus is novel or not. We show that the model approximates a Bayesian inference process, providing an explanation as to why the current flowing into the output neuron is proportional to the expected probability of that stimulus. Finally, we extend this model to show that the same circuit can detect deviations in temporal patterns, like the expectation violations that elicit the EEG mismatch negativity signal.

Keywords: novelty detection; insect olfaction; mismatch negativity; neural representation; hyperdimensional computing; fractional binding; spatial semantic pointers; Bayesian inference

Introduction

It is critically important for any cognitive agent to recognize their sensory stimuli as novel or unexpected. Different strategies may be applicable, depending on whether one is in a familiar situation (in which case one can safely rely on previously learned knowledge), or in a novel situation (in which case a more careful and exploratory strategy may be appropriate). In mammals, an example of this is seen in the Mismatch Negativity signal, a strong EEG signal that appears approximately 200ms after a surprising stimulus (Pazo-Alvarez et al., 2003). This seems to be an automatic process, occurring regardless of whether the participant is paying attention to the stimulus or not. This automaticity and the speed of the response suggests that this novelty detection is a simple and basic process that may be understood without involving the entire brain.

Furthermore, a specific neural circuit for novelty detection has also been identified in the insect brain. The MBON (Mushroom Body Output Neuron) $\alpha'3$ neuron is consistently active in the presence of novel odours, and is silent for odours that have been previously encountered. The inputs to this neuron come from Kenyon Cells, which form a very sparse representation of the current odour, so each odour corresponds to a different (sparse) pattern of activity in these neurons. In (Dasgupta et al., 2018), this system is compared to the computer-science idea of a Bloom Filter, a type of hashtable where input data is converted into a sparse representation, and then

individual elements of that representation (i.e. the activity of the Kenyon Cells) are used to quickly determine whether the current input is likely to be novel or not. The core idea is to do this without requiring a complete database of every odour that has been previously observed; instead, use the overlap in the sparse representation as a fast estimate as for the input's novelty.

In this paper, we present a simple model of this novelty detection system that is compatible with the above idea, but interprets the computation being performed by the neurons in a slightly different way. In particular, we suggest that the neurons (and the connection weights between them) are in fact representing a probability distribution, and “novelty” is detected if the current input is highly unlikely according to that distribution. We show that a very simple learning rule, combined with a particular method for encoding information in neurons, results in a network that accurately estimates the observed probability distribution of different inputs, and that a single neuron (such as the MBON $\alpha'3$ neuron) can use this distribution to signal novelty.

Given this insect-based model, we then expand the system to encode information over time, and show that the very same model is capable of detecting the sort of temporal novelty that is the hallmark of the Mismatch Negativity signal in mammals. This expanded system makes use of Legendre Memory Units (LMUs), a recurrent neural structure that has been mapped to Time Cells (Voelker et al., 2019), temporal patterns in the cerebellum (Stöckel et al., 2021), and has been shown to improve performance on Machine Learning tasks over LSTMs (Voelker et al., 2019) and Transformers (Chilkuri et al., 2021).

Vector Representation

Given the wide range of possible inputs for a novelty detection system, we define our inputs simply as vectors. These can be of arbitrary dimensionality, and we do not constrain their magnitude. In this way we can deal with inputs such as odours (which in insects can be thought of as vectors in a 50-dimensional space based on 50 different odour-sensing neurons), or images, or numerical values in general. Interestingly, the resulting model we define here will not need to be changed in any way to handle different types of inputs, other

than using our general-purpose input mapping.

When inputting a numerical vector into a neural network, it is generally useful to transform it in some way. Typically, this is some sort of normalization, ensuring that the input has a mean of zero and a standard deviation of one, for example. However, normalizing requires knowing the overall range of possible input values. As an alternative, we adapt a method used for encoding arbitrary-length lists into a fixed-length vector.

In particular, Plate (1995) suggests using randomly chosen high-dimensional unit vectors for representing structured information. The list $[A, B, C]$ can be represented as the vector $A \otimes X + B \otimes X \otimes X + C \otimes X \otimes X \otimes X$, where \otimes is circular convolution, A, B, C , and X are randomly chosen unit vectors (and X is also *unitary*, ensuring that its magnitude stays 1 after repeated circular convolution). Since circular convolution is element-wise multiplication in the Fourier domain, the repeated convolution $X \otimes X \otimes X$ can be written as X^3 , leading naturally to a generalization where the exponent is a real number instead of an integer.

In other words, we can represent the position x on the x -axis as X^x , where X is a randomly chosen unitary vector, and where the exponent means “take the Fourier transform, then raise each element to the power x , then take the inverse Fourier transform”. For a two-dimensional input $\langle x, y \rangle$ we can compute $X^x \otimes Y^y$, and so on for higher dimensions. Importantly, the resulting vector is *always* a D -dimensional unit vector (where D is the dimensionality of the base vectors X, Y , etc., which we set to be 1024 here). This is true regardless of the number of actual inputs, meaning that we do not need to change anything about the internals of our novelty detection model to deal with different inputs. Furthermore, since the input is always a unit vector, we do not need to further normalize the inputs.

While this approach to representation was mentioned in (Plate, 1995), it is further analyzed in (Lu et al., 2019). In particular, the Fourier transform of a unitary vector leads to a vector with complex components $[e^{i\phi_1}, e^{i\phi_2}, e^{i\phi_3}, \dots]^T$, so raising this to a power x gives $[e^{i\phi_1 x}, e^{i\phi_2 x}, e^{i\phi_3 x}, \dots]^T$. This produces a series of oscillations, and as long as the ϕ_i values are relatively prime, the exact pattern of oscillation will never repeat, no matter how much one varies x . Of course, for some x values the resulting vectors will be very close to each other, leading to the possibility of confusion between some points, but in a high-dimensional space (D), this will be uncommon.

This idea of vectors being close to each other also provides us with an important parameter for the model. In particular, a small change in x will produce a small change in the vector X^x . If we quantify this with the dot product, it can be shown that, in the limit as $D \rightarrow \infty$, the similarity between X^a and X^{a+x} approaches the normalized sinc function, $\text{sinc}(\pi x)/(\pi x)$. That is, for $x = 1$, the two vectors will be orthogonal (no similarity), but for smaller values of x the vectors will be closer and closer to each other. This gives the representation a particular *scale*. Depending on our inputs, we may want to con-

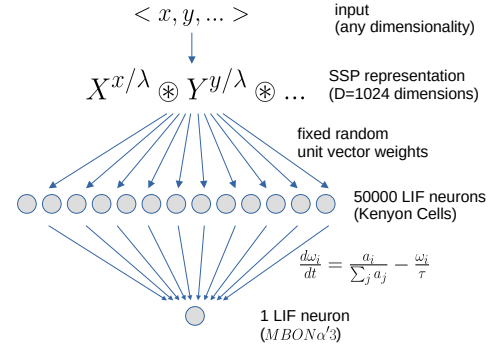


Figure 1: Our novelty detection model. Arbitrary input is converted into an SSP and passed to LIF neurons via randomly chosen fixed connection weights. Output weights are increased whenever a neuron is active, and decayed over time. In the insect, the neurons correspond to Kenyon Cells and the output is the MBON $\alpha'3$ neuron.

trol this scale, and we do this by introducing a length scale parameter λ , and encode information as $X^{x/\lambda} \otimes Y^{y/\lambda}$. In this way, values that are less than λ apart will yield vectors with high similarity. For our novelty detection system, this gives us a reference for how different an input needs to be from other inputs to be considered novel.

We call this style of representation a *Spatial Semantic Pointer*, or SSP, since it gives a compressed representation of an infinitely large space, but maintains semantic information in that it yields high similarity for nearby x values. This approach to continuous representation is first found in (Plate, 1992), and more recently in (Frady et al., 2018), where it is known as *Fractional Power Encoding*, or *Fractional Binding*, since the \otimes operator is thought of as a binding operator in Vector Symbolic Architectures.

Computational Model

Our computational model is shown in Figure 1. The first step is to convert the input into an SSP vector, using the above approach. We use $D = 1024$ here. Next, we define 50,000 neurons, each with a separate randomly-chosen 1024-dimensional vector for its input connection weights. That is, each neuron receives as input the dot product between the actual input and a randomly chosen “preferred stimulus” for that neuron. This is a generalization of the standard finding of preferred-direction-vectors in sensory and motor cortices (Georgopoulos et al., 1982; Schwartz et al., 1988). Each neuron also has a negative bias input that controls how similar the input needs to be to its ideal stimulus in order for the neuron to be active. This controls the sparsity of the neural representation. While any rectified neuron model can be used for this, here we use spiking Leaky Integrate-and-Fire (LIF) neurons with a bias of 0.9.

Given this input, the 50,000 neurons will form a sparse representation of that input, corresponding to Kenyon Cells observed in the insect. In order to learn what stimuli are common

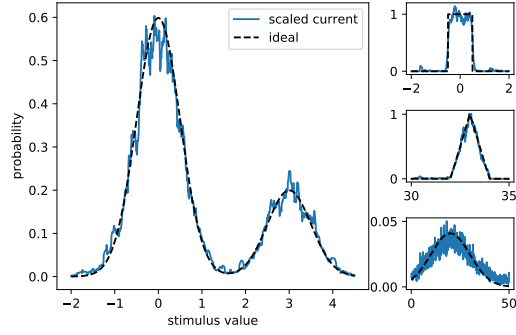


Figure 2: Queried probability after exposure to four different distributions of data. Blue line is the total current flowing into the output neuron, linearly scaled to unit area. Black dashed line is the ideal distribution. Four different distributions are shown. Note that even though the x-axis is very different for each distribution, the model itself is not changed in any way.

given this representation, we add a simple learning rule to the output connection weights of the model. These weights are initialized to zero, and each is increased proportional to the normalized activity of its corresponding, pre-synaptic neuron. Finally, we also decay the weights toward 0 over time, resulting in a weight learning rule of $\frac{d\omega_i}{dt} = \frac{a_i}{\sum_j a_j} - \frac{\omega_i}{\tau}$, where a_i is the activity of the i^{th} neuron, ω_i is the connection weight, and τ is a time constant for the decay.

The overall output from this system should be large for familiar inputs and small for unfamiliar inputs. Surprisingly, as shown in Figure 2, the output current is *proportional to the probability of the input!* That is, rather than just detecting novel vs familiar inputs, the system learns to directly represent the probability distribution of the input. Here we present inputs sampled from four different distributions (black dashed lines), and then measure the output current over a range of values from across the input domain. Importantly, this method automatically calibrates itself for whatever range of input values it receives. For example, the triangular distribution consists of values between 32 and 34, while the square and bimodal distributions cover values in the smaller ranges of -2 to 5). This is because the neurons have preferred inputs that cover the entire 1024-dimensional space, which itself covers the entire infinite range of possible inputs to the model, thanks to the SSP representation.

However, the wide distribution (Figure 2, bottom-right) between 0 and 50 shows higher variance in the probability representation. This is due to the length-scale parameter λ . The effects of this parameter are explored in Figure 3, showing that high variance can result from a value that is too small, but a value that is too large can lead to reduced accuracy. Thus, while this approach is robust to values anywhere on the x-axis, it is sensitive to the overall scale of values being represented.

We can also represent multidimensional probability representations just by encoding samples as $X^x \otimes Y^y$. Again, nothing is changed about the neural aspect of the model; all that is

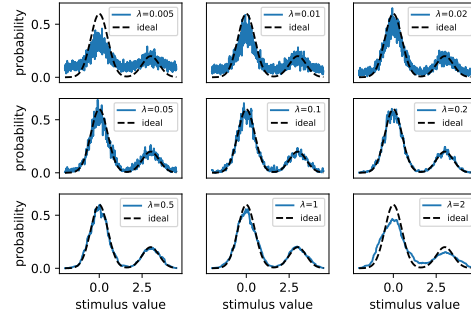


Figure 3: Queried probability given exposure to a particular distribution of data. Nine different versions of the model are shown, each with a different length scale (λ). When λ is too small (< 0.05) or too large (> 1), the representation is less accurate, but is fairly robust in-between.

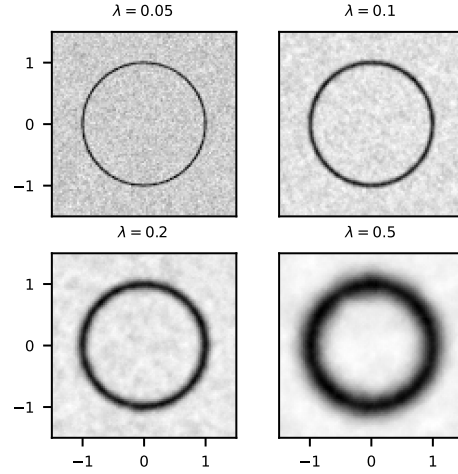


Figure 4: Queried probability given exposure to a two-dimensional data sampled evenly from a unit circle. Four different versions of the model are shown, each with a different length scale (λ). With larger λ the model generalizes across a larger region, so points slightly off the unit circle are not considered novel.

changed is the mapping into the 1024-dimensional SSP space. Figure 4 shows the resulting probability distribution (the blue curve in Figures 2 and 3) for two-dimensional input data that is sampled evenly from the unit circle. Notice that the length scale λ controls the resolution of the representation, controlling how far off the unit circle an input needs to be before it is considered to be different enough from observed data to be novel.

For a more detailed analysis of the accuracy of this model, Figure 5 demonstrates the overall linearity of the representation, and Figure 6 shows the accuracy as the number of neurons and length scale λ are varied.

Why This Works

To understand why this works, we have to consider the embedding of the SSP representation. The dot product be-

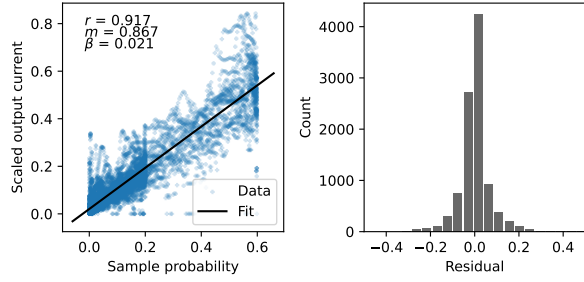


Figure 5: Characterization of relationship between sample probability and output current. Left: Scatterplot depicting individual samples of output current against observation probabilities given the distribution, with data merged across 10 runs of the model. Apparent is a linear relationship (regression line shown in black) and uniform variance in the representation error as a function of probability. Results shown are for $N = 1000$ neurons, length scale $\lambda = 0.2$. Right: Histogram of residuals between true and estimated distribution pooled across 10 runs of the model. The mean of errors is not different from zero ($P > 0.999$, two-sided one-sample t-test).

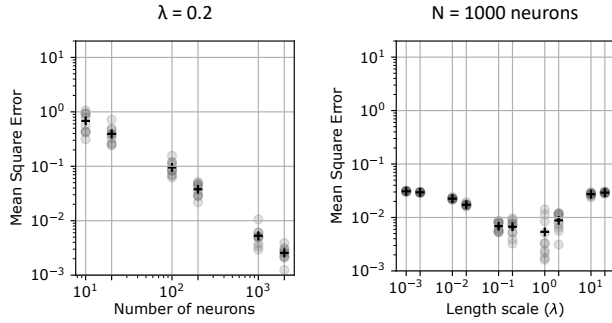


Figure 6: Representation error as a function of model parameters. Left: Representation error falls with the number of neurons in the model. Each gray dot represents one independent run of the model, and the black markers indicate the mean error across runs for that condition. Right: λ has a modest effect on representation error. Representation error is higher for both small and large values of λ . Intermediate values of λ result in more variable performance but generally lower error, indicated by a larger vertical spread of performance.

tween SSP representations, X^x and $X^{x'}$, approximates a quasi-kernel function, in this case the normalized sinc function, $X^x \cdot X^{x'} \approx \text{sinc}(\|x - x'\|) = \sin(\pi\|x - x'\|) / (\pi\|x - x'\|)$. Since the sinc function is an admissible kernel for kernel density estimation (Tsybakov, 2009, §1.3), we can view the dot product between an SSP encoded-vector and other vectors as approximating a probability. The argument is as follows:

Consider a randomly chosen unit vector, \mathbf{w}_i , of synaptic weights feeding into the i^{th} neuron of a network. The weights \mathbf{w}_i will have some similarity with X^x , although for randomly chosen \mathbf{w}_i , it is likely to be small. Consequently, we can consider any synaptic weight matrix as being the sum of a vector that is orthogonal to X^x , and one that is a sum of a (possibly empty) set of points encoded using the SSPs. That is, $\mathbf{w}_i = \frac{1}{n_i} \sum_{k=1}^{n_i} X^{x_k} + w_{\text{orthogonal}}$.

We can then consider the input current of any given neuron as being an approximation of the probability of the input

point, conditioned on a binary variable:

$$\begin{aligned} \mathbf{w}_i \cdot X^x &= \frac{1}{n_i} \sum_{k=1}^{n_i} X^x \cdot X^{x_k} + w_{\text{orthogonal}} \cdot X^x \\ &\approx \frac{1}{n_i} \sum_{k=1}^{n_i} \text{sinc}(\|x - x_k\|) \\ &\approx P(X = x | V_i) \end{aligned}$$

where the distribution conditioned on variable V_i is defined by the sinc kernel and the points $\{x_1, \dots, x_{n_i}\}$. If we assume a rectified linear neuron, $a_i = \text{ReLU}(\mathbf{w}_i \cdot X^x - b_i)$, with a bias, b_i chosen according to the method identified by Glad et al. (2003), then the activity of a neuron is exactly a probability. When we normalize the population's activities, $\hat{a}_i = a_i / \sum_j a_j$, we see that this population is conducting Bayesian inference on the variables, assuming a uniform prior over $P(V_i)$, as shown below.

$$\begin{aligned} P(V_i | X = x) &= \frac{P(X = x | V_i)P(V_i)}{P(X = x)} \\ &= \frac{P(X = x | V_i)P(V_i)}{\sum_j P(X = x | V_j)P(V_j)} \end{aligned}$$

If we let $P(V_i)$ be a non-informative prior, then we can remove it from the equation, yielding

$$P(V_i | X = x) = \frac{P(X = x | V_i)}{\sum_j P(X = x | V_j)} \approx \frac{a_i}{\sum_j a_j} = \hat{a}_i$$

which is the normalized neuron activity.

While the above analysis assumes a ReLU neuron, we note that LIF neurons, when averaged over time, produce an output that is fairly similar to a ReLU, other than the saturation behaviour. This saturation reduces the neuron output at high similarity values, but since it is still monotonically increasing (and non-negative), the LIF neuron's overall firing rate will still preserve the properties of the kernel function that make them suitable for density estimation.

Once normalized, we know that their firing rates will always be scaled between 0 and 1. Consequently, the firing rates of the neurons in the population can be used as the bases in a reproducing kernel Hilbert space (for a good short introduction see Ghogh et al., 2021). Thus, they can approximate the probability density function over the points, encoded in the activity of those neurons whose input weights are not orthogonal to the manifold $X^x, \forall x \in \mathbb{R}$. Our learning rule then simply learns the appropriate weights, α_i for the expression

$$P(X = x) = \sum_i^n \alpha_i k'(x, x_i) \quad (1)$$

where $k'(x, x_i) \approx \text{LIF}(\text{sinc}(\|x - x_i\|) - b_i)$, and x_i is a solution to $\arg \min_x \|1 - \mathbf{w}_i \cdot X^x\|_2^2$. Appealing to the representer theorem (Schölkopf et al., 2001), then there exists an optimal approximation of the probability distribution being learned, constrained by the kernel function induced by the SSP encoding, neural activation function, and the implicit collection

of points in the domain sampled by the synaptic weights. Depending on how the synaptic weights are chosen, better or worse approximations can be found.

Mismatch Negativity

The theoretical argument and simulations above establish that this system is able to represent multi-dimensional probability distributions, and then detect when a stimulus is a low-probability event. However, this is all based on the instantaneous input to the model. What if we also want to detect novel *temporal* patterns? This would be needed to, for example, respond to a stimulus being presented for an unexpected length of time, or at a different frequency. The well-known Mismatch Negativity signal is observed for exactly these sorts of novel stimuli. However, the model as presented so far, is only producing output based on the current input, and so cannot be sensitive to such temporal differences.

To address this problem, we need a way to take an input value that changes over time and convert it into a multi-dimensional value that represents the recent history of that signal. Fortunately, a method for doing this already exists: Voelker et al. (2019) presented a linear differential equation ($\frac{dm}{dt} = Am + Bx$) that converts an input signal x into a vector m which encodes the recent history of x as a set of coefficients of Legendre polynomials. This LMU (Legendre Memory Unit) and the associated LDN (Legendre Delay Network) can be implemented in spiking neurons, resulting in neural activity corresponding to Time Cells (Voelker et al., 2019), and has also been shown to out-perform LSTMs, GRUs, and Transformers on standard machine learning benchmarks that require temporal information (Chilkuri et al., 2021; Voelker et al., 2019).

With this in mind, we can construct a version of our model that responds to temporal signals by passing input data into an LDN to create the Legendre representation of the input, and then feeding that representation into the same novelty detection system defined above. As before, we do not need to change anything about the model to handle the increased dimensionality of the input.

The results for an input pattern that starts as a 1Hz signal, then switches to a 2Hz signal, and then back again are shown in Figure 7. We use a two-dimensional Legendre representation to encode the previous 2 seconds of the input (second graph). We also set the weight decay on the connection weights to $\tau = 5$ seconds. If we now feed the temporal pattern into the novelty detection system, the probability estimate (third graph) shows an increasing estimate of probability as the 1Hz signal becomes more familiar, and then a sudden drop in the probability estimate when it switches to a 2Hz signal, and then another drop when it returns to 1Hz. If we connect this probability estimate output as an inhibitory signal to a single spiking Leaky Integrate-and-Fire neuron (corresponding to the MBON $\alpha/3$ neuron), then we can see this neuron firing when the temporal pattern changes (fourth graph). Figure 8 shows the same result, but for a different input pattern

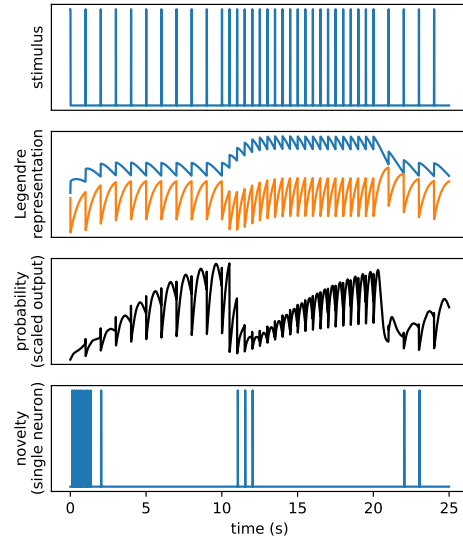


Figure 7: Detection of temporal novelty with the same circuit. The stimulus (first graph) is a regular pattern which changes frequency. The previous 2 seconds of the temporal input pattern is encoded using Legendre coefficients (second graph), and then fed into the same novelty detection model as before. When the output probability is low (third graph), novelty is detected (fourth graph).

with occasional outlier values.

Conclusions

We have demonstrated a simple single-hidden-layer neural network that learns to represent the probability distribution of its recent inputs. The output from this network can be used to detect novel inputs (the output is proportional to the learned probability, so it will be small if the learned probability for that input is low). Furthermore, the model works for different input dimensionalities and ranges, as inputs are converted into points on a D-dimensional sphere no matter what dimensionality those inputs are originally. The main parameter affecting performance is the chosen length scale λ , which does need to be tuned (Figure 6). Increasing the number of neurons improves performance, and the only other parameters are D (the dimensionality of the SSP space) and the neuron bias parameter, which controls the sparsity of the representation. Characterizing the effects of these parameters is ongoing.

Furthermore, we have extended this model to detecting temporal novelty as well, by exploiting a separate neural system (the LMU) to convert an input signal into a vector that represents the recent history of that signal. This gives a potential mechanism for detecting novelty that could trigger the observed Mismatch Negativity signal. This is somewhat surprising, in that our original model was inspired by the insect mushroom body system, while Mismatch Negativity is observed in mammals. Our ongoing work is to further investigate parallels between these two systems. However, it should

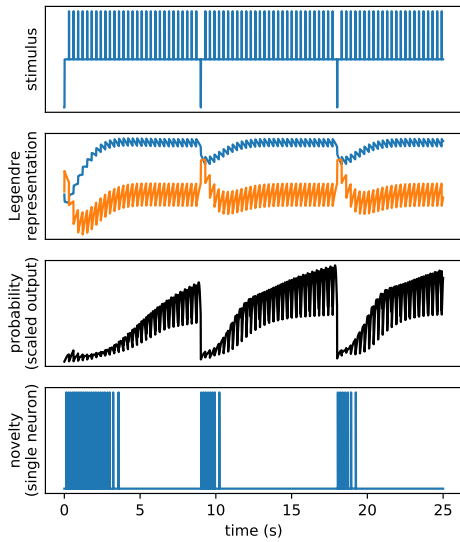


Figure 8: Detection of temporal novelty with the same circuit. The stimulus (first graph) is a regular pattern with occasional rare inputs (e.g. auditory tones of one frequency with occasional tones of a different frequency). The previous 2 seconds of the temporal input pattern is encoded using Legendre coefficients (second graph), and then fed into the same novelty decision model as before. When the output probability is low (third graph), novelty is detected (fourth graph).

be noted that the current model cannot directly explain the Mismatch Negativity signal, since nothing in the current system would generate a large and coherent EEG signal. That said, we do believe our model could act as a trigger telling the brain that a novel stimulus has occurred, which then leads to some other brain mechanism coming online which does generate the large change in electric field that is detected as Mismatch Negativity.

Acknowledgments

This project was supported by collaborative research funding from the National Research Council of Canada's Artificial Intelligence for Logistics program.

References

Chilkuri, N., Hunsberger, E., Voelker, A., Malik, G., & Eliasmith, C. (2021). Language modeling using lmus: 10x better data efficiency or improved scaling compared to transformers. *arXiv preprint*.

Dasgupta, S., Sheehan, T., Stevens, C., & Navlakha, S. (2018). A neural data structure for novelty detection. *Proceedings of the National Academy of Sciences*, 115, 201814448.

Frady, E. P., Kanerva, P., & Sommer, F. T. (2018). A framework for linking computations and rhythm-based timing patterns in neural firing, such as phase precession in hip-

pocampal place cells. *Conference on Cognitive Computational Neuroscience*, 1–5.

Georgopoulos, A. P., Kalaska, J. F., Caminiti, R., & Massey, J. T. (1982). On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *The Journal of Neuroscience*, 2, 1527–1537.

Ghojogh, B., Ghodsi, A., Karray, F., & Crowley, M. (2021). Reproducing kernel Hilbert space, Mercer's theorem, eigenfunctions, Nyström method, and use of kernels in machine learning: Tutorial and survey. *arXiv preprint arXiv:2106.08443*.

Glad, I. K., Hjort, N. L., & Ushakov, N. G. (2003). Correction of density estimators that are not densities. *Scandinavian Journal of Statistics*, 30(2), 415–427.

Lu, T., Voelker, A. R., Komer, B., & Eliasmith, C. (2019). Representing spatial relations with fractional binding. *Proc. of the Cognitive Science Society*.

Pazo-Alvarez, P., Cadaveira, F., & Amenedo, E. (2003). MMN in the visual modality: A review. *Biological Psychology*, 63(3), 199–236.

Plate, T. A. (1992). Holographic recurrent networks. In S. Hanson, J. Cowan, & C. Giles (Eds.), *Advances in neural information processing systems*. Morgan-Kaufmann.

Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6, 623–641.

Schölkopf, B., Herbrich, R., & Smola, A. J. (2001). A generalized representer theorem. *Computational Learning Theory: 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001 Amsterdam, The Netherlands, July 16–19, 2001 Proceedings 14*, 416–426.

Schwartz, A. B., Kettner, R. E., & Georgopoulos, A. P. (1988). Primate motor cortex and free arm movements to visual targets in three-dimensional space. i. relations between single cell discharge and direction of movement. 8, 2913–2927.

Stöckel, A., Stewart, T. C., & Eliasmith, C. (2021). Connecting biological detail with neural computation: Application to the cerebellar granule-golgi microcircuit. *Topics in Cognitive Science*, 13(3), 515–533.

Tsybakov, A. B. (2009). *Introduction to nonparametric estimation*. Springer.

Voelker, A. R., Kajić, I., & Eliasmith, C. (2019). Legendre memory units: Continuous-time representation in recurrent neural networks. *Advances in Neural Information Processing Systems*, 15544–15553.

Efficient Memory Encoding Explains the Interactions Between Hippocampus Size, Individual Experience, and Clinical Outcomes: A Computational Model

Andrea Stocco (stocco@uw.edu)

Department of Psychology, University of Washington, Seattle, WA 98195 USA

Briana M. Smith (brianam2@uw.edu)

Department of Bioengineering, University of Washington, Seattle, WA 98195 USA

Bridget Leonard (bl1313@uw.edu)

Department of Psychology, University of Washington, Seattle, WA 98195 USA

Holly Sue Hake (hakehs@uw.edu)

Department of Psychology, University of Washington, Seattle, WA 98195 USA

Abstract

The relationship between hippocampal volume and memory function has produced mixed results in neuroscience research. However, an experience-dependent efficient encoding mechanism underlies these varied observations. We present a model that utilizes an autoencoder to prioritize sparseness and transforms the recurrent loop between the cortex and hippocampus into a deep neural network. We trained our model with the Fashion MNIST database and a loss function to modify synapses via backpropagation of mean squared recall error. The model exhibited experience-dependent efficient encoding, representing frequently repeated objects with fewer neurons and smaller loss penalties and similar representations for objects repeated equally. Our findings clarify perplexing results from neurodevelopment studies: linking increased hippocampus size and memory impairments in ASD to decreased sparseness, and explaining dementia symptoms of forgetting with varied neuronal integrity. Our findings propose a novel model that connects observed relationships between hippocampus size and memory, contributing to the development of a larger theory on experience-dependent encoding and storage and its failure.

Keywords: Memory; Hippocampus; Autoencoder; Autism Spectrum Disorder; Alzheimer's Disease; Computational models; ACT-R

The hippocampus is a region of the medial temporal lobe that is critical for long-term memory storage and retrieval. The size of the hippocampus can vary significantly between individuals and these variations in size have been associated with corresponding differences in memory function (Pohlack et al. 2014; Hardcastle et al. 2020; Botdorf, Canada, & Riggins 2022). The relationship between hippocampus size and memory function, however, is complex and not always straightforward. On one hand, there is evidence that greater hippocampus volume is associated with better memory function. For example, greater hippocampus volume is associated with better spatial memory performance in a laboratory task (Erickson et al. 2011; Guderian et al. 2015). Conversely, reduced hippocampus size is associated with significant impairments in long-term memory. For example, in frontotemporal dementia and Alzheimer's disease, neuronal loss results in markedly reduced hippocampal volume, and the degree of

volume loss positively correlates with the severity of amnesic symptoms (Dickerson et al. 2009).

The relationship between hippocampus size and memory performance is also, at least partially, mediated by experience. A notable case is the fact that London taxi-cab drivers have larger hippocampi than the normal population, likely due to the amount of information that cab drivers need to memorize ("The Knowledge") to pass the license test (Maguire et al. 2000). In fact, a follow-up study revealed that changes in hippocampus size follow, and do not precede, the amount of studying necessary to pass the test (Maguire, Woollett, & Spiers 2006). Similarly, changes in hippocampus size correlate with an individual's years of education (Nobis et al. 2019).

An intuitively appealing explanation for these effects might be that the hippocampus grows with the amount of data it needs to, or can, store. Thus, pressure to store more information results in the growth of the hippocampus, and a reduction in hippocampal size results in loss of memory. This simple explanation, however, is complicated by a number of other findings. Reductions in hippocampus size are observed in a variety of mental disorders, including post-traumatic stress disorder (PTSD) and anxiety. In these cases, significant reductions in hippocampal size are *not* accompanied by corresponding changes in long-term memory function (Karl et al. 2006). Conversely, larger hippocampal volume has been observed in autism spectrum disorder (ASD), where a corresponding increase in memory function was not observed (Varghese et al., 2017). In fact, evidence suggests that the prevalence of amnesic forms of dementia in ASD is up to four times higher than the neurotypical average, despite the fact that greater hippocampus size could have represented a buffering factor against neuronal loss (Fyfe, 2021). Thus, while it has been shown that experience drives changes in hippocampus size, changes have also been observed in clinical conditions *without* corresponding changes in memory.

At least three possible explanations can be proposed to reconcile these findings. The first and most mundane is that changes in hippocampus size might not always reflect

underlying changes in the number of hippocampal cells or synapses. Virtually all of the studies assess hippocampal size through anatomical MRI, and the sheer volume of a region in an MRI scan can be affected by a variety of other factors, such as greater water density (Bansal et al. 2013).

A second explanation is that two or more biological mechanisms might be at play. Thus, while experience-dependent growth following intense memory training and dementia-related loss of memory function are connected to the number of cells and synapses, changes in other clinical domains might be related to other processes. For example, prolonged stress exposure causes neuronal death through the accumulation of cortisol. Thus, it is possible that the volume loss in PTSD and anxiety are due to cortisol-related pruning, which does not play a role in dementia or ASD (Kim, Pellman, & Kim 2015).

The third and last explanation is that these varied phenomena are indeed connected by experience-dependent efficient allocation of hippocampal cells and synapses to varying memory demands, but that this relationship is complex and non-linear.

In this paper, we put forward a neurocomputational framework that provides a possible account for the latter hypothesis. According to this framework, the need to store and retrieve memories demands an efficient allocation of neural resources, and the principles underlying this allocation can be understood in terms of information theory.

The remainder of the paper is structured as follows. First, we review previous computational attempts to model the relationship between memory function and hippocampus size. Specifically, we review a model that explicitly links resource allocation in the hippocampus to the information entropy of its memories, and how entropy is altered in PTSD. Second, we propose a possible neural network model of how such changes could happen. The model, based on the autoencoder architecture, shows that, under realistic conditions, the hippocampus can spontaneously learn to allocate neurons adaptively according to the demands. Finally, we speculate on the implications of this mechanism for two important memory-related phenomena, sleep, and spontaneous brain activity,

Previous Models

To the best of our knowledge, the first computational account of the relationship between memory demands and hippocampus size was put forward by Smith et al (2021). The authors proposed a mathematical model of memory storage and retrieval based on information entropy.

The model is based on the framework originally proposed by Anderson and Schooler (1991) and currently implemented in the ACT-R architecture (Anderson, 2009). According to this framework, each memory is a collection of *traces*, each corresponding to a specific episode in which the memory's contents were encoded. This makes the model broadly consistent with the Multiple Trace Theory of memory (Moscovitch et al. 2005). The strength of each trace decays over time according to a power function. The *memory's* total strength, or *activation*, is the log of the sum

of its traces. Thus, if a memory m is made of n traces encoded at times $t_1, t_2 \dots t_n$, its activation at time t is:

$$A(m, t) = \log \sum_i (t - t_i)^{-d}$$

where d is an individual-specific decay rate (Sense et al., 2016; Zhou et al, 2021). Note that Equation 1 naturally captures the effects of recency (through the decay term d) and frequency (through the accumulation of traces). The probability $P(m)$ of retrieving a memory can be computed as a function of its activation, relative to all other memories:

$$P(m) = e^{-A(m, t)} / \sum_j e^{-A(j, t)} \quad (1)$$

Smith et al. (2021) proposed that the distribution of probabilities across memories could be used to predict changes in hippocampus volume. The authors assumed that the hippocampus would use efficient coding, and allocate fewer resources to store information that is most likely to be retrieved. This is a common principle in lossless compression algorithms (Huffman, 1952). Consider, for example, the problem of efficiently encoding the quote "All those moments will be lost in time like tears in rain". Using standard ASCII coding, each character in the string would be represented by 8 bits and the entire string would take a total of 456 bits. To *efficiently* encode the string, however, one would first count the occurrence of each character in the string and then proceed to assign the shortest possible code to the most common character, the second shortest to the second, and so on. In this case, the letter "e", "i", and "l", which appeared six times each, would be assigned the three-bit codes 001, 010, and 011, while the letter "k", which appears only once, would be given the six-bit code 101001. This would result in the entire string being encoded with only 206 bits.

We currently do not know with sufficient precision how information is encoded in the hippocampus. However, independently of the specific code, the degree of compression allowed by any adaptive scheme of this sort is functionally related to the information entropy H of the data:

$$H = -\sum_i p(i) \log p(i) \quad (2)$$

Smith et al. (2021) showed that the reduced hippocampus size in individuals suffering from PTSD could be predicted by calculating the entropy of the retrieval probabilities (Equation 2) associated with every memory in the model. Specifically, when the model was modified to simulate emotional trauma, the persistence of intrusive memories had a significant effect on the probability distribution of the memories that could be retrieved. The more likely the intrusive memory was to be retrieved, the lower the entropy of the model's memory system, and as a correlate, the lower the volume of the hippocampus.

Limits of the Model

The original model by Smith et al (2021) was noteworthy but did not address a number of limitations. First, it provided no biological mechanisms by which neurons could be efficiently allocated to different representations. In fact, it could not solve the problem of how the hippocampus could

form an efficient engram without knowing in advance its future activation level.

A second limitation of the original model was its scope: it only addressed changes in hippocampal volume due to PTSD. The same framework can be arguably applied to anxiety, which shares with PTSD the transdiagnostic symptom of intrusive thoughts that are ruminated upon. It could be possibly extended to include experience-dependent changes as well (such as the effects of education). It does *not* address, however, other findings, such as the greater hippocampus volume in ASD and the association between smaller hippocampus size and memory loss in dementia.

A Neural Network Model of the Hippocampus

To address these limitations, we examined the behavior of a neural network model of the hippocampus and conducted a series of simulations to test whether (a) Efficient coding spontaneously emerges in more biological models, and (b) Whether the model can account for the diversity of findings relating hippocampus size and memory function.¹

The connections between the cortex and the hippocampus form a recurrent loop. The exact set of synapses varies slightly across regions; as an example, this paper will consider the connectivity between the inferior temporal lobe and the hippocampus. This specific circuit is well understood and underlies memory for higher-level visual objects, which will be used as experimental stimuli. Projections from the inferior temporal cortex pass through the entorhinal cortex and the dentate gyrus before reaching area CA3 of the hippocampus, which is considered the initial seat of an engram (Tonegawa et al., 2015).

During recall, memories are then reactivated in the cortex (Danker and Anderson, 2010) through a series of connections that originate in CA3 and progress through area CA1, the entorhinal cortex again, and finally return to the temporal cortex.

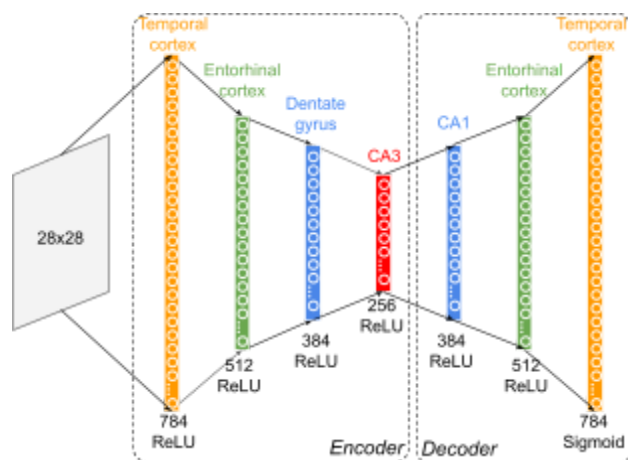


Figure 1: Architecture of the neural network model

For convenience, this recurrent loop can be “unrolled” and transformed into a feedforward deep neural network with multiple layers. The first half of the model corresponds to the neural populations encountered from the cortex to the hippocampus, and the second half to the neurons encountered from the hippocampus back to the cortex. In this design, the cortex is both the input and the output of the network, and the hippocampus is the central bottleneck. This architecture is technically known as an *autoencoder* (Kramer, 1991) and is used, in deep-learning applications, to learn a set of features that would efficiently compress the original input so that its output is minimally different from its input. To a large extent, the application of autoencoders can be construed as exactly the function of episodic memory and, by extension, of the hippocampus.

The model’s final architecture is shown in Figure 1. Its input is a 28×28 matrix that contains a visual representation of an object. This representation is then flattened to a layer of 784 neurons, which represents the object as encoded in the inferior temporal cortex. This representation is passed through a smaller layer of 512 neurons, representing the entorhinal cortex, and an even smaller one of 384 neurons, representing the dentate gyrus. It finally reaches a layer of 256 neurons that hold a compression representation of the original content and stands for the hippocampus’ CA3 field. The output of the hippocampus is then passed through a mirror series of layers representing CA1, the entorhinal cortex, and the temporal cortex again (784 neurons), generating a reconstructed version of the original stimulus. All of the neurons in the model are Rectified Linear Units (ReLUs), with the exception of the very last layer, which uses a sigmoid function to ensure that all of the predicted pixel values are, like the inputs, between 0 and 1.

The model was used in five different simulations, each of which addresses a different facet of the relationship between hippocampus size and memory function.

Materials and Methods

Model Implementation

The model was implemented in Keras with an underlying TensorFlow engine. In addition to those of Figure 1, the model contains four additional layers that perform purely technical operations such as reshaping inputs and outputs and computing penalty terms for the cost functions (see below); although necessary, these layers are not functionally relevant. Altogether, the model has a total of 1,347,282 trainable parameters.

Training and Testing Data

The model was trained on a selection of objects from the Fashion MNIST database (Xiao, Rasul, & Vollgraf, 2017), a collection of 70,000 28×28 black-and-white images from 10 clothing categories. A subset of 1,111 images was randomly selected at every run. The images were repeated with varying frequencies across simulations (see below) but always formed a training set of 4,000 stimuli.

¹ All data and code are available at <https://osf.io/wxh2r/>

Model Training and Loss Function

In all simulations, the model was trained on all of the training set items for five consecutive epochs using stochastic gradient descent with adaptive moment estimation (Adam: Kingma & Ba, 2014). We used a combined loss function L that included two terms:

$$L = \sum_{i,o}^N (y_i - y_o)^2 / N + \lambda \sum_{h \in CA3} |y_h|$$

The first term is the *accuracy cost* of the network's recall function, and is the mean squared difference between the activations y_i and y_o of each input neuron i and corresponding output neuron o . The second term is the *resource cost* and is the sum of the activation y_h of each hippocampal neuron h in the CA3 layer. (Note that the penalty cost only interests the CA3 neurons). The hyperparameter λ regulates the weight of the penalty and was set to 0.00001 throughout these simulations; pilot tests showed that the results did not qualitatively change for different λ values, as long as λ was below a critical threshold of 0.0001, above which the penalty became too severe.

Note that the resource cost penalty is equivalent to the L1 penalty used in regularization methods, such as LASSO (Tibshirani, 1996). Unlike other penalties, the L1 penalty can force its terms to zero, thus reducing the number of active neurons within a representation.

Dependent Variables

For each of the 1,111 objects in the training set, four dependent variables were computed. Two variables measured the sparseness of the hippocampus representation: the value of the resource cost *L1 penalty* and the total *number of active neurons* (that is, with activation $y_h \neq 0$) in the hippocampus. The other two variables measured the model's recall accuracy, and they were the value of the *error penalty*, i.e. the squared sum of differences between target and predicted activations in the output layer, $\sum_{i,o} (y_i - y_o)^2$, and the Pearson *correlation coefficient* between the encoded and recalled image. Because of their constrained range, correlation coefficients were normalized using Fisher's r -to- Z transform: $Z = [\log(1 + r) - \log(1 - r)] / 2$.

Results

Simulation 1: Emergence of Efficient Coding

In the first simulations, the set of 1,111 objects was used to create a 4,000-item training set in which different objects were repeated with different frequencies. Specifically, 1,000 objects occurred only once; 100 objects occurred 10 times, 10 objects occurred 100 times, and a single image occurred 1,000 times. If the model is learning a form of efficient coding, the internal hippocampal representation of an object should depend on its frequency in the training set, and, therefore, objects that are repeated the most should have representations with fewer neurons and smaller L1 penalties than objects that are repeated the least.

Figure 2 illustrates the results of one such simulation. The top row shows four example objects from one specific simulation, chosen from the sets of stimuli repeated 1, 10,

100, or 1,000 times, respectively. The middle row represents the corresponding responses of the simulated CA3 layer, with the activations of its 256 neurons arranged in a 16x16 grid. The dependent variables for hippocampus sparseness (L1 penalty and number of neurons) are also reported. Finally, the bottom row depicts the recalled memory.

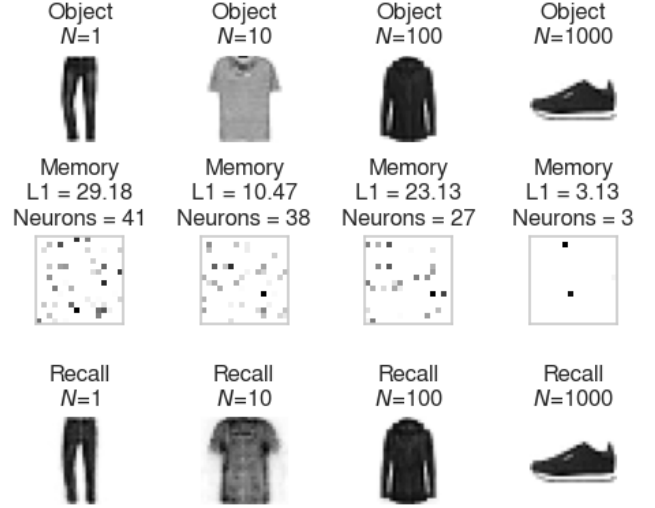


Figure 2: (Top) Four example stimuli that were repeated 1, 10, 100, or 1,000 times in the training set. (Middle) Corresponding CA3 representations of the stimuli; (Bottom) Recalled stimuli reconstructed by the decoder from the CA3 representations.

Although representative, Figure 2 only illustrates four examples from a single run. A complete overview of all simulations is instead reported in Figure 3, where the mean L1 penalty and the mean number of neurons are reported as the blue lines in the two panels. As the figure shows, higher frequency results in a dramatic reduction in the number of neurons needed to represent an object.

This change in representation has no consequences for the model, which has a fixed and immutable structure in which all synapses exist all the time, even when they are connected to silenced neurons. In a biological hippocampus, however, synapses and neurons change over time: synapses with a value of zero are non-existent, and those connected to mute cells would simply be pruned. Thus, the sparser representation in Figures 2 and 3 could be associated with changes in hippocampus size.

But how closely does the reduction in the CA3 representations match the predictions of information theory? According to Huffman (1952), efficient codes are such that the length of a code for an object x matches its information content $I(x)$, which is the negative log of its probability: $I(x) = -\log_2 p(x)$. The value of $p(x)$ can be calculated from the number of occurrences of stimulus x in the training set. Figure 4 compares the relationship between the number of neurons used to encode an object in the CA3 layer and its corresponding information content. As the figure shows, the number of neurons closely mirrors ($r = .97$) the information content, a hallmark of efficient coding.

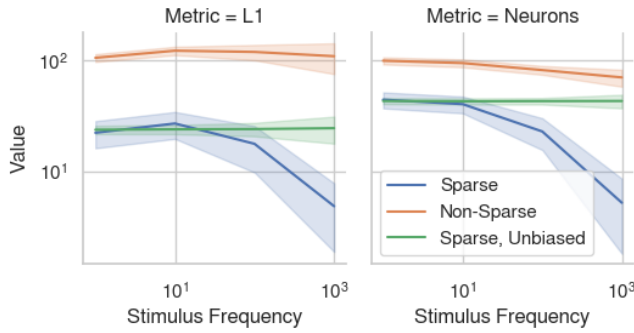


Figure 3: A summary overview of sparseness metrics (left: L1 penalty; Right: Number of neurons) across stimuli of different frequencies, averaged over 50 simulations. Colors represent different model conditions (Blue = Sparse; Red = Non-Sparse; Green = sparse with an unbiased training set). Lines and ribbons represent means \pm SD.

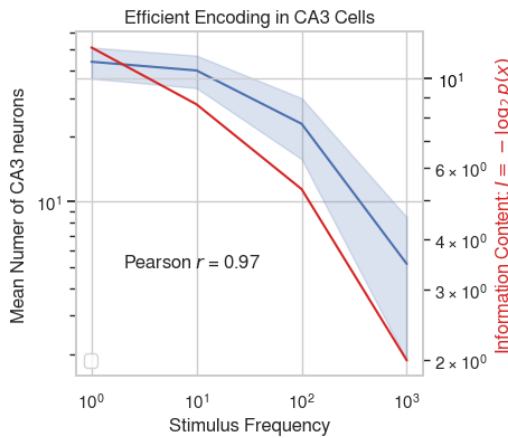


Figure 4: Relationship between the number of neurons that encode a stimulus (blue line and ribbon, representing mean \pm SD) and its information content (red line).

Simulation 2: Frequency Drives Efficient Coding

To ensure that the effect is driven by frequency and not by other confounding factors, a second series of simulations were run. In these series, the model was trained with a dataset of identical size (4,000 items) and containing the same 1,111 objects but with an *unbiased* number of repetitions per object, i.e., with each object being repeated 3 or 4 times. The results of these simulations are shown as the green line in Figure 3. Under these conditions, both the L1 penalty and the number of neurons remain invariant and equal to the values of the least-frequent memories in Simulation 1. Note that, for the unbiased training model, the frequencies on the x -axis do not actually reflect the frequencies of the simulation training set; instead, they are used to identify the corresponding group of objects in Simulation 1.

The results of this simulation could be used to explain the increased hippocampal volume in London taxi cab drivers compared to bus drivers (Maguire et al., 2006): as noted by Smith et al. (2021), taxi cab drivers, unlike bus drivers, have

to rehearse the streets of London with comparable frequency to prepare for the license test.

Simulation 3: Enlarged Hippocampus in ASD

In the model, sparseness is achieved by adding a penalty to the loss function. In a biological network, however, sparseness must be achieved through some neural mechanism. The most straightforward candidate is lateral inhibition, that is, inhibitory synapses between neurons belonging to the same region. Inhibitory synapses typically express GABA receptors, and abnormally low expression of GABA receptors is a key characteristic of ASD (Cellot & Cherubini, 2014), one of the disorders also characterized by abnormalities in hippocampus size. Recent studies estimate that individuals with ASD express as much as 40% fewer GABA receptors than healthy controls. Thus, we hypothesized that the reduced availability of GABA receptors in ASD may lead to a decrease in lateral inhibition, resulting in less efficient coding and thus the larger hippocampus observed in ASD.

To test this hypothesis, a series of simulations were carried out using the biased training set but with the λ parameter set to $\lambda=0$, allowing for a minimum amount of sparseness based solely on the thresholds of the ReLU units. The results of the simulations with such as Non-Sparse model are shown in the red lines of Figure 2. As it can be seen, without the resource cost penalty, the model now uses a disproportionately large number of neurons and incurs in large L1 penalties. Furthermore, both measures remain remarkably stable even when encoding extremely high-frequency stimuli, indicating that the hippocampus is not using efficient coding.

Simulation 4: Hippocampal Damage in Dementia

As noted in the introduction, some reductions in hippocampus size *are* associated with distinctive deficits in memory. This is the case, for example, of neurodegenerative diseases such as Alzheimer's Disease. In these cases, neuronal loss afflicts long-term memory by harming the engram associated with a specific memory.

To simulate the effects of dementia, we ran a fourth series of simulations, identical in nature to Simulation 1 but with an additional manipulation. After completing the training phase, the model's hippocampus was artificially damaged by applying a binary mask to the activation of its units. Binary masks were generated by creating a null vector of 256 elements, and randomly setting a percentage of its elements to 1. The proportion of units set to 1 represents the *neuronal integrity* of the hippocampus and was parametrically varied from 0.1 to 0.9. After every simulated lesion, the model's recall was tested again, and the two accuracy measures (squared recall error and recall correlation) were recorded. Figure 5 illustrates these results.

Interestingly, and consistent with the observed symptoms of dementias, less frequent memories are more affected, even at higher levels of neuronal integrity, than the more frequent ones, which remain comparatively well preserved even at lower levels of neuronal integrity.

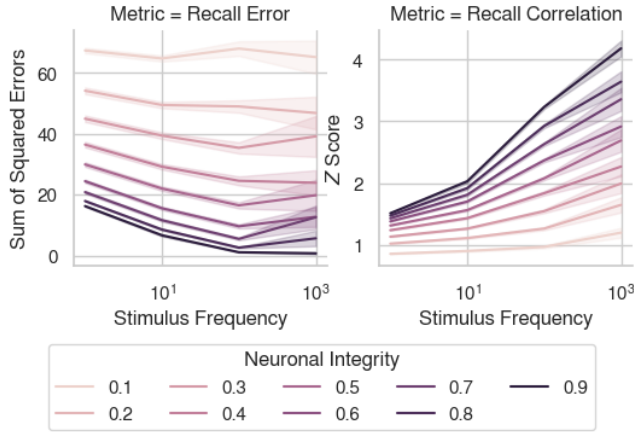


Figure 5: Effects of neural damage on recall accuracy. Lines and ribbons indicate means \pm SD.

Simulation 5: Interactions Between ASD and Neurodegenerative Disorders

Although ASD *per se* is not associated with notable changes in long-term memory function, it has been noted that dementia has a higher prevalence in individuals with ASD than in neurotypical controls (Fyfe, 2021). The results of Simulations 3 and 4 suggest that an additional advantage of efficient coding of memories is to buffer against neuronal death. Conversely, the less sparse memory representations in ASD might be more susceptible to damage from neuronal loss, thus explaining the greater prevalence of dementia in ASD. To test this hypothesis, we conducted a second series of lesion simulations, identical to the ones in Simulation 4 but with the model's λ parameter set to $\lambda = 0$. Because we are especially interested in the earlier stages of neurodegenerative disease, only the neuronal integrity values from 0.5 to 0.9 were examined. The results are summarized in Figure 6. As the figure shows, the Non-Sparse model is consistently more affected than the sparse model by damage, across all levels of stimulus frequency and neuronal integrity.

Discussion

This paper has dealt with the relationship between hippocampus size and memory functions across clinical and neurotypical populations. Specifically, it has shown that some puzzling findings in the literature can be reconciled when one analyzes the behavior of a neural network model of the hippocampus whose loss function includes a resource cost. The resource cost penalty induces sparseness in a form that is consistent with the principles of efficient coding and with the idea, first proposed by Smith et al. (2021), the hippocampus size reflects the information content of the stored memories. These contributions notwithstanding, a number of limitations must be acknowledged. First and foremost, the model uses an autoencoder architecture, while hippocampus models are more commonly implemented as autoassociators (e.g., Treves & Rolls, 1994).

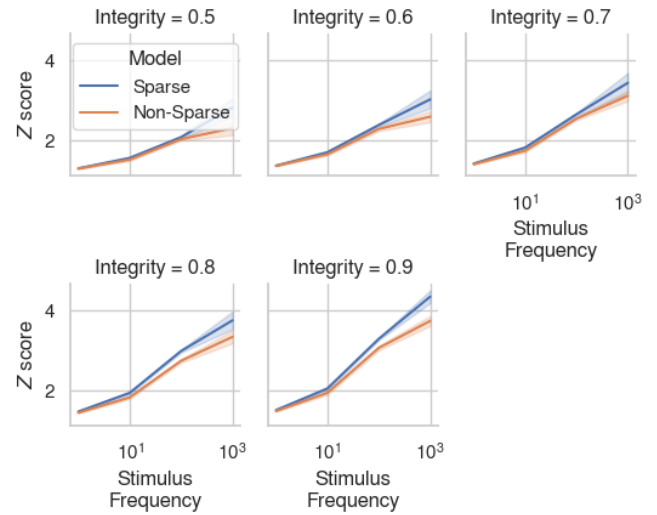


Figure 6. The Z-scored recall correlation coefficients of Sparse (blue) and Non-Sparse (red) models. Line and ribbons indicate means \pm SD.

As a consequence, the model requires error-driven methods to learn properly and is incapable of “one-shot” Hebbian learning. Autoencoders were chosen because they make it easier to capture the dynamics of encoding and recall and the relationship between cortical areas and the hippocampus. A proper model, however, should attempt to combine both architectures and include principles of auto-associative Hebbian learning with the hippocampus.

Second, a number of factors that affect the model's memory recall and performance are left unexplored. Among those, perhaps the most important is the role played by the number of epochs used in training. It is possible, for example, that sparse models would require longer epochs to achieve the same recall accuracy. The combined use of error-driven learning and multiple training epochs highlights another aspect of the model, namely, its need for multiple learning passes to discover efficient representations. As noted in the introduction, the hippocampus cannot assign efficient memory codes right away, as they require knowledge of an object's frequency. In the autoencoder, it is the presence of multiple learning passes and gradient descent that pushes for sparser and more efficient coding. It is possible that spontaneous brain activity, which is prominently displayed in the hippocampus at rest and during sleep (Pfeiffer, 2020), provides a biological surrogate for the necessary re-experience of memories that are needed for efficient coding.

Lastly, the model is silent about the nature of forgetting, another prominent feature of memory that might be connected to the spontaneous replay of memories at rest (Zhou et al., 2021). Future research will be needed to further explore the nature of these processes within the model.

References

- Anderson, J. R. (2009). *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press.
- Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, 2(6), 396-408.
- Bansal, R., Hao, X., Liu, F., Xu, D., Liu, J., & Peterson, B. S. (2013). The effects of changing water content, relaxation times, and tissue contrast on tissue segmentation and measures of cortical anatomy in MR images. *Magnetic Resonance Imaging*, 31(10), 1709-1730.
- Botdorf, M., Canada, K. L., & Riggins, T. (2022). A meta-analysis of the relation between hippocampal volume and memory ability in typically developing children and adolescents. *Hippocampus*, 32(5), 386-400.
- Cellot, G., & Cherubini, E. (2014). GABAergic signaling as therapeutic target for autism spectrum disorders. *Frontiers in Pediatrics*, 2, 70.
- Dickerson, B. C., Bakkour, A., Salat, D. H., Feczko, E., Pacheco, J., Greve, D. N., ... & Buckner, R. L. (2009). The cortical signature of Alzheimer's disease: regionally specific cortical thinning relates to symptom severity in very mild to mild AD dementia and is detectable in asymptomatic amyloid-positive individuals. *Cerebral Cortex*, 19(3), 497-510.
- Erickson, K. I., Voss, M. W., Prakash, R. S., Basak, C., Szabo, A., Chaddock, L., ... & Kramer, A. F. (2011). Exercise training increases size of hippocampus and improves memory. *PNAS*, 108(7), 3017-3022.
- Fyfe, I. (2021). Early-onset dementia in autism spectrum disorder. *Nature Reviews Neurology*, 17(10), 595-595.
- Guderian, S., Dzieciol, A. M., Gadian, D. G., Jentschke, S., Doeller, C. F., Burgess, N., ... & Vargha-Khadem, F. (2015). Hippocampal volume reduction in humans predicts impaired allocentric spatial memory in virtual-reality navigation. *Journal of Neuroscience*, 35(42), 14123-14131.
- Hardcastle, C., O'Shea, A., Kraft, J. N., Albizu, A., Evangelista, N. D., Hausman, H. K., ... & Woods, A. J. (2020). Contributions of hippocampal volume to cognition in healthy older adults. *Frontiers in Aging Neuroscience*, 12, 593833.
- Huffman, D. A. (1952). A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9), 1098-1101.
- Karl, A., Schaefer, M., Malta, L. S., Dörfel, D., Rohleder, N., & Werner, A. (2006). A meta-analysis of structural brain abnormalities in PTSD. *Neuroscience & Biobehavioral Reviews*, 30(7), 1004-1031.
- Kim, E. J., Pellman, B., & Kim, J. J. (2015). Stress effects on the hippocampus: a critical review. *Learning & Memory*, 22(9), 411-416.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv preprint arXiv:1412.6980*.
- Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2), 233-243.
- Maguire, E. A., Gadian, D. G., Johnsrude, I. S., Good, C. D., Ashburner, J., Frackowiak, R. S., & Frith, C. D. (2000). Navigation-related structural change in the hippocampi of taxi drivers. *PNAS*, 97(8), 4398-4403.
- Maguire, E. A., Woollett, K., & Spiers, H. J. (2006). London taxi drivers and bus drivers: a structural MRI and neuropsychological analysis. *Hippocampus*, 16(12), 1091-1101.
- Moscovitch, M., Rosenbaum, R. S., Gilboa, A., Addis, D. R., Westmacott, R., Grady, C., ... & Nadel, L. (2005). Functional neuroanatomy of remote episodic, semantic and spatial memory: a unified account based on multiple trace theory. *Journal of Anatomy*, 207(1), 35-66.
- Nobis, L., Manohar, S. G., Smith, S. M., Alfaro-Almagro, F., Jenkinson, M., Mackay, C. E., & Husain, M. (2019). Hippocampal volume across age: Nomograms derived from over 19,700 people in UK Biobank. *NeuroImage: Clinical*, 23, 101904.
- Pfeiffer, B. E. (2020). The content of hippocampal "replay". *Hippocampus*, 30(1), 6-18.
- Pohlack, S. T., Meyer, P., Cacciaglia, R., Liebscher, C., Ridder, S., & Flor, H. (2014). Bigger is better! Hippocampal volume and declarative memory performance in healthy young men. *Brain Structure and Function*, 219, 255-267.
- Smith, B. M., Thomasson, M., Yang, Y. C., Sibert, C., & Stocco, A. (2021). When fear shrinks the brain: A computational model of the effects of posttraumatic stress on hippocampal volume. *Topics in Cognitive Science*, 13(3), 499-514.
- Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267-288.
- Tonegawa, S., Pignatelli, M., Roy, D. S., & Ryan, T. J. (2015). Memory engram storage and retrieval. *Current Opinion in Neurobiology*, 35, 101-109.
- Treves, A., & Rolls, E. T. (1994). Computational analysis of the role of the hippocampus in memory. *Hippocampus*, 4(3), 374-391.
- Varghese, M., Keshav, N., Jacot-Descombes, S., Warda, T., Wicinski, B., Dickstein, D. L., ... & Hof, P. R. (2017). Autism spectrum disorder: neuropathology and animal models. *Acta Neuropathologica*, 134, 537-566.
- Vivanti, G., Tao, S., Lyall, K., Robins, D. L., & Shea, L. L. (2021). The prevalence and incidence of early-onset dementia among adults with autism spectrum disorder. *Autism Research*, 14(10), 2189-2199.
- Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *ArXiv preprint arXiv:1708.07747*.
- Zhou, P., Sense, F., van Rijn, H., & Stocco, A. (2021). Reflections of idiographic long-term memory characteristics in resting-state neuroimaging data. *Cognition*, 212, 104660.

A Cognitive Model of the Effects of Workload on Perceptual Span

Garrett Swan (gswan@aptima.com)

Aptima, Inc
Woburn, 01801 USA

Christopher A. Stevens (christopher.stevens.28@us.af.mil)

Air Force Research Laboratory
Wright Patterson AFB, OH USA

Samantha Klosterman (samantha.klosterman@ballaerospace.com)

Ball Aerospace
Fairborn, OH 45324

Abstract

Understanding how individuals deploy attention in multitasking environments helps us develop models that more accurately capture human performance and variability. Here, we implemented a method of measuring subjective workload in an ACT-R model and constrained the model's ability to use bottom-up capture for stimuli outside of a peripheral window (i.e., perceptual span). Stimuli outside of the perceptual span window could thus only be detected via top-down attention. Our subjective workload metric was based on event-frequency and was compared to NASA-TLX reports from multitasking data using the AF-MATB in Bowers, Christensen, and Eggemeier (2014). The metric successfully differentiated between Easy and Hard task demands. We then evaluated performance and eye movements of an ACT-R model with different fixed levels of perceptual span. As expected, when the model was limited to mostly top-down visual attention, performance declined because the model could not directly attend to malfunctions in peripheral vision. Similarly, saccade amplitude decreased and eye movements became more systematic. Interestingly, when comparing the model's simulation to behavioral data, the size of the perceptual span window increased as task demands increased, suggesting that participants were using less systematic scans when subjective workload increased. We then implemented this transition in the ACT-R model.

Keywords: Workload, Perceptual Span, ACT-R, AF-MATB

Introduction

Bottom-up (stimulus driven) and top-down (internal) factors affect visual attention (Carrasco, 2011). Implementing these factors in cognitive architectures, such as Adaptive Control of Thought – Rational (ACT-R; Anderson et al., 2004), is worthwhile because it enables simulations to more accurately capture human variability and performance. For example, the Pre-attentive and Attentive Vision (PAAV) module for ACT-R added pre-attentive bottom-up mechanisms to better emulate the cognitive processes that result in differing search slopes in simple and conjunction vision search tasks (Nyamsuren & Taatgen, 2013). Here, we are interested in modeling how task demands change how individuals deploy visual attention given the importance of performance in high workload situations.

Generally, as the experienced level of subjective workload increases, there are changes to the deployment of visual attention. The amount of information that can be perceived in a single fixation, coined as perceptual span or useful field of view, decreases as workload increases (Bertera & Rayner,

2000; Young & Hulleman, 2013). This could contribute to the increased bias for attentional control to use systematic search strategies in high workload situations (Carrasco, Evert, Chang, & Katz, 1995; Carrasco & Yeshurun, 1998; Pomplun, Garaas, & Carrasco, 2013) and related to the phenomenon of cognitive tunneling (Thomas & Wickens, 2001).

Here, we are interested in modeling how changes in task demands affect the visual attention of a cognitive model in a complex multitasking environment, the Air Force Multi-Attribute Task Battery (AF-MATB, Miller, Schmidt, Estepp, Bowers, and Davis 2014). We first implemented a mechanism for restricting perceptual span in an ACT-R model by constraining which stimuli can cause bottom-up attentional capture. Then, we developed a method for measuring subjective workload continuously in the ACT-R model. We ran simulations of the model with different sized perceptual span windows and compared the performance of those models to behavioral data to determine how much the perceptual span window should change as a function of subjective workload. We predicted that the best fitting change to perceptual span would be a narrowing of the perceptual span window when task demands increased. Instead, we found that the best fitting model had a perceptual span window that increased with task demands, suggesting participants were using less strategic visual search behaviors as the task became more difficult.

Method

Participants

Model simulations utilized event lists consisting of malfunction information in the AF-MATB from Bowers et al. (2014). Sixteen participants (11 male, 5 female, ages 18 to 28) from neighboring universities (Air Force Institute of Technology, Wright State University, University of Dayton, and Wright State Junior Force Council) participated in that study. Participants were unfamiliar with the task and completed informed consent prior to participation. The study was approved by Air Force Research Laboratory Institutional Review Board.

AF-MATB Task Description

The AF-MATB is a laboratory environment that replicates multitasking behaviors similar to those encountered by aircraft

pilots. Full details regarding the AF-MATB can be found in Miller et al. (2014). Participants monitored and responded to scripted events that occurred concurrently and were distributed pseudorandomly throughout the trial. Difficulty in Bowers et al. (2014) was determined by increasing the frequency of events resulting in greater overlap between events for the Hard difficulty compared to the Easy difficulty and by increasing the variability and movement speed of the Tracking subtask.

In Bowers et al. (2014), the AF-MATB included all of the subtasks (System Monitoring, Tracking, Communications, and Resource Management). In the System Monitoring subtasks, participants pressed a key when a Light (color change) or Gauge (exceeding a y-axis threshold) malfunctioned within a limited time (3 and 6 seconds, respectively). In the Tracking subtask, participants used a joystick to center the position of a randomly moving reticle. In the Communications subtask, participants listened for audio commands and adjusted and submitted the frequency and channel if the audio matched the participant's call sign. In the Resource Management subtask, participants monitored fluid levels in two tanks and adjusted the state of 8 pumps to maintain fluid levels within a threshold.

ACT-R Model

The ACT-R cognitive architecture consists of discrete modules for distinct types of perceptual and cognitive processing (e.g., visual, auditory, declarative memory). Cognition manifests as information moves between the different modules via production rules (if-then statements) that control the behavior of the model.

Our model was designed to detect and respond to events in the AF-MATB task environment. The model interacted with a custom built version of the AF-MATB in Python, which had reduced visual fidelity but the same event timing and spatial layout as the AF-MATB that participants experienced. The Scheduling and Pump Status panels typically present in the AF-MATB were omitted. We designed the simplest model that was similar to human behavior, given that a more complex model designed specifically to fit the data would theoretically be less generalizable. The structure described below is the core version of the model used in all of the simulations. We first describe how the model operated in the AF-MATB environment, and then how the model deployed visual attention to detect different stimuli. Next, we describe how we constrained attention with a perceptual span, estimated subjective workload in the ACT-R model, and then how subjective workload affected the size of the model's perceptual span.

Core Model

The core model has been previously described (Swan, Stevens, Fisher, & Klosterman, 2022). In short, the model serially attended each stimulus and utilized ACT-R productions to determine how to respond to the stimuli. If a keyboard response was necessary, the model waited until a response concluded before moving attention. For the Lights and Gauges subtasks in the System Monitoring panel, the model responded with a key press. For the Resource Management subtask, the model

modulated the level of Tank A and B by turning on or off Pump 2 and 4 while the model kept Pumps 1, 3, 5, and 6 always on. For the Tracking subtask, the model moved the joystick maximally in the direction of the reticle, tracked the stimulus as it moved towards the center position, and then once the stimulus had reached a sufficient distance (37.5 pixels), the model moved the joystick back to a neutral position. The model completed events in the Communication subtask separately. The model shifted attention to a channel in the Communication subtask panel after the channel and frequency information had been aurally received. The model first selected the appropriate channel, adjusted the frequency, and then pressed the enter key to submit the request.

Visual Attention The model's attention was brought to stimuli in two ways: top-down or bottom-up visual attention. When there was not currently a stimulus in the visual-location buffer, top-down control utilized a find production to search the display clockwise for a stimulus that had not been recently attended. Once the model had a visual location, then visual attention (i.e., focus of attention) was brought to that stimulus. This process was therefore systematic and exhaustive, given that it would cycle through all of the stimuli in an ordered fashion.

On the other hand, unrequested stimuli could be in the visual-location buffer through "buffer stuffing", which is an ACT-R property whereby suddenly appearing stimuli are automatically placed in the visual-location buffer. The model's visual attention could then move to that stimulus without the find production. This second method reflected task-driven bottom-attention, which was how salient information (e.g., a stimulus malfunctioning via a color change) in peripheral vision could grab attention. Unlike the model described in Swan et al. (2022), all subtasks could capture bottom-up attention via buffer stuffing. Buffer stuffing occurred when: any Light malfunctioned, any Gauge exceeded 65 pixels from center (i.e., 15 pixels beyond the vertical threshold), the Tracking reticle exceeded 62.5 pixels from center, Tank A or Tank B exceeded 200 units from center, and Pumps 1, 3, 5, or 6 were off.

Perceptual Span We developed a novel approach to incorporating perceptual span in an ACT-R model. Each time the display was refreshed (100ms), we used the location of the model's focus of attention to determine whether a given stimulus could trigger buffer stuffing. In other words, stimuli within this radius that malfunctioned or exceeded a threshold for triggering buffer stuffing could cause buffer stuffing, whereas stimuli outside of the threshold could not cause buffer stuffing regardless of their status. In this initial implementation, we varied the size of the perceptual span radius to see which size best fit behavioral data (Figure 1). Later, we use subjective workload to alter the radius continuously.

Subjective Workload Previous research has established metrics for measuring subjective workload in ACT-R, such as using weighing ACT-R module activity over time (Jo, Myung,

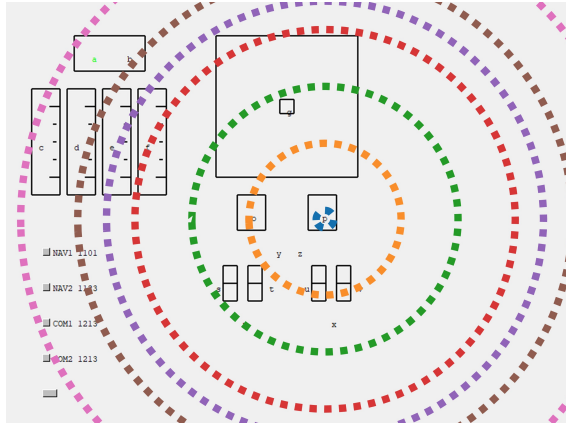


Figure 1: The AF-MATB environment for the ACT-R model. The circles represent the size of the perceptual span window for different fixed thresholds.

& Yoon, 2012; Stevens, Morris, Fisher, & Myers, 2022). Here, we were interested specifically in the relationship between visual attention to events and subjective workload. Thus, we developed a method to capture continuous moment-by-moment changes in subjective workload based on event frequency. Each time an event occurred within the perceptual span window that could capture attention (i.e., the same criteria that caused buffer stuffing described in subsection *Visual Attention*), workload increased by 2 units. Workload decreased by 0.03% each time the display refreshed (100ms). Thus, workload increased as more events occurred closer in temporal proximity, but decreased if events occurred further apart in time (Figure 2). These values were selected because they produced a range of subjective workload values across the different perceptual span window sizes.

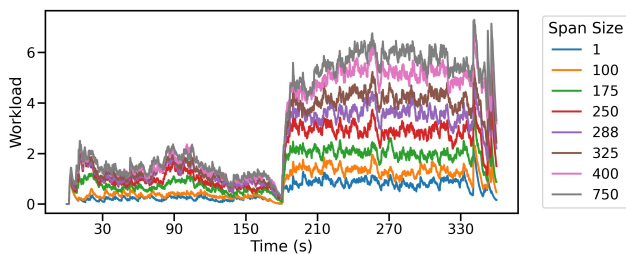


Figure 2: Averaged subjective workload when perceptual span was fixed (colors represent pixel radii) across Easy to Hard trial simulations.

Model Parameters The majority of the ACT-R model parameters were kept at their default level. We enabled subsymbolic (:esc = t) and full base level learning computations (:ol = nil). Malfunctions maintained their visual state until corrected or the malfunction timed-out after a period of time and returned to a normal function state. We therefore set the visual-onset-span parameter to 3.0 seconds, which was the minimum

time-out time and represented the ability to detect the malfunction after it had occurred in the model's peripheral vision. We set base-level learning (bll) to the recommended level (0.5). Only the Communications subtasks utilized declarative memory retrieval (i.e., storing and retrieving the frequency and channel information), which involved setting the base-level-constant (blc) to 2 and retrieval threshold to 2.9.

Performance Measures

Trial Simulation We simulated the same event lists generated from the participants in Bowers et al. (2014). Each trial was 6 minutes and had a transition from Easy to Hard (6 trials) or Hard to Easy (6 trials) half way through the trial. The order of the transition was counterbalanced by transition block. Two of the participants did not fully complete a trial, so there were a total of 190 trials per simulation. For validating our subjective workload metric, we utilized the full 190 trials. For evaluating the effect of perceptual span on performance and eye movements, we focused on Easy to Hard transition trial (i.e., 96 trials).

For the simulations, we varied the size perceptual span window radius and ran the full set of events lists (190 or 96 trials) for each radius. The simulations included a condition where only top-down attention was used (1 pixel), a condition where bottom-up attention could be used for any eccentricity (750 pixels), and gradations (100, 175, 250, 275, 325, and 400 pixels).

Dependent Measures Performance of the model was determined by accuracy (accuracy = correct / total) and reaction time (RT) for correct responses in the System Monitoring and Communications subtasks. For the Tracking subtask, we averaged the Euclidean distance of the reticle with respect to the center. For the Resource Management subtask, we averaged the deviations of both tank fluid levels with respect the target tank fluid level.

We also measured eye movements of the model to determine if the perceptual size implementation was successfully impacting how the model deployed visual attention. We recorded the position of the ACT-R model's gaze using the model's focus of attention¹. We calculated two measures of directness, or scanning efficiency (Moacdieh, Devlin, Jundi, & Riggs, 2020), to capture how the model's search changed as a function of perceptual span window size: (1) saccade magnitude and (2) transition entropy (Krejtz, Szmidt, Duchowski, & Krejtz, 2014). Saccade magnitude was measured as the line distance between the starting and ending position of an eye movement. We calculated transition entropy by first defining the possible stimuli as areas of interest (AOI). For each AOI, we calculated the transitional probability to the other AOIs, then summed each probability multiplied by log2 of that probability. That sum was then weighted by the proportion of saccades originat-

¹While there are methods for approximating eye movements in ACT-R (EMMA; Salvucci 2001), including EMMA resulted in implausibly slower response times. However, integrating modules like EMMA are a worthwhile direction for future research.

ing from that AOI. Larger saccade magnitude and transition entropy correspond to less efficient and more random and complex scanning, respectively.

Results

Validating Workload

Participants in Bowers et al. (2014) completed the NASA-TLX (Hart & Staveland, 1988), which is a subjective workload metric, at the end of each trial based on what they experienced during the second half of the trial. Given that our subjective workload metric utilized event information, we could compute continuous subjective workload for each participant in Bowers et al. (2014). We compared the NASA-TLX scores to our subjective workload metric² to determine how our metric compared to their reported experiment of workload. We found that our subjective workload metric was highly correlated with the overall NASA-TLX score (Spearman's $\rho = 0.77$, $p < 0.001$) (Figure 3 Top). It is also clear from the figure that our workload metric differentiated between task demands. Interestingly, within condition correlations were significant for the Easy ($\rho = 0.26$, $p = 0.01$) but not Hard ($\rho = 0.07$, $p = 0.5$) task demands.

Next, we measured subjective workload in the ACT-R model without perceptual span and compared those values to the NASA-TLX scores to see if there was a similar relationship between the model's experience of workload and the participants reported workload (Figure 3 Bottom). Subjective workload from the model should be similar to the workload from the event lists, given that the model used those same event lists, but differences would arise from randomness in the Tracking subtask and different strategies for the Resource Management subtask. Similar to the previous correlations, there was a strong relationship when including both workload conditions ($\rho = 0.71$, $p < 0.001$) indicating agreement that the Harder difficulty had higher subjective workload. As can also be seen, the model strongly differentiates workload levels between Easy and Hard. Neither Easy ($\rho = 0.003$, $p = 0.98$) nor Hard ($\rho = -0.02$, $p = 0.88$) were significantly correlated.

Validating Perceptual Span

To determine how well the implementation of perceptual span worked, we looked at the proportion of stimuli that could cause buffer stuffing as a function of the size of the perceptual span window. As intended, smaller perceptual span windows had fewer stimuli that could capture attention via bottom-up attention (1 pixel: 0.04, 100: 0.15, 175: 0.34, 250: 0.50, 288: 0.64, 325: 0.76, 400: 0.91, 750: 1.0).

We next measured the effect perceptual span had on the eye movements of the model given that increasing the size of the perceptual span window should allow the model to make larger and less predictable eye movements.

²We averaged the last 90s of the subjective workload metric for each trial to approximately emulate the information participants were using when submitting their NASA-TLX scores. Note that this approach was used for subsequent analyses using the subjective workload metric.

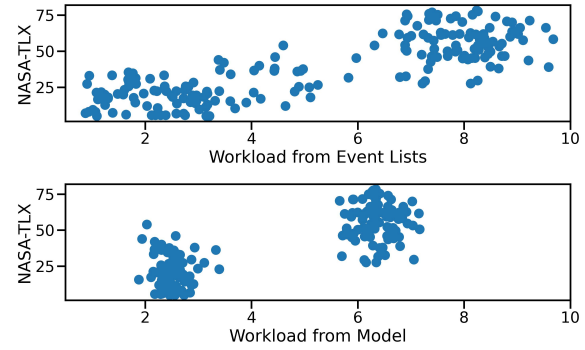


Figure 3: (Top) Subjective workload calculated from participant event lists compared to NASA-TLX reports from the participant data Bowers et al. (2014). (Bottom) Subjective workload from the model compared to NASA-TLX reports from the participant data.

We first looked at the transition entropy. As expected, there was overall more transition entropy for the Hard difficulty (mean = 2.1, standard deviation = 0.4) than the Easy difficulty (mean = 1.3, standard deviation = 0.1). Transition entropy also increased as the size of the perceptual span window increased and increased more rapidly in the Hard than Easy condition, as can be seen (Figure 4). This was expected given that there were more events to grab bottom-up attention in the Hard condition relative to the Easy condition.

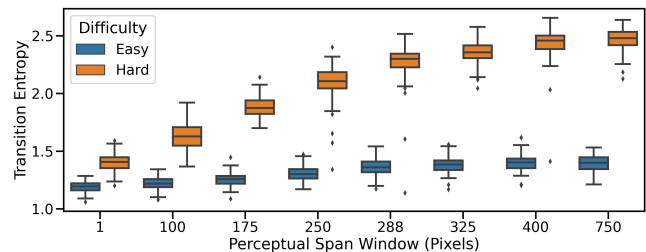


Figure 4: Transition entropy as a function of task difficulty and the size of the perceptual span window.

We next looked at the saccade amplitude. Saccades were larger in the Hard (mean = 162.5 pixels, standard deviation = 17.8) than Easy condition (mean = 141.6, standard deviation = 3.7). Saccade amplitude also increased more rapidly in the Hard than Easy condition, as can be seen by comparing the average Easy and Hard task difficulty saccade amplitudes (e.g., 1 pixel: 136.6 Easy vs 143.0 Hard, 250: 142.2 vs 155.9, 750: 145.0 vs 189.4), mimicking the results of transition entropy.

Changes in Performance

We examined how performance in the AF-MATB ACT-R model changed as a function of the different perceptual span window sizes. Generally, model performance improved when transition entropy increased, as can be summarized by looking at performance for perceptual window sizes 1, 250, and

750 in table 1. The exceptions were in [1] the Communication subtask, which we did not expect to be affected by our implementations given that the Communication subtask was completed separately without buffer stuffing, [2] Gauge subtask accuracy, which was relatively stable given the long period of time to respond before the malfunction returned to a functioning state (6 seconds), and [3] Resource Management subtask, which was at floor for Easy task demands and the Hard task demands required too much intervention to prevent the tank level from deviating too far from the center.

Table 1: Dependent variables (DV) for the different performance metrics across different perceptual span window sizes (1, 250, and 750) for Easy (E.) and Hard (H.) conditions.

DV	1 (m, sd)	250 (m, sd)	750 (m, sd)
Light Rt. E.	1.67(0.2)	1.21(0.2)	1.0(.1)
Light Rt. H.	1.64(0.2)	1.4(0.2)	1.4(.1)
Light Acc. E.	0.83(0.1)	0.94(0.1)	0.97(.1)
Light Acc. H.	0.4(0.1)	0.54(0.1)	0.62(.1)
Gauge Rt. E.	2.3(0.3)	2.1(0.2)	1.8(.2)
Gauge Rt. H.	3.1(0.2)	2.8(0.2)	2.7(.2)
Gauge Acc. E.	0.98(0.0)	0.98(0.0)	0.98(.0)
Gauge Acc. H.	0.67(0.1)	0.67(0.1)	0.70(.1)
Comm. Rt. E.	8.9(0.5)	8.9(0.4)	8.9(.4)
Comm. Rt. H.	9.4(0.3)	9.4(0.3)	9.4(.3)
Comm. Acc. E.	0.96(0.1)	0.96(0.1)	0.97(.1)
Comm. Acc. H.	0.95(0.1)	0.93(0.1)	0.94(.1)
Res. Man. E.	66(7)	65(10)	66(9)
Res. Man. H.	260(123)	336(208)	333(147)
Tracking E.	43(3)	41(3)	41(3)
Tracking H.	134(13)	133(30)	129 (20)

Comparing Model to Bowers et al. (2014)

We next compared model performance to Bowers et al. (2014). We were interested in the perceptual span window size closest to the behavioral data separately for the different task demand conditions (Easy and Hard) to determine if participant's visual attention changed as a function of task demand. For example, if the same perceptual window size best fit both the Easy and Hard task conditions, then that would suggest that perceptual span was not affected by task demands. Alternatively, finding that Easy and Hard were best fit by different windows suggests that task demand did change the size of the perceptual span window. We used normalized root mean square error (NRMSE) to compare the model's performance to the behavioral data. We normalized using the average participant performance, given that the subtasks have different scales, and averaged across dependent variables to determine the overall closest performance of the model to the behavioral data.

In the Easy condition, there was a u-shaped pattern whereby NRMSE decreased from 0.268 (1 pixel) to 0.223 (175) to the minimum NRMSE 0.211 (288) until rising to 0.221 (750). The model overperformed relative to the participant's data when

the perceptual span window was broad, hence worse fit to the data.

In the Hard condition, there was increased variability as a function of the Tracking and Resource Management subtask. The worst performing spans were 0.309 (1), 0.298 (100), and 0.286 (325) and the best were 0.262 (288), 0.268 (400), and 0.246 (750), which generally suggests participants utilized a broad perceptual span window.

We used the best perceptual span window sizes for Easy (288) and Hard (750) conditions and our subjective workload metric to come up with a method whereby the model would increase the perceptual span window as task demands increased. We used the following equation such that the Easy and Hard conditions approximated the same percentage of capture as the 288 and 750 sizes, respectfully: $\text{span} = 190.1 * \text{Workload} + 184.6$. When comparing across conditions (i.e., the NRMSE for the entire trial), this version of the model where subjective workload affected the size of the perceptual model mostly outperformed the versions of the model where perceptual span was constant (model with varying span: 0.19 vs. 0.2 for static spans of 250, 288, 325, and 750).

Discussion

We implemented a visual attention mechanism in an ACT-R model that could vary in real-time the perceptual span of the model as a function of subjective workload. We believe implementations like these provide theoretical constraints that improve simulations of human behavior and provide new avenues for interesting predictions.

Our workload metric differentiated Easy and Hard levels of workload. This suggests that this metric may be useful in capturing subjective workload continuously. Interestingly, the metric was not strongly associated within condition, which was surprising given that our metric was based on the number of concurrent events. There are two likely interpretations of this outcome. One is that the within condition rankings were driven by individual variability not captured by the model. For example, other work has revealed that subjective reports of workload can be subject to bias under certain circumstances, such as individuals having different perceptions of workload (Hart & Staveland, 1988). Another is that the criteria for incrementing the workload metric did not capture some aspect of the event sequences that drove the workload ratings. Both of these possibilities can be investigated with further modeling and exploration.

Similar to PAAV (Nyamsuren & Taatgen, 2013), this work provided a mechanism for incorporating eccentricity effects into the visual processing of ACT-R models. Our implementation of perceptual span successfully constrained visual attention, such that stimuli within the window could grab attention via bottom-up attention and stimuli outside the window could only be attended through other mechanisms, such as a find production through top-down attention. Larger perceptual span windows thus resulted in larger saccade amplitudes and less predictable eye movements and better performance because

the model was able to directly attend and respond to distal malfunctions.

We predicted that the perceptual span window would decrease in size as task demands increased. Instead, we found the opposite in this dataset. The best fitting models involved perceptual span sizes that increased from Easy to Hard difficulty. A couple of factors likely drove finding. First, when the model was able to use bottom-up capture to detect malfunctions, the model outperforms participants when the task demands were Easy. Fewer events in the Easy condition also meant the model could detect and respond to malfunctions through top-down attention. Conversely, there were significantly more events when the task demands were Hard and the model was delayed when top-down attention brought attention to non-malfunctioned stimuli. It is possible that participants did utilize more systematic search strategies, given that individual do search differently (Boot, Becic, & Kramer, 2009), but that these strategies were washed out in the aggregate. Without eye tracking data to examine scan paths and/or more behavioral data, it is difficult to determine if this was the case.

We designed and implemented subjective workload and visual attention mechanisms in an ACT-R model to more accurately capture behavior in complex multitasking environments. Our implementations produced expected changes in eye movements and performance. This work thus provides a mechanism whereby visual attention could shift from broad to focal, or vice versa, as a function of subjective workload.

Acknowledgments

The opinions expressed herein are solely those of the authors and do not necessarily represent the opinions of the United States Government, the U.S. Department of Defense, the U.S. Air Force, or any of their subsidiaries, or employees. Distribution A: Approved for Public Release. Case Number: AFRL-2023-2124.

References

- Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological review*, 111(4), 1036.
- Bertera, J. H., & Rayner, K. (2000). Eye movements and the span of the effective stimulus in visual search. *Perception & psychophysics*, 62(3), 576–585.
- Boot, W. R., Becic, E., & Kramer, A. F. (2009). Stable individual differences in search strategy?: The effect of task demands and motivational factors on scanning strategy in visual search. *Journal of Vision*, 9(3), 7–7.
- Bowers, M. A., Christensen, J. C., & Eggemeier, F. T. (2014). The effects of workload transitions in a multitasking environment. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 58, pp. 220–224).
- Carrasco, M. (2011). Visual attention: The past 25 years. *Vision research*, 51(13), 1484–1525.
- Carrasco, M., Evert, D. L., Chang, I., & Katz, S. M. (1995). The eccentricity effect: Target eccentricity affects performance on conjunction searches. *Perception & psychophysics*, 57, 1241–1261.
- Carrasco, M., & Yeshurun, Y. (1998). The contribution of covert attention to the set-size and eccentricity effects in visual search. *Journal of Experimental Psychology: Human Perception and Performance*, 24(2), 673.
- Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (task load index): Results of empirical and theoretical research. In *Advances in psychology* (Vol. 52, pp. 139–183). Elsevier.
- Jo, S., Myung, R., & Yoon, D. (2012). Quantitative prediction of mental workload with the act-r cognitive architecture. *International Journal of Industrial Ergonomics*, 42(4), 359–370.
- Krejtz, K., Szmidt, T., Duchowski, A. T., & Krejtz, I. (2014). Entropy-based statistical analysis of eye movement transitions. In *Proceedings of the symposium on eye tracking research and applications* (pp. 159–166).
- Miller, W. D., Schmidt, K. D., Estep, J. R., Bowers, M., & Davis, I. (2014). *An updated version of the us air force multi-attribute task battery (af-matb)*. (Tech. Rep. No. AFRL-RH-WP-SR-2014-0001). Retrieved from the Defense Technical Information Center website: <https://apps.dtic.mil/sti/pdfs/ADA611870.pdf>.
- Moacdieh, N. M., Devlin, S. P., Jundi, H., & Riggs, S. L. (2020). Effects of workload and workload transitions on attention allocation in a dual-task environment: Evidence from eye tracking metrics. *Journal of Cognitive Engineering and Decision Making*, 14(2), 132–151.
- Nyamsuren, E., & Taatgen, N. A. (2013). Pre-attentive and attentive vision module. *Cognitive systems research*, 24, 62–71.
- Pomplun, M., Garaas, T. W., & Carrasco, M. (2013). The effects of task difficulty on visual search strategy in virtual 3d displays. *Journal of vision*, 13(3), 24–24.
- Salvucci, D. D. (2001). An integrated model of eye movements and visual encoding. *Cognitive Systems Research*, 1(4), 201–220.
- Stevens, C. A., Morris, M. B., Fisher, C. R., & Myers, C. W. (2022). Profiling cognitive workload in an unmanned vehicle control task with cognitive models and physiological metrics. *Military Psychology*, 1–14.
- Swan, G., Stevens, C. A., Fisher, C. R., & Klosterman, S. (2022). Exploring multitasking strategies in an act-r model of a complex piloting task. In *Virtual mathpsych/iccm 2022*. mathpsych.org/presentation/719.
- Thomas, L. C., & Wickens, C. D. (2001). Visual displays and cognitive tunneling: Frames of reference effects on spatial judgments and change detection. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 45, pp. 336–340).
- Young, A. H., & Hulleman, J. (2013). Eye movements reveal how task difficulty moulds visual search. *Journal of Experimental Psychology: Human Perception and Performance*, 39(1), 168.

Preferred Mental Models in Syllogistic Reasoning

Sara Todorovikj (sara.todorovikj@hsw.tu-chemnitz.de)

Daniel Brand (daniel.brand@hsw.tu-chemnitz.de)

Marco Ragni (marco.ragni@hsw.tu-chemnitz.de)

Department Behavioural and Social Sciences, Technische Universität Chemnitz,
Straße der Nationen 62, 09111 Chemnitz

Abstract

Inspired from previous research in the spatial reasoning domain, in this paper, we address the varying interpretations of premises of syllogistic problems among individuals and the differences in their resulting mental models. We conducted an experiment whose results show that model building is a relatively easy task for humans to do correctly and they do in fact have preferred models for most syllogisms, yet, without a relation to their responses. We report in-depth analysis of the models' canonicity in order to compare the model building behavior in humans to the processes implemented in mReasoner, a cognitive model that implements the Mental Model Theory.

Keywords: Syllogistic Reasoning; Preferred Mental Models; mReasoner

Introduction

With over a century of research history (Störring, 1908), syllogisms are one of the core domains of examining human reasoning abilities. A syllogism consists of two quantified premises describing the relationships between three terms through a common middle term. In a world of colourful shapes, consider the following syllogism:

All red shapes are circles.
Some red shapes are marked with a star.

What, if anything, follows?

The task at hand is to determine what kind of relation, if any, exists between the two end-terms, circles and (marked with a) star, also called subject and predicate, respectively.

There exist at least twelve theories that aim to explain and model the processes behind human syllogistic reasoning (for an overview, see Khemlani & Johnson-Laird, 2012). One of the most prominent theories among them is the Mental Model Theory (MMT; e.g., Johnson-Laird, 1975, 2010). MMT postulates that given some observations, individuals create iconic representations – *mental models* – of possibilities. They create their own subjective mental representation of the information presented in a reasoning task. Considering the example above, one possible representation would be:

circles [red] [star]
circles

The square brackets around an instance denote that the set of entities described by it is exhaustively represented. Another possible mental model representation is:

circles
circles red [star]
¬circles ¬red

where \neg denotes negation. Both mental representations support the conclusion “Some circles are marked with a star” – the logically valid conclusion to this syllogism. However, in order to confirm the validity, an individual should think of all possible premise interpretations and check if they hold. The expansion of the interpretation search space can make solving such problems difficult for humans (Johnson-Laird, 2006).

In the spatial relational reasoning domain researchers have repeatedly shown that individuals have *preferred mental models*, namely that they prefer creating some models while struggling with others (e.g., Ragni & Knauff, 2013; Jahn, Knauff, & Johnson-Laird, 2007; Rauh et al., 2005). Interestingly, experimental setups for the syllogistic domain do not typically address the model building process of reasoning. Namely, they do not involve examinations of which models the individuals create, if they are correct, or if they even have preferred models at all. To this end, we conducted an experiment where participants had to provide visual responses showing their representation of the given syllogistic premises, and with that we tackle our first research question:

[RQ1] Can we examine what kind of models do individuals create from the premises of syllogistic tasks and do they have preferred mental models?

In mathematical and computer sciences, the minimal, simplest representation of an expression is referred to as its canonical form. This concept is also discussed in the context of mental models in the syllogistic reasoning domain (Khemlani, Lotstein, Trafton, & Johnson-Laird, 2015). Namely, it denotes which entities form a canonical set for a given syllogism and which non-canonical instances do not have to be present in an individual's model but are still consistent with the premises. For example, in the representations above “circles red” is a canonical instance for the first syllogistic premise (“All red shapes are circles”), whereas “¬circles ¬red” is not. Thereby, canonicity can be interpreted as a mean to assess the “incompleteness” of the model in the sense of the coverage of all possible interpretations of the premises.

From the perspective of cognitive modeling, it is especially interesting if the canonicity of the models provided

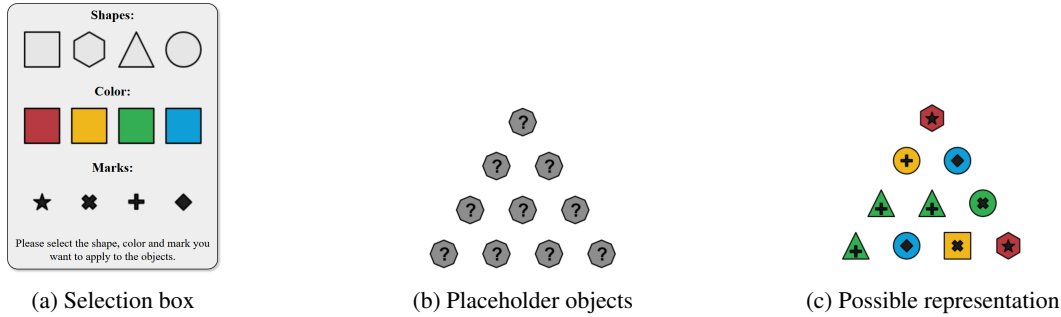


Figure 1: Experimental design – Participants used the selection box to pick out their desired attributes that they can assign to placeholder objects in order to share their mental representation of a given syllogism.

by participants align with the assumptions made by MMT. The most prominent implementation of MMT for syllogistic reasoning is the LISP-based model *mReasoner*¹ (Khemlani & Johnson-Laird, 2013), which will therefore serve as a foundation of our analysis. Distinguishing between three systems, *mReasoner* creates intensional representations of the premises (System 0), builds and interprets an initial model (System 1) and performs a search for counterexamples (System 2) (Khemlani & Johnson-Laird, 2013). System 1 parameterizes the number of entities in a model and their canonicity - the likelihood whether they are drawn from a canonical set of typical entities or the full set of entities consistent with the premises (Khemlani et al., 2015). We analyze our participants' built models further and contrast them to the output of *mReasoner*'s model building stage to address our bipartite second research question:

[RQ2.1] How influential is the canonicity of mental models that individuals build for syllogistic premises on the correctness of derived conclusions?

[RQ2.2] Is the model building behavior observed in humans in line with the model building processes of *mReasoner*?

Our paper is structured as follows – we first provide the necessary theoretical background regarding reasoning with syllogisms and *mReasoner*, followed by an in-depth explanation of our experiment. Afterwards, we analyze the experimental data and the participants' models (RQ1) and the correspondence of *mReasoner*'s canonicity approach to the data (RQ2). We then conclude the article with a discussion of our findings.

Theoretical Background

Syllogisms

The two syllogistic premises and conclusion are characterized by their quantifiers and term order. We take into consideration the four first-order logic quantifiers *All*, *Some*, *No* and *Some not*, abbreviated by A, I, E and O, respectively. The order of the subject, predicate and middle terms in the premises determine the *figure* of the syllogism. We use the following

notation (adopted from Khemlani & Johnson-Laird, 2012):

Figure 1	Figure 2	Figure 3	Figure 4
A-B	B-A	A-B	B-A
B-C	C-B	C-B	B-C

Using the abbreviations and figures, the example syllogism introduced above is denoted by AI4. Similarly, the conclusions are denoted by using the quantifier's abbreviation and take into consideration the direction of the end-terms – *ac* or *ca*, e.g. *Oca* indicates that Some C are not A. Finally, 'No valid conclusion' is abbreviated by NVC.

mReasoner

According to MMT, given syllogistic premises, individuals represent the entities described by the quantifiers using mental models and aim to derive a conclusion based on that. Before accepting a certain conclusion, they engage in a search for counterexamples, which, if successful, can lead to rejecting and correcting the original conclusion or concluding that there is no valid conclusion.

These processes are implemented within the cognitive model *mReasoner* (Khemlani & Johnson-Laird, 2013, 2016). Using the following four parameters it builds models and searches for counterexamples: λ determines the *size*, i.e. the number of entities as drawn from a Poisson distribution; ϵ determines the *canonicity*, i.e. how complete is the set of represented possibilities, given the premises; σ describes how likely is it to engage in a search for *counterexamples* and ω decides what happens when a counterexample is found – whether the conclusion is weakened or NVC is reported.

Experiment

The main objective of the experiment was to obtain a visual representation of the participants' (preferred) mental representation of given syllogisms. In order to achieve that, they were presented with a syllogism, whose terms are descriptions of objects and were asked to demonstrate what they imagined ten objects look like when taking into consideration the syllogistic premises.

An object is described using its *shape* (square, hexagon, triangle, circle), *color* (red, blue, green, yellow) and *mark*

¹<https://github.com/skhemlani/mReasoner>

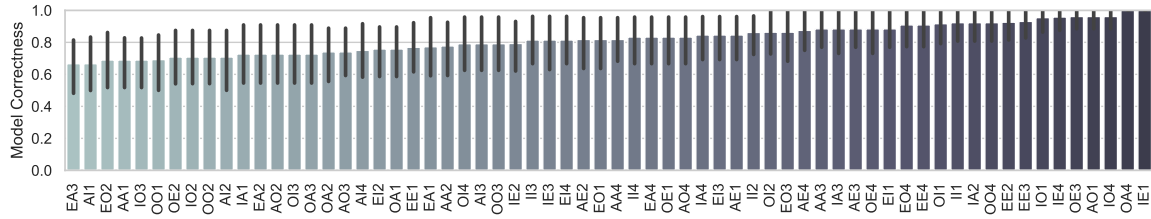


Figure 2: Mean correctness of the constructed models by syllogism.

(cross, plus, star, diamond). The presented syllogisms had the shapes as subjects, the colors as middle terms and the marks as predicates. For example, the syllogism AE3 with content ‘triangles’, ‘green’ and ‘diamond’ is as follows:

All triangles are green.

No shapes that are marked with a diamond are green.

The object attributes were randomized among syllogisms. Once presented with a syllogism, participants see a selection box and placeholder objects (Fig. 1a, 1b). By clicking on any shape, color and mark they select their desired object attributes that they can apply on a placeholder object by clicking on it. Once they are done defining the properties of each object, they end up with a visual representation of their mental model of the syllogism, i.e. what they imagine the 10 objects to look like based on information provided in the syllogism. For example, Fig. 1c depicts a possible mental representation of the syllogism AE3 presented above.

In a second part of the experiment, participants are once again presented with the same syllogisms and are prompted to select which of the 9 possible responses follow (e.g. Brand, Riesterer, & Ragni, 2022).

Participants are divided in eight groups based on the presented syllogisms. In order to maintain a similar experience among participants, we used the Ragni-2016 dataset obtained from the Cognitive Computation for Behavioral Reasoning Analysis (CCOBRA) Framework², to determine the difficulty of syllogisms based on the amount of correct responses. The Ragni-2016 experimental data provides responses from 139 participants for all 64 syllogisms. We divided all syllogisms in eight difficulty groups and created the final sets of presented syllogisms by selecting one from each group.

Participants

We obtained data from 200 participants (age 19-76, 42% female) recruited on Prolific³ and the experiment was performed online as a web-experiment. After completing the experiment, the participants received compensation of 3 GBP. All of them were native English speakers.

Procedure

Participants were first introduced to all possible attribute options in terms of shapes, colors and marks that an object can

have. Following is an explanation on how to select and apply the desired object attributes on the placeholder objects. It was emphasized that they must select an option for each attribute and the appearance of all ten objects has to be specified. Then the experiment started and they had to show what they imagine the objects look like, using the introduced selection-box and placeholder objects, for 8 syllogisms. Once these tasks were completed, participants started the second portion of the experiment – the single choice tasks for the same 8 syllogisms.

Analysis

General Experimental Data Analysis

Since the object attribute descriptions were randomized among tasks, throughout our analysis and model comparisons we focus on whether the attributes in the responses correspond to the attributes presented in premises or not. That means, for example the model instance “square red” for the syllogistic premise “All squares are red” is treated equally with the model instance “circle green” for the premise “All circles are green”.

We analyzed the correctness of the provided representations. Given a syllogistic premise with terms X and Y, we distinguish the following scenarios in which the representations are correct, based on the quantifier. For *All*, there must be no $X \rightarrow Y$ instances. In the case of *No*, there must be no XY instances. Finally, for *Some* and *Some not*, there should be at least one XY or $X \rightarrow Y$ instance, respectively. Out of 1600 observations, participants provided a correct representation in 1314 of them (82.12%). The mean correctness of the models by syllogism is visually represented in Figure 2. In 497 cases (31.06%) the participants gave a logically correct response and for only 408 (25.50%) they provided a correct representation *and* a logically correct response. Despite substantial differences in the model correctness between different syllogisms, it does not appear to be related to the difficulty of the syllogism: The correctness of the representations and responses do not have a significant correlation (Spearman’s $r = -.0005$, $p = .9819$). Besides no apparent connection to task difficulty, a comparison between the best and worst 32 tasks also indicates that the model correctness seem to not be affected by negativity of quantifiers (24/24 in best/worst, respectively), particularity (25/23) or validity (17/20 invalid syllogisms). The only peculiarity is related to the figure of the syllogism, with figure 2 leading to more incorrect models

²<https://orca.informatik.uni-freiburg.de/ccobra/>

³<https://www.prolific.co/>

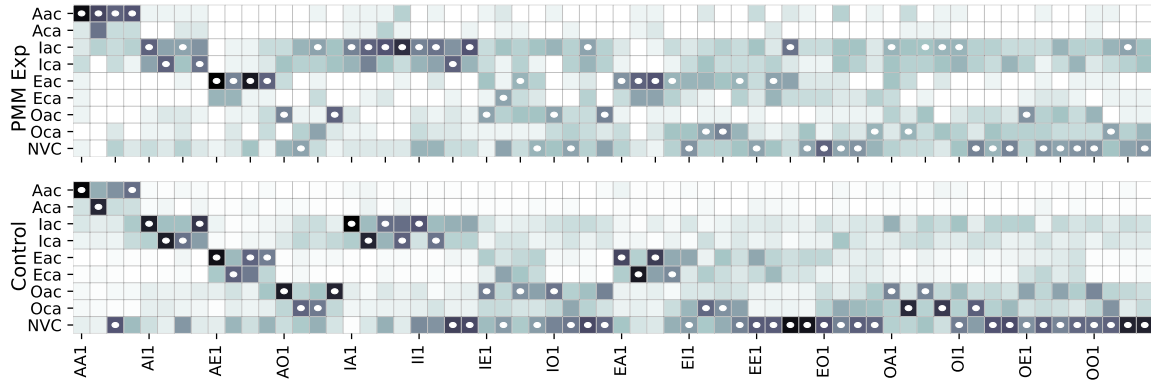


Figure 3: Response distributions for all syllogisms in the conducted experiment and the Ragni-2016 control dataset. Higher percentage of given response is depicted with a darker color and the most frequently selected response for each syllogism is denoted with a white dot.

(12/4) and figure 4 seemingly being easier for model construction (3/13). Overall, the correctness of the models remains arbitrary with respect to typical structural properties of syllogisms commonly known to affect syllogistic reasoning performance. The response distribution among all participants is illustrated in Figure 3. We contrast the obtained responses to neutral, control data – the Ragni-2016 data from CCOBRA, as introduced above. It can be immediately observed there is a tendency for participants to not choose NVC answers as often as in the control data and to avoid the *ca* direction in their responses. This implies that there was a belief bias effect among participants, namely that some superficial beliefs and background knowledge were induced by introducing a world with a discrete amount of possible object attributes.

Preferred Mental Models

Similarly to above, in the following analysis we focus solely on the presence and absence of the attributes in the participants' responses, without considering the specific contents. That narrows down the instance space to 8 different entities, for a syllogism with terms X, Y and Z:

$$\begin{array}{cccc} X & Y & Z & X & Y & \neg Z & X & \neg Y & Z & X & \neg Y & \neg Z \\ \neg X & Y & Z & \neg X & Y & \neg Z & \neg X & \neg Y & Z & \neg X & \neg Y & \neg Z \end{array}$$

For each representation, we created a binary vector of size 8 that indicates whether an instance was present in the model (= 1) or not (0) denoting an individual's preferred model pattern. We then counted among all participants, how many times each pattern occurred for each syllogism. The one pattern with the most occurrences is then the preferred mental model for a given syllogism. Figure 4 shows a visual representation of the participants' preferred mental models. Note that not all 64 syllogisms are represented, only those that have only one preferred model and more than 2 individuals have given them as responses (binomial test with likelihood 2^8).

Honorable Mentions Here, we briefly report on interesting findings among the other syllogisms that did not have a

clearly preferred mental model. Starting with AA1, which had a tie for a preferred model - 24% of participants created an "XYZ" representation, and another 24% added the entity " $\neg X \neg Y \neg Z$ " to it. In other words, one part of the population represented only the terms they were presented with, whereas the other part made a point to include terms not mentioned at all, as an offset. For EA4, 21% of the participants created an " $\neg XYZ$ " representation and other 21% added the instances " $X \neg Y \neg Z$ " and " $\neg X \neg Y \neg Z$ " to it. Namely, the second group explicitly represented that "No Y are X", but both X and Y can exist without the other one. For the rest of the syllogisms, no particularly interesting patterns were found – there are no preferred mental models for them.

mReasoner

When building a model, two of mReasoner's parameters are relevant: λ - which controls the number of instances in the model and ϵ - which determines the likelihood that the model representation is constructed with instances from the full set in contrast to only canonical ones (Khemlani et al., 2015). In Table 1 we show the canonical and noncanonical instances that can be drawn from the sets, according to the LISP implementation of mReasoner.

First, we looked into the instances of the participants' representations in terms of canonicity. Based on the amount of noncanonical models, we derived which ϵ value would be used according to mReasoner's postulates. For example, for the premise "All circles are red", if we have 8 instances of "circles red" and 2 instances of "diamonds red", following Table 1, we have 8 canonical and 2 noncanonical entities, out of 10. That means that the assigned⁴ ϵ value would be 0.2. The distribution of obtained ϵ values is shown in the left-most barplot in Figure 5. We did not find any correlation between the assigned ϵ values and the correctness of responses (Spearman's $r = -.0343, p = .1702$).

⁴Please note that mReasoner's ϵ value is a *likelihood* - what we assign is a value based on the proportion of noncanonical instances we observe in one specific individual outcome, not in terms of probabilities.

Afterwards, we fit mReasoner to all task response pairs using a grid-search to determine the parameters. For ϵ , we used values in the range of 0 to 0.9 with steps of 0.1. While the maximum value for ϵ is 1.0, it was omitted since mReasoner frequently fails the model creation phase. The parameters associated with the search for counterexamples (σ and ω) had a less fine-grained stepsize of 0.25. With respect to λ , which controls the size of the constructed model, we used two different approaches:

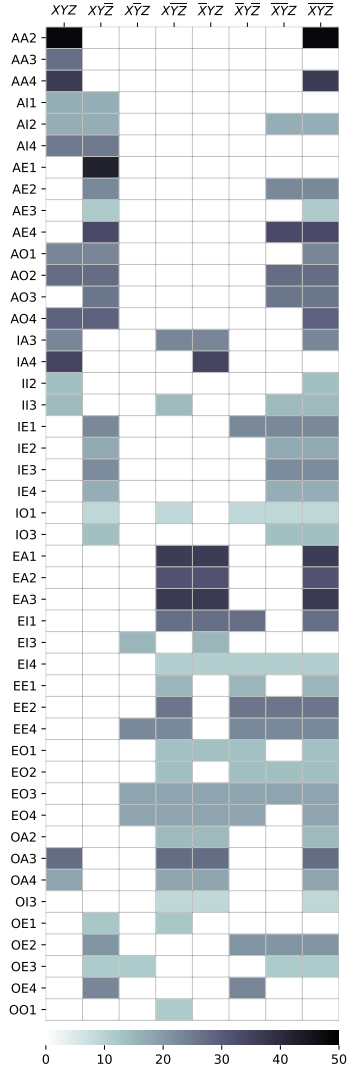


Figure 4: Preferred models provided by the participants for different syllogisms. Only syllogisms with a unique preferred model are shown. The columns denote possible instances present in the provided model. Shading illustrates the proportion of participants creating a model containing the respective instances.

First, we bypassed the λ -parameter and instead “forced” mReasoner to create exactly 10 instances to reflect the experimental setup. Furthermore, we ensured that the constructed instances precisely reflect the value of ϵ -parameter (i.e., in-

stead of using ϵ as the probability to draw from the full set including non-canonical interpretations, it now defines the proportion of non-canonical instances). This provides an opportunity for a direct comparison between the participants’ representations and mReasoner’s created models, especially in terms of parameter values. The distribution of the proportion of non-canonical instances in the models created by participants as well as the distributions of ϵ is shown in the first and second barplot of Figure 5, respectively. Visually, the distributions seem to differ substantially with little common trends observable. This is supported statistically, since no significant correlation between the distributions was found (Spearman’s $r = .0058, p = .0694$).

Quantifier	Canonical	Noncanonical
All	X Y	$\neg X$ Y
		$\neg X \neg Y$
Some	X Y	$\neg X$ Y
	X $\neg Y$	$\neg X \neg Y$
No	$\neg X$ Y	$\neg X \neg Y$
	X $\neg Y$	
Some not	X Y	
	X $\neg Y$	$\neg X \neg Y$
	$\neg X$ Y	

Table 1: Canonical and noncanonical instances for a syllogistic premise with terms X and Y according to mReasoner (Khemlani et al., 2015)

Second, we fitted mReasoner with an active λ parameter fitting, obtaining results with the intended configuration and thereby eliminates potentially introduced problems due to our manipulation. Additionally, participants might not use all 10 instances to reason about the conclusion, even if the scenario suggests it. Here we report the distribution of the best-fitting ϵ values (for any λ) in the third plot of Figure 5, followed by the distribution with “estimated” ϵ values based on the actual proportion of noncanonical instances (ϵ_{est}). Thereby, ϵ_{est} resembles the same interpretation as the ϵ -values in the previous scenario with a fixed size. Note that in some cases, ϵ_{est} can still have the value 1.0, since it reflects the actually created model and not the likelihood. We did not find correlation neither between our assigned ϵ values and mReasoner’s ϵ (Spearman’s $r = .0330, p = .2980$) nor with ϵ_{est} (Spearman’s $r = .0576, p = .0720$). We ensure that forcing mReasoner to work with exactly 10 instances does not influence the ϵ distributions (Spearman’s $r = .7564, p < .0001$). It is important to note however that there are multiple potential ϵ values that could be used for fitting to a task response pair. In the case of a fixed model size, there were on average 6.1 values leading to the same response, while there are 6.6. for the regular approach (out of 10 possible values for ϵ in the grid-search).

Discussion

In this paper we investigate two research questions regarding the mental model building process in syllogistic reasoning. For **RQ1** we examined what kind of mental models individuals create when presented with syllogistic premises. Towards that we designed and conducted an experiment centered around an imaginary world of colourful shapes with marks, where participants had to provide their visual representation of syllogisms first, and afterwards gave their conclusions. We noted a tendency for a belief bias effect in their conclusions. Namely, this suggests that an individual might be hesitant to conclude NVC, when some background knowledge regarding the existence other shapes might go against it. This is of interest for potential investigations of belief bias effect in a controlled content environment. Regarding the mental models, 82% of them were correctly representing what is stated in the syllogistic premises, indicating a general ability to correctly interpret them, and no particular syllogistic property was found to affect the correctness. We found preferred mental models for 46 out of 64 syllogisms, some occurring within a larger proportion of participants than others. There is a noticeable tendency among syllogisms with an A-premise to include noncanonical instances with terms that were not presented at all, likely due to them being an easy addition without introducing errors. We note a weakness in the PMMs for syllogisms with particular quantifiers (I, O) – though a preferred model was found, it was a smaller proportion of participants, i.e. their interpretation is rather varying. This could be associated with the quantifier’s low informativeness allowing for more possible models without a clear preference. This in turn might be a reason for a lower confidence in an individual’s interpretation, which is a proposition by another prominent syllogistic reasoning model - the Probability Heuristics Model (PHM; Chater & Oaksford, 1999; Oaksford & Chater, 2001).

For **RQ2** we looked into the canonicity of the individuals’ mental models, whether that ties into their responses and ultimately whether the observed behavior is in line with the model building process of mReasoner. In order to quantitatively analyze the canonicity of the models, we leaned on mReasoner’s canonicity parameter, ϵ . We contrast correlation analyses between response correctness and a) ϵ values assigned based on observed noncanonicity proportions; b) fitted mReasoner ϵ values on task responses, with “forced” 10 instances and with the regular intended configuration. We did not find any significant correlation in any scenario, pointing to a potential lack of relevance of the models for the responses. On the other hand, another reason might be that we cannot confirm with confidence that the built models in the experiment were indeed used for the reasoning portion of it. In MMT, the model building process is rather important, however in the mReasoner implementation (and our grid-search when fitting), we have more than 6 values out of 10 that can be used on average. This leads to the question if having two parameters for the model building process is really neces-

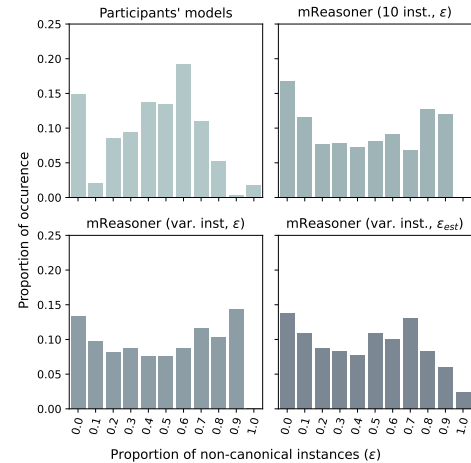


Figure 5: Distribution of ϵ (proportion of noncanonical instances) of the instances directly provided by participants and fits of mReasoner to their responses. For mReasoner, distinctions between a fit with 10 instances and a variable number of instances are made. In the case of a variable number of instances, the ϵ -parameters used by mReasoner and the estimated ϵ based on the resulting models is shown.

sary, from a complexity perspective. However, one of mReasoner’s assumptions is that humans build correct representations, which is mostly in line with our observations. In case of errors, a potential source can be incomplete representation, which is also in line with our observations. As an example, we take the syllogism AA4 and its preferred mental model that consists solely of “XYZ” and “ $\neg X \neg Y \neg Z$ ” meaning that no instance supports the logically correct conclusions, *Iac* and *Ica*. In order for an individual to conclude NVC, only one single model is not sufficient in order to deduce that there is a contradiction. There are two possibilities, we either use some additional processes (e.g. heuristics, search for counterexamples) or we create and test multiple models. By definition, mReasoner does not build two models for NVC. In the initial phase of model building, it assumes a correct construction and then uses epsilon to draw the exact instances. Later on, the initial representation is manipulated to e.g. add counterexamples and enable the conclusion of other responses.

To summarize, we can conclude that while individuals do have preferred mental models for a large portion of syllogisms, the initially built mental models are not substantial for finding conclusions. It is very likely that this is due to syllogisms being generally imbalanced in terms of validity, meaning that the majority of them can not be solved straightforwardly with an initial model anyway. It is, however, important to know that even when a final representation might be incomplete, its instances are still appropriately chosen in line with the premises. In contrast to manipulation of an existing model, as proposed and implemented by mReasoner, the model building phase seems to be a rather easy task for humans, so certainly, a plausible way to solve the tasks would in fact be a repeated construction of models.

References

- Brand, D., Riesterer, N., & Ragni, M. (2022). Model-based explanation of feedback effects in syllogistic reasoning. *Topics in Cognitive Science*, 14(4), 828–844. doi: <https://doi.org/10.1111/tops.12624>
- Chater, N., & Oaksford, M. (1999). The probability heuristics model of syllogistic reasoning. *Cognitive Psychology*, 38(2), 191–258.
- Jahn, G., Knauff, M., & Johnson-Laird, P. (2007). Preferred mental models in reasoning about spatial relations. *Memory & Cognition*, 35(8), 2075–2087. doi: 10.3758/BF03192939
- Johnson-Laird, P. (1975). Models of deduction. In R. J. Falmagne (Ed.), *Reasoning: Representation and process in children and adults* (pp. 7–54). New York, US: Psychology Press.
- Johnson-Laird, P. (2006). *How We Reason*. Oxford, New York: Oxford University Press.
- Johnson-Laird, P. (2010). Mental models and human reasoning. In *National academy of sciences* (Vol. 107, pp. 18243–18250).
- Khemlani, S., & Johnson-Laird, P. (2012). Theories of the syllogism: A meta-analysis. *Psychological Bulletin*, 138(3), 427–457.
- Khemlani, S., & Johnson-Laird, P. (2013). The processes of inference. *Argument & Computation*, 4(1), 4–20.
- Khemlani, S., & Johnson-Laird, P. (2016). How people differ in syllogistic reasoning. In A. Papafragou, D. Grodner, D. Mirman, & J. C. Trueswell (Eds.), *38th annual conference of the cognitive science society* (p. 2165–2170). Austin, TX: Cognitive Science Society.
- Khemlani, S., Lotstein, M., Trafton, J. G., & Johnson-Laird, P. N. (2015). Immediate inferences from quantified assertions. *The Quarterly Journal of Experimental Psychology*, 68(10), 2073–2096. doi: 10.1080/17470218.2015.1007151
- Oaksford, M., & Chater, N. (2001, aug). The probabilistic approach to human reasoning. *Trends in Cognitive Sciences*, 5(8), 349–357.
- Ragni, M., & Knauff, M. (2013). A theory and a computational model of spatial reasoning with preferred mental models. *Psychological Review*, 120(3), 561–588. doi: 10.1037/a0032460
- Rauh, R., Hagen, C., Knauff, M., Kuss, T., Schlieder, C., & Strube, G. (2005). Preferred and Alternative Mental Models in Spatial Reasoning. *Spatial Cognition and Computation*, 5, 239–269. doi: 10.1080/13875868.2005.9683805
- Störring, G. (1908). *Experimentelle untersuchungen über einfache schlussprozesse*. W. Engelmann.

Modelling the Effects of ACT-R Working Memory Demands on Accuracy Rates of Relational Reasoning Problems

Nico V. Turcas (nico.turcas@carleton.ca)

Jim Davies (jim@jimdavies.org)

Robert L. West (robert.west@carleton.ca)

Department of Cognitive Science, Carleton University, Ottawa, ON K1S 5B6, Canada

Abstract

Relational reasoning is a core cognitive ability necessary for intelligent behaviour as it evaluates relationships between mental representations. Laboratory-based tasks of relational reasoning problems have long been used to investigate how individuals make inferences about such problems, with theories of mental models arguing that to solve such problems, individuals construct an integrated mental model based on the provided premises to generate or verify conclusions. Computational models of relational reasoning offer insights into how individuals generate such mental models and why some cognitive strategies may be preferred over others. However, many of these models do not directly account for what is often cited as a primary reason for the difficulty of different problems: the effects of increased working memory demand. In this paper, we present four ACT-R models that simulate the negative relationship between accuracy rates and relational problem complexity and demonstrate how different memory errors of omission and commission can account for qualitatively different reasoning processes. Our cognitive models demonstrate the importance of future work to consider individual differences in working memory processing, micro-strategy preferences, and the effects of different memory errors on the reasoning process.

Keywords: cognitive models; ACT-R; mental models; working memory; relational reasoning

Introduction

Relational reasoning is the cognitive ability to identify and evaluate relations between mental representations. Relational reasoning has long been of interest to cognitive scientists as it is crucial for problem-solving and fluid intelligence (Crone et al., 2009; Krawczyk, 2012). An effective method for investigating how reasoning about different relations occurs is examining how inferences are made from syllogistic deductions (Byrne & Johnson-Laird, 1989). These relational reasoning problems are laboratory-based tasks that require participants to evaluate a set of given premises and then generate or verify a logical conclusion. Consider, for example, the three premises below.

The apple is above the banana.
The banana is above the orange.
The lemon is below the orange.

Once presented with these three premises, one could confirm or produce a logical conclusion such as "the apple is above the lemon". These relational reasoning tasks are not restricted to spatial relations but can also include visual descriptions like "dirtier-cleaner" or abstract relations such as "better-worse", which can affect the reasoning process itself (Knauff & Johnson-Laird, 2002; Sima et al., 2013).

Mental Models and Relational Reasoning

Mental Model Theory is one of the most favoured frameworks for understanding how humans can make inferences to solve such relational reasoning problems (Johnson-Laird, 2010). Mental Model Theory proposes that when people reason about a problem, they do not hold onto information as separate pieces of information but rather leverage their visuospatial faculties to construct an integrated representation, i.e., a mental model, in working memory, which may be used to infer conclusions.

According to Mental Model Theory, the reasoning process is comprised of three stages: model comprehension, description, and validation (Johnson-Laird & Byrne, 1991). These are also referred to as model construction, inspection, and variation stages (Ragni & Knauff, 2013). During the model comprehension stage, a mental model integrates information from the premises using relevant general knowledge to represent the problem space. The constructed model is then inspected in the second stage of model description to evaluate a putative conclusion or relations that may not have been explicitly stated. Finally, a conclusion is generated or verified in the validation stage based on the constructed model. This final stage of reasoning also allows for the generation of variations of the mental model, should the premises permit it.

According to Mental Model Theory, the difficulty of the relational reasoning question is determined by the number of possible alternative models the reasoner constructs. It is argued that reasoners attempt to find alternative models of the presented premises, which may contradict the conclusion as false. Reasoners will iterate through each phase until all possible models are generated and examined. A conclusion is considered true if such a contradictory alternative model cannot be found (Johnson-Laird & Byrne, 1991). However,

this interpretation of difficulty has been challenged by theories such as the Preferred Model Theory (Ragni & Knauff, 2013), which argues that individuals only construct a single mental model in most situations and remain almost blind to other interpretations unless explicitly told to acknowledge alternatives. According to the Preferred Model Theory, the difficulty of a relational reasoning problem can be measured by the number of necessary operations the theoretical spatial focus system uses to solve a problem.

Research on relational reasoning problems has collectively found several factors that affect the difficulty between problem variations as measured by accuracy rates and reaction times. Core findings include the continuity effect, the figural effect, premise phrasing effects, and the difficulty of indeterminate problems (Byrne & Johnson-Laird, 1989; Johnson-Laird & Bara, 1984; Knauff et al., 1998). Indeterminate problems are those for which different mental models may be constructed based on the same premises. For example, the premises "A is to the left of B" and "C is to the right of A" lead to two possible figures: "A-B-C" or the equally valid model of "A-C-B". The difference between these two models demonstrates two separate micro-strategies of what Preferred Model Theory argues individuals may differ on.

Surprisingly, only recently has a study investigated the relationship between properties that affect relational reasoning. The Multidimensional Relational Reasoning Task (MRRT), developed by Cortes et al. (2021), consists of 90 problems that systematically vary on the following stimulus properties: number of premises (2 or 3), number of dimensions (1 or 2), relation type (spatial or non-spatial), solution (true, false, and indeterminate), premise order (continuous or discontinuous), conclusion phrasing ("A first" or "A second"). Cortes et al. (2021) demonstrated that reasoning problems containing more premises and multi-dimensional relations increased the difficulty of validating a conclusion (Figure 1). This was reflected in a decrease in accuracy rates and an increase in response times, and is credited to an increase in working memory demands due to additional premises and the number of relational dimensions per premise necessitating the construction of more complex mental models to be reasoned over (Cortes et al., 2021; Goodwin, & Johnson-Laird, 2005). Data provided by Cortes et al. (2021) and specific questions from the MRRT will be used to construct and compare our cognitive models in this paper and can be found at (<https://osf.io/qfvp2/>).

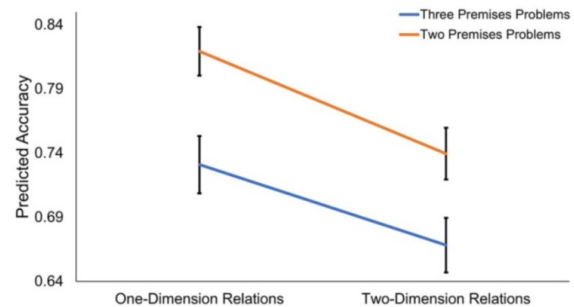


Figure 1: Predicted marginal means of Dimension x Premise interaction taken from Cortes et al. (2021).

Results of the mixed effects model indicate lower accuracy rates for more complex problems, with three premise two-dimension relational reasoning problems being associated with the lowest accuracy.

Computational Models of Relational Reasoning

Although psychological theories of mental models have received considerable support, their application to computational models of mental processing is often underspecified. Computational accounts of relational reasoning are preferable as they allow for a formal presentation of a theory with fully specified operations to process necessary information. Spatial and visual properties are often intertwined in mental representations, so computational imagery is often conceptualised as incorporating both properties through the ability to represent, retrieve, and reason about such information (Glasgow & Papadakis, 1992). Therefore, many representations can be used to model different types of relational reasoning problems. Using the *Cognitive Computation for Behavioural Reasoning Analysis (CCOBRA)* framework, Ragni et al. (2021) illustrated that despite differences in information processing assumptions, different spatial reasoning models can yield several predictions consistent with each other.

Interestingly, despite working memory being recognised as a key factor contributing to the heightened difficulty of relational reasoning problems, many computational models of spatial relational reasoning do not specify how this occurs. One approach to investigating the impact of working memory demands on reasoning with mental models is using cognitive architectures (Kostseruba & Tsotsos, 2020).

Preferred Inferences in Reasoning with Spatial Mental Models (PRISM)

One theory that has modelled in ACT-R (Anderson & Lebiere, 1998; Ritter et al., 2019), is the *Preferred*

Inferences in Reasoning with Spatial Mental Models (PRISM) theory (Ragni & Knauff, 2013). The Preferred Model Theory argues that individuals prefer to reason over a single mental model constructed with a specific layout. PRISM is intended to simulate and explain how individuals construct, inspect, and vary preferred mental models during the spatial relational reasoning process. ACT-R implementations of PRISM have successfully demonstrated order effects (Boeddinghaus et al., 2006) and predicted BOLD-responses of brain regions (Ragni et al., 2010).

PRISM operates on a spatial working memory representation that is operationalised as a spatial array structure and a spatial focus. The spatial focus can move forward, backward, right, and left, as well as insert or remove token objects into this spatial array and, in doing so, construct a mental model to reason over. PRISM reasons via binary relations provided by premises defined as a triple (X, r, Y) in which X is the to be located object (LO), r is the binary relation, and Y is the relatum, or reference object (RO). An example might be (Apple, above, Banana). At the beginning of the construction phase of a reasoning problem, PRISM's spatial focus begins at the coordinate position (0,0). As the premises are integrated, token insertion or deletion operations in the spatial model will be facilitated by the spatial focus until a mental model is generated.

One of the abilities of PRISM is that it can demonstrate and explain differences in model constructions of different individual strategies. In the *fff-strategy* (first free fit), the focus inserts a token at the first free position that fits, compared to the *ff-strategy* (first fit) in which the focus inserts the token at the first cell that fulfils the premise, squeezing it in between others that may occupy the space and relocating the other tokens. Experiments in Ragni and Knauff (2013) showed that despite both strategies being suitable for reasoning, individuals construct models according the *fff-strategy* significantly more than the *ff-strategy*. Studies have also demonstrated that non-spatial relations are often conceptually mapped onto spatial relations (Gattis & Holyoak, 1996; Tversky, et al., 1991). Therefore, PRISM is not limited to only reasoning about spatial relations, but can also demonstrate the previously mentioned strategy preferences in reasoning about non-spatial relations as well.

An attractive feature for facilitating PRISM is that, unlike many other theories of spatial relations, it provides a unit of difficulty measure through operations of its spatial focus so that problems that require more operations are theorised as being predicted to be more difficult and, subsequently, have lower accuracy rates of correct responses, along with lengthier response times. This can be used as a

theoretical basis for systematically raising the ACT-R threshold parameter to simulate working memory demand during the relational reasoning process. By working memory demand, we specifically mean the demands placed on the ACT-R conceptualization of working memory as information held in module buffers.

Two common memory errors studied in experimental psychology are errors of omission, a failure to remember, and errors of commission, in which a participant performs an incorrect or additional action. ACT-R allows us to model both types of errors (Kelly et al., 2000; Lebiere et al., 1994).

ACT-R Models of Working Memory Demands

The following describes our ACT-R models, which demonstrate how the effects of working memory demand may impact accuracy rates and the reasoning process through memory errors of omission and commission. To do this, we created four ACT-R models of four problems taken from the MRRT, which vary in the number of premises and dimensions but are all spatial relational reasoning problems of determinate and continuous premise orders.

ACT-R is often considered a hybrid cognitive architecture in that it incorporates symbolic and sub-symbolic operations. A full description of ACT-R is beyond the scope of this paper, but in every ACT-R model, there are generally two implicit theoretical commitments. The first commitment is to the theory of the cognitive architecture itself; in our case, this is Python ACT-R (Stewart & West, 2007). The second commitment is how knowledge is represented within this architecture that drives the agent - who may or may not be able to alter this knowledge over time through different learning.

Although other computational theories could have been used, PRISM was chosen as the primary theory of knowledge representation for our ACT-R models because it provides a metric of difficulty through accumulated operations of its spatial focus. PRISM does not set a decay parameter to its models and can, therefore, maintain a model in the spatial array forever. To address this, we modelled the increase in working memory demands in a principled way by manipulating ACT-R declarative memory threshold parameters based on each operation of the spatial focus in accordance with PRISM theory. Each operation of the spatial focus increasing ACT-R's threshold parameter therefore decreases the probability of a successful chunk recall. This threshold increase is intended to account for the increased demand on the memory system of the cognitive agent; we are not committed to what this is would mean on a mechanistic level apart from the ACT-R theory of declarative and working memory systems. Errors of omission in our models

occur when a chunk in long-term declarative memory cannot gather enough activation to be retrieved. Errors of commission are facilitated by ACT-R's partial matching system so that names and locations of objects within the mental model may be incorrectly retrieved for one another.

Processing example of the ACT-R Models

All ACT-R models begin their construction phase at default parameter settings, and mental models are constructed based on information found in premises. Consider question six of the MRRT, a two premise one dimension question.

MRRT Question: six (Two Premise, One Dimension)

P1) Edward is to the left of Derek.

P2) Derek is to the left of Travis.

C) Edward is to the left of Travis (solution: True)

The model begins by firing a production to read the first premise, "Edward is to the left of Derek". The spatial focus, which has been implemented as an ACT-R buffer, begins by inserting Edward at coordinates (0,0), moves a cell to the right, and inserts Derek at coordinates (0,1), with each new operation increasing the threshold parameter. For each new token object inserted into the mental model, a chunk is added to the ACT-R declarative memory module with the slot-pair values of the individual's name and the coordinates they reside. Now that Edward and Derek have been inserted into the mental model, the next premise is read, "Derek is to the left of Travis". Since Derek was the last object inserted into the mental model, the agent's spatial focus buffer is already on Derek, so the agent need only move one cell to the right to insert Travis at coordinate (0,2). Finally, the conclusion is read, and the agent must now enter the second stage of verification of the model: Edward - Derek - Travis.

The conclusion to be verified in this case is "Edward is to the left of Travis", but since the spatial focus buffer is already on Travis, the model must recall the location of Edward and fire a declarative memory request production. In our models, every declarative memory request introduces a branching of logic to select the next possible production to be fired. In these ACT-R models, three possibilities exist after a declarative memory request is made. The first possibility is that Edward's location is correctly recalled. The second possibility is that a memory error of omission occurs in which Edward's location cannot be remembered. The third possibility is a memory error of commission in which Edward's location is misremembered for another - in this case, it would be Derek's since the spatial focus is still on Trevor. Because the spatial focus must move left from Trevor's position (2,0), all objects within the array left of the

spatial focus are candidates for declarative memory retrieval.

If the declarative memory request is successful, a production fires, printing Edwards location of (0,0). The spatial focus moves toward coordinate (0,0), and once it is on Edward, a production validates the conclusion as "True, Edward is Left of Travis!". The second possibility of an error of omission occurs when a declarative memory chunk cannot reach the threshold for successful retrieval, so the agent realises they cannot remember the object's location within the mental model. It is difficult to ascertain what occurs at this moment as individuals may rely on a wide variety of strategies to overcome such an error of omission. When such memory errors occur in our models, all productions involving conclusions have a random equal chance of selecting a true or false verification. Finally, the third possibility that may occur instead of a successful memory request or memory error of omission is a memory error of commission in which the ACT-R agent would believe that Edward's location is incorrectly Derek's, at coordinate (1,0). Notice, however, that this interestingly would result in a true conclusion verification despite the incorrect location of the to-be-located object. The occurrence of a memory error of commission, yet still inferring a correct conclusion, highlights an overlooked problem in the literature of how different memory errors may result in different responses or solutions but still be reasoned as true. In the case of our example, this issue is especially pertinent and possibly problematic, as our models also predict that current accuracy rates of spatial relational reasoning problems may be overinflated due to false positive declarative memory recalls due to errors of commission.

ACT-R Model Results

Four ACT-R models of relational reasoning were created to simulate the findings in Cortes et al. (2021) of decreased accuracy rates for more complex conditions of increased premises and dimension problems (2P1D, 3P1D, 2P2D, 3P2D). All models began the construction phase with the same default parameter values and facilitated using PRISM to model the knowledge representations by which the ACT-R agent reasons. For each focus buffer operation, ACT-R's threshold parameter increases by an equivalent rate to model working memory demands in a principled way, and to be in accordance with PRISMS accumulated number of focus operations as a measurement of difficulty. In doing so, our models can simulate memory errors of omission and commission, which subsequently have a qualitative impact on the reasoning process itself, and a quantitative effect on accuracy rates of each complexity condition.

To test the ACT-R models, all four were run equivalent to the number of participants in Cortes et al.'s (2021) *X* condition ($n=310$) to receive an average proportion of accuracy per model. This was then repeated for a total of ($n=30$) accuracy rate measures per complexity condition. Although our ACT-R models were based only on four spatial relational reason questions of the MRRT (questions 6, 36, 21, and 66), the results replicated the negative trend in accuracy rates decreasing with each complexity condition in both spatial and non-spatial questions alike. The output of our ACT-R models was: 2P1D ($M=83.76$, $SD=2.29$), 3P1D ($M=80.93$, $SD=1.96$), 2P2D ($M=75.19$, $SD=2.59$), 3P2D ($M=66.34$, $SD=2.36$), which closely matches the results of human participant data provided by Cortes et al. (2021)., see Figure 2.

To examine correlations between our ACT-R models and participant data, both spatial and non-spatial, we drew 100 samples from separate Gaussian distributions of each category. Again, even with our ACT-R models only modelling four spatial relational reasoning questions, this was sufficient for a strong positive correlation between distributions of both spatial relations ($n=100$, $r=0.62$) as well as non-spatial relations ($n=100$, $r=0.64$), as predicted. A regression analysis revealed a negative correlation between accuracy rates and the accumulated number of focus operators, with the total number of accumulated operations accounting for a larger proportion of the variance in accuracy rates in the ACT-R models ($R^2 = 0.85$) compared to the simulated human data of spatial problems ($R^2=0.48$), as well as non-spatial ($R^2=0.42$).

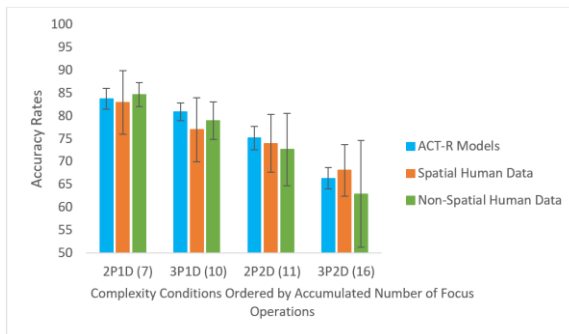


Figure 2: Results of the four ACT-R models compared to participant data from Cortes et al. (2021) for spatial and non-spatial problems. The results are depicted with standard deviations and ordered according to complexity conditions based on the accumulated number of focus operations per problem for a correct response.

Discussion

This paper sought to investigate how different types of memory errors due to increased working memory demand may be modelled to simulate the accuracy rates of relational reasoning problems of varying complexity. Four ACT-R models were created, each based on a different question of complexity from the Multidimensional Relational Reasoning Task (MRRT) (Cortes et al., 2021). Preferred Inference in Reasoning with Spatial Mental Models (PRISM) (Ragni & Knauff, 2013) was implemented in our ACT-R models to simulate the reasoning process of the agent, the chunk value representation of knowledge used by the ACT-R models, and the accumulated number of focus operations allowed for a systematic increase of the ACT-R threshold parameter. Since indeterminate question phrasing was found to be a factor for lower accuracy by Cortes et al. (2021), we opted to exemplify the effects of our models through determinate questions only and to leave indeterminate problems for future work.

Our ACT-R models successfully demonstrated that the effects of working memory demand, as conceptualized by ACT-R theory, could be simulated to model accuracy rates of various relational reasoning problems, and in doing so, our models provide several motivations for future work. Even though our ACT-R models were based on spatial relational reasoning questions from the MRRT, they were still found to be satisfactory models of determinate non-spatial relational reasoning problems. When controlling for all other stimulus properties, Cortes et al. (2021) did find a main effect of relation type of problem difficulty, something other studies have often found null relationship-type effects (Carreiras & Santamaria, 1997). More often, however, the effects of visual imagery hindering the reasoning process, as argued by the *visual impedance hypothesis*, are judged by reaction times (Knauff & Johnson-Laird, 2002). Individual differences have been found regarding the degree to which mental imagery may play in the reasoning process by means of the degree of individual mental imagery vividness (Gazzo Castaneda & Knauff, 2013; Knauff & May, 2006). The impact of mental imagery vividness may be a future avenue to pursue in modelling individual reasoners, however, there is some disagreement on how this might be done (Albrecht et al., 2015). PRISM is largely motivated to model individual strategy preferences, and so is well poised for future modelling of individual differences, especially within cognitive architectures.

A limitation of our ACT-R models is that the only way an incorrect conclusion may be provided is through occurrences of memory errors of omission, at which point there is an equivalent chance of the ACT-

R agent providing a correct or incorrect response. There may be differences in how individuals recover from a memory error of omission, and because Cortes et al. (2021) desired to collect normative reaction time data, they elected not to impose time constraints, which provides even more possible individual strategies, especially when motivated to ease the cognitive burden of increased working memory load. Only errors of omission could provide incorrect conclusions because of the way spatial determinate relational reasoning questions are structured in the MRRT. The spatial relational reasoning problems of the MRRT always have the reference object be the last object inserted in the mental model, such as: A-B-C-(D). This holds true even if one represents the non-spatial reasoning problems in a spatial format. With the PRISM spatial focus being on this last object, any to-be-located object is always in the same direction of all other possible objects to be recalled. Therefore, our models support the idea that any error of commission will result in the same conclusion even if the wrong to-be-located object is recalled. Current psychological measures of relational reasoning do not account for these differences in reasoning.

Future work should seek to construct models which consider individual differences in varying domains such as working memory capacity, imagery vividness, micro-strategies, how memory errors affect the reasoning process, and how individuals handle such memory error occurrences.

References

- Albrecht, R., Schultheis, H., & Fu, W.-T. (2015). Visuo-Spatial Memory Processing and the Visual Impedance Effect. In P. Bello, M. Guarini, M. McShane, & B. Scassellati (Eds.), *Proceedings of the 37th Annual Conference of the Cognitive Science Society* (pp. 59-64). Austin, TX: Cognitive Science Society.
- Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Erlbaum.
- Boeddinghaus, J., Ragni, M., Knauff, M., & Nebel, B. (2006). Simulating Spatial Reasoning Using ACT-R. *Proceedings of the ICCM 06*, 62-67.
- Byrne, R. M. J., & Johnson-Laird, P. N. (1989). Spatial reasoning. *Journal of Memory and Language*, 28(5), 564-575. [https://doi.org/10.1016/0749-596x\(89\)90013-2](https://doi.org/10.1016/0749-596x(89)90013-2)
- Carreiras, M., & Santamaría, C. (1997). Reasoning about relations: Spatial and nonspatial problems. *Thinking & Reasoning*, 3(3), 191-208. <https://doi.org/10.1080/135467897394347>
- Cortes, R. A., Weinberger, A. B., Colaizzi, G. A., Porter, G. F., Dyke, E. L., Keaton, H. O., Walker, D. L., & Green, A. E. (2021). What Makes Mental Modeling Difficult? Normative Data for the Multidimensional Relational Reasoning Task. *Frontiers in psychology*, 12, 668256. <https://doi.org/10.3389/fpsyg.2021.668256>
- Crone, E. A., Wendelken, C., van Leijenhorst, L., Honomichl, R. D., Christoff, K., & Bunge, S. A. (2009). Neurocognitive development of relational reasoning. *Developmental science*, 12(1), 55-66. <https://doi.org/10.1111/j.1467-7687.2008.00743.x>
- Gattis, M., & Holyoak, K. J. (1996). Mapping conceptual to spatial relations in visual reasoning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22(1), 231-239. <https://doi.org/10.1037/0278-7393.22.1.231>
- Gazzo Castaneda, L. E., & Knauff, M. (2013). Individual Differences, Imagery and the Visual Impedance Effect. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 35. <https://escholarship.org/uc/item/11n377xb>.
- Glasgow, J. and Papadias, D. (1992), Computational Imagery. *Cognitive Science*, 16: 355-394. https://doi.org/10.1207/s15516709cog1603_2
- Goodwin, G. P., & Johnson-Laird, P. N. (2005). Reasoning about relations. *Psychological Review*, 112(2), 468-493. <https://doi.org/10.1037/0033-295X.112.2.468>
- Johnson-Laird P. N. (2010). Mental models and human reasoning. *Proceedings of the National Academy of Sciences of the United States of America*, 107(43), 18243-18250. <https://doi.org/10.1073/pnas.1012933107>
- Johnson-Laird, P. N., & Bara, B. G. (1984). Syllogistic inference. *Cognition*, 16(1), 1- 61. [https://doi.org/10.1016/0010-0277\(84\)90035-0](https://doi.org/10.1016/0010-0277(84)90035-0)
- Johnson-Laird, P. N., & Byrne, R. M. J. (1991). *Deduction*. Lawrence Erlbaum Associates, Inc.
- Kelley, T., P. Wiley, and F. Lee, (2000). Developing an ACT-R model of mental manipulation. Army Research Laboratory Tech. Rep. ARL-TR-2179
- Knauff, M., & Johnson-Laird, P. N. (2002). Visual imagery can impede reasoning. *Memory & Cognition*, 30(3), 363-371. <https://doi.org/10.3758/BF03194937>
- Knauff, M., & May, E. (2006). Mental imagery, reasoning, and blindness. *The Quarterly Journal of Experimental Psychology*, 59(1), 161-177. <https://doi.org/10.1080/17470210500149992>
- Knauff, M., Rauh, R., Schlieder, C., & Strube, G. (1998). Continuity effect and figural bias in spatial relational inference. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society* (pp. 573-578). Mahwah, NJ: Lawrence Erlbaum Associates.

- Kotseruba, I., & Tsotsos, J. K. (2020). 40 years of cognitive architectures: Core cognitive abilities and practical applications. *Artificial Intelligence Review*, 53(1), 17–94
- Krawczyk D. C. (2012). The cognition and neuroscience of relational reasoning. *Brain research*, 1428, 13–23. <https://doi.org/10.1016/j.brainres.2010.11.080>
- Lebiere, C., Anderson, J. R., & Reder, L. M. (1994). Error modeling in the ACT-R production system. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, 555–559. Hillsdale, NJ: Erlbaum.
- Ragni, M., Brand, D., & Riesterer, N. (2021). The Predictive Power of Spatial Relational Reasoning Models: A New Evaluation Approach. *Frontiers in psychology*, 12, 626292. <https://doi.org/10.3389/fpsyg.2021.626292>
- Ragni, M., Fangmeier, T., & Brüssow, S. (2010). Deductive spatial reasoning: From neurological evidence to a cognitive model. *Cognitive Science*, 34(8), 1517–1541.
- Ragni, M., & Knauff, M. (2013). A Theory and a Computational Model of Spatial Reasoning with Preferred Mental Models. *Psychological Review*, 120(3), 561–588. <https://doi.org/10.1037/a0032460>
- Ritter, F. E., Tehranchi, F., & Oury, J. D. (2019). ACT-R: A cognitive architecture for modeling cognition. *Wiley interdisciplinary reviews. Cognitive science*, 10(3), e1488. <https://doi.org/10.1002/wcs.1488>
- Sima, J. F., Schultheis, H., & Barkowsky, T. (2013). Differences between Spatial and Visual Mental Representations. *Frontiers in psychology*, 4, 240. <https://doi.org/10.3389/fpsyg.2013.00240>
- Stewart, T. C., & West, R. L. (2007). Deconstructing and reconstructing ACT-R: Exploring the architectural space. *Cognitive Systems Research*, 8(3), 227–236. <https://doi.org/10.1016/j.cogsys.2007.06.006>
- Tversky, B., Kugelmass, S., & Winter, A. (1991). Cross-cultural and developmental trends in graphic productions. *Cognitive Psychology*, 23(4), 515–557. [https://doi.org/10.1016/0010-0285\(91\)90005-9](https://doi.org/10.1016/0010-0285(91)90005-9)

Alleviating 4 Million Cold Starts in Adaptive Fact Learning

Maarten van der Velde^{1,3} (m.a.van.der.velde@rug.nl), Florian Sense¹ (f.sense@rug.nl),
Jelmer Borst² (j.p.borst@rug.nl), & Hedderik van Rijn^{1,3} (d.h.van.rijn@rug.nl)

¹Dept. of Experimental Psychology & Behavioural and Cognitive Neuroscience, University of Groningen, the Netherlands

²Bernoulli Institute, Dept. of Artificial Intelligence, University of Groningen, the Netherlands

³SlimStampen B.V., Groningen, the Netherlands

Keywords: cold start problem; learning and memory; individual differences; ACT-R; Bayesian modelling

Introduction

Adaptive learning systems enable any learner to study at a level that is appropriately challenging to them. The adaptive nature of such a system is typically realised through learner- and material-specific parameters within the system's internal model, describing the knowledge state or ability of the learner and the difficulty of the material. The success of an adaptive learning system hinges on the accuracy of these parameter estimates—a poorly calibrated system may present material that is too easy or too hard, or give excessive or insufficient feedback and support.

The *cold start problem* occurs whenever an adaptive system has not yet had the opportunity to adapt to its user or content. The current study focuses on the cold start problem in *SlimStampen*, an adaptive learning system for acquiring and rehearsing declarative knowledge through spaced retrieval practice (see Sense, Behrens, Meijer, & van Rijn, 2016, for a description). In this system, trials are scheduled on the basis of an ACT-R model of the learner's memory (Anderson, 2007): facts are rehearsed when their simulated activation is about to drop below the retrieval threshold (Fig. 1A). SlimStampen adapts to the learner and study material by maintaining an individual estimate of the *rate of forgetting* (α) of each fact for that learner. The value of α influences how quickly a fact's activation decays and thereby how soon it is repeated. When a learner first starts studying a new set of facts, the rate of forgetting of each of these facts for this specific learner is still unknown. The system therefore uses a default starting estimate, which it continually revises based on the accuracy and speed of the learner's responses. Over time, it identifies which facts are difficult to memorise for the learner (facts with a high α) and which facts are easy (low α). If there is an initial mismatch between the system's estimates and reality, facts are repeated sooner than they need to be—not an efficient use of study time—or too late—frustrating and detrimental to learning.

The current study evaluates four different methods for alleviating the cold start problem. These methods all use prior learning data to predict rate of forgetting in future learning sessions, but they do so using different subsets of the data: all prior data (*Domain*), other learners studying the same fact (*Fact*), the same learner studying other facts (*Learner*),

and a combination of the latter two (*Fact & Learner*). In a previous lab-based study, we found that using the predicted values as starting estimates for α resulted in better learning, as the system allocated study time more effectively from the start (van der Velde, Sense, Borst, & van Rijn, 2021). Here, we test the same principle at a much larger scale in a real-world scenario: secondary school students practising foreign-language vocabulary.

Methods

We performed a post-hoc simulation study on a large set of retrieval practice data from the *SlimStampen* adaptive fact learning system, containing over 98 million trials from about 140 thousand secondary school students in the Netherlands. The data were highly varied, covering a range of year groups (years 1–4; ages 12–16) and education levels (pre-vocational *vmbo*, general secondary *havo*, and pre-university *vwo*), and vocabulary in two different languages (English and French).

We grouped the data into *learning sequences* (Fig. 1B), each containing the complete set of trials in which a particular learner studied a particular fact. We then created an 80%/20% train-test split, allocating learning sequences in their entirety to one of the two sets. Rate of forgetting predictions were made for the 4.6 million sequences in the test set by feeding different subsets of the training data (see above) into a Bayesian model that estimates $\alpha \sim \mathcal{N}(\mu, \lambda^{-1})$; the estimated mean (μ) becoming the predicted value. This process is illustrated in Fig. 1C and described in detail in van der Velde et al. (2021). The quality of the predictions was assessed in two ways (Fig. 1D). Firstly, we compared predicted to observed α . Secondly, we simulated the effect of using predicted α values as starting estimates in the learning session, looking specifically at the accuracy of the ACT-R model's behavioural predictions on the first delayed repetition of a fact within a learning session.

Results

Fig. 2 confirms that predicting α using prior learning data leads to more accurate estimates than using the default prediction. It also shows that the largest gains in predictive accuracy come from taking individual differences in difficulty between facts into account, more so than accounting for individual differences in ability between learners. As Fig. 3 shows, the behavioural predictions made by the ACT-R model when using predicted α values as starting estimates tell a similar story: the

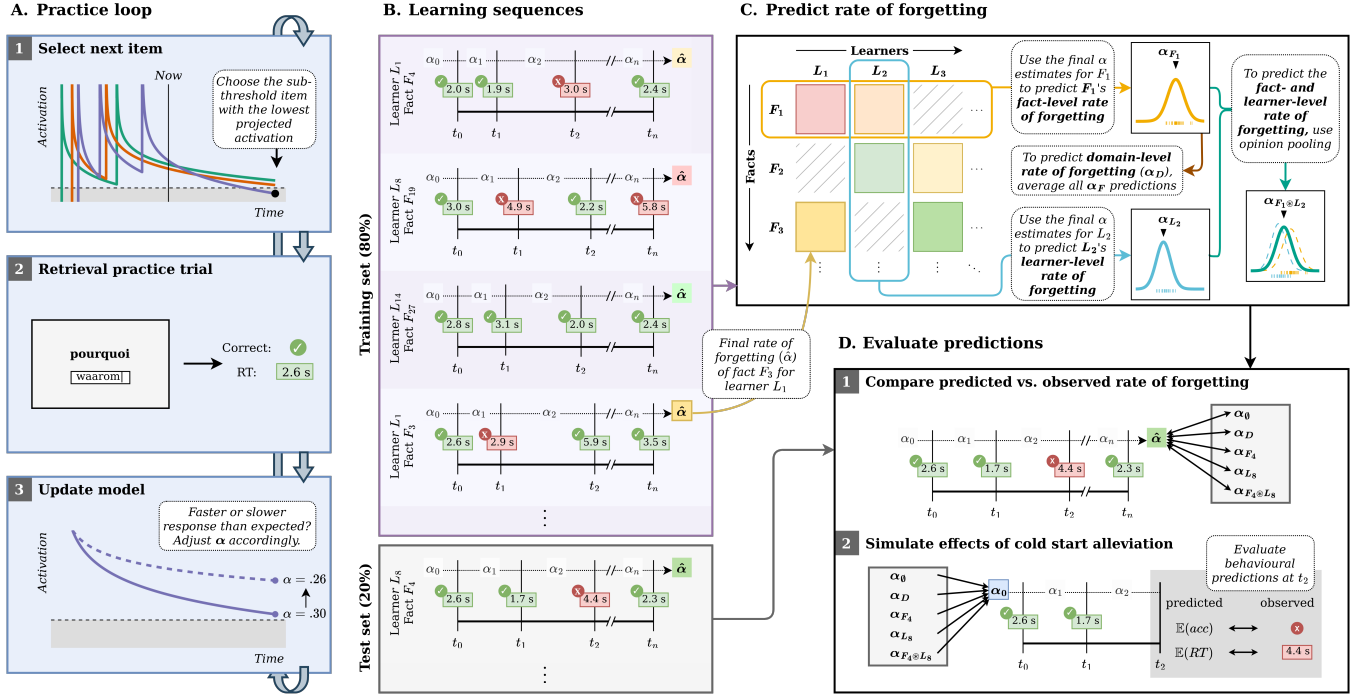


Figure 1: Visual summary of the current study. **A.** The learning system completes a three-step practice loop in each trial. **B.** Each learning sequence contains all trials of a learner L studying a fact F at times t_0, \dots, t_n and yields a final rate of forgetting estimate $\hat{\alpha}$. 80% of these estimates are used for training. **C.** Rate of forgetting predictions are made for the remaining 20% of learning sequences using different subsets of the training data. **D.** Predictions are evaluated by (1) comparing final rate of forgetting estimates to predicted values (where α_0 is the default value of 0.3), and (2) simulating the model's behavioural predictions when using predicted rates of forgetting as starting estimates.

default prediction is outperformed by all data-based prediction methods, and methods that involve fact-specific predictions perform best.

Conclusion

It is possible to predict rates of forgetting from prior learning data, and to use these predictions to improve item scheduling in an adaptive fact learning system. The observed improvements in prediction accuracy are similar in magnitude to those in our earlier lab study, where we found that using predicted α values as starting estimates in a learning session increased posttest retention by 6.8 percentage points. We expect that comparable retention gains can be achieved in real-world educational practice.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford, UK: Oxford University Press.

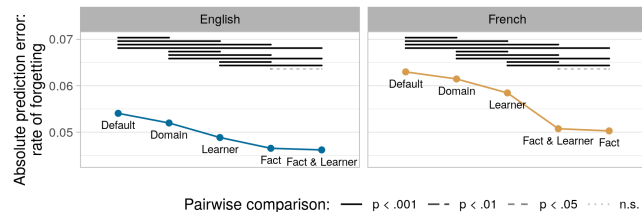


Figure 2: Rate of forgetting (α) prediction error.

Sense, F., Behrens, F., Meijer, R. R., & van Rijn, H. (2016). An Individual's Rate of Forgetting Is Stable Over Time but Differs Across Materials. *Topics in Cognitive Science*, 8(1), 305–321. doi: 10.1111/tops.12183

van der Velde, M., Sense, F., Borst, J., & van Rijn, H. (2021). Alleviating the Cold Start Problem in Adaptive Learning using Data-Driven Difficulty Estimates. *Computational Brain & Behavior*, 4(2), 231–249. doi: 10.1007/s42113-021-00101-6

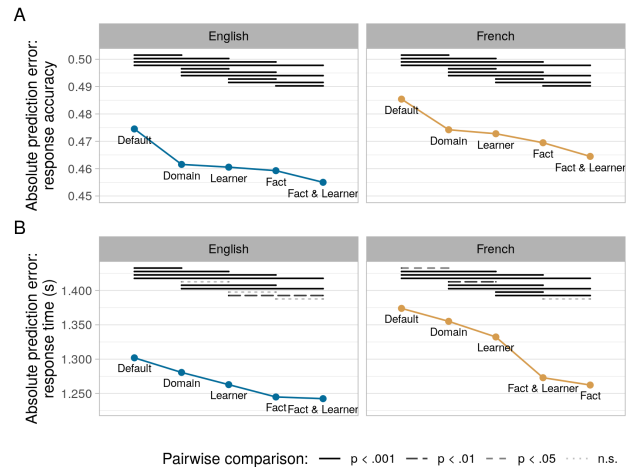


Figure 3: Behavioural prediction error on the first delayed repetition: (A) response accuracy, and (B) response time.

A Neural Network Simulation of Event-Related Potentials in Response to Syntactic Violations in Second-Language Learning

Stephan Verwijmeren (stephan.verwijmeren@ru.nl)

Department of AI, Radboud University
Nijmegen, The Netherlands

Stefan L. Frank (stefan.frank@ru.nl)

Centre for Language Studies, Radboud University
Nijmegen, The Netherlands

Hartmut Fitz (hartmut.fitz@mpi.nl)

Max Planck Institute, Radboud University
Nijmegen, The Netherlands

Yung Han Khoe (yunghan.khoe@ru.nl)

Centre for Language Studies, Radboud University
Nijmegen, The Netherlands

Abstract

Event-related potentials (ERPs) are used to study how language is processed in the brain, including differences between native (L1) and second-language (L2) comprehension. In low-proficiency L2 learners, syntactic violations give rise to an N400, but this changes into a P600 as their L2 proficiency increases. The precise functional interpretation of ERPs, however, remains a matter of debate. Fitz and Chang (2019) proposed a theory where ERPs reflect learning signals that arise from mismatches in predictive processing. These signals are propagated across the language system to make future predictions more accurate. We test if this theory can account for the N400-to-P600 switch in late bilinguals, by implementing a model capable of simulating the N400 and P600. We perform an experiment designed to elicit a P600 effect in simulated L2 learners progressing through learning stages. Simulated Spanish-English participants showed similar ERP effects in their L2 (English) as human participants did in ERP studies. Over the course of L2 learning, simulated N400 size decreased while P600 size increased, as it does in humans. Our findings support the viability of error propagation as an account of ERP effects, and specifically of how these can change over L2 learning.

Keywords: Event-related potential; N400; P600; prediction error; bilingualism.

Introduction

Psycholinguistic studies investigating neural mechanisms underlying adult second-language (L2) learning and processing often use electroencephalography (EEG), a technique for recording electrical voltage potentials produced by neural activity. Recorded potentials can be analyzed in relation to cognitive events, and can yield interpretable patterns called event-related potentials (ERPs) (Morgan-Short, 2014). ERP effects have been observed in response to syntactic violations in first language (L1) processing, as an increased positivity in the ERP waveform that starts around 600 ms after observing an anomalous word, as compared to its correct counterpart (Osterhout and Mobley, 1995). This effect is called a P600. Another ERP effect is reliably elicited in response to a lexico-semantic violation. This effect, called an N400, is a negative voltage deflection around 400 ms after an anomalous word,

as compared to a semantically appropriate word (Kutas and Hillyard, 1980).

ERP research has been done to find out if L2 learners show similar ERP effects as native speakers for morpho-syntactic and lexico-semantic processing. Research has shown that L2 learners can show native-like ERP waveforms for L2 grammatical features that are present in their L1 as well as for features unique to their L2 (Morgan-Short, 2014). ERPs of L2 learners differing in proficiency indicate that some learners progress through stages of syntactic learning, suggesting that there is an intermediate stage of learning between no L2 grammatical knowledge and grammaticalization (McLaughlin et al., 2010). The observed ERP effects differ between studies. Some L2 learning studies that investigated syntactic processing found an N400 for learners with low proficiency and a P600 for learners with high proficiency, suggesting that L2 learners might rely more on lexical processing at early learning stages (Alemán Bañón et al., 2014; Antonicelli and Rastelli, 2022; Díaz et al., 2016; Esfandiari et al., 2021; Grey, 2022; Mickan and Lemhöfer, 2020; Nichols and Joannis, 2019; Osterhout et al., 2008; Tanner et al., 2013, 2014). Other related studies found a similar effect for proficiency but ERPs were biphasic at low proficiency levels, resembling an N400 followed by a P600. With increasing proficiency, the amplitude of the N400 decreased and the P600 amplitude increased but ERP waveforms remained biphasic to a degree (Bian et al., 2021; Bowden et al., 2013; Caffarra et al., 2015; Esfandiari et al., 2020; Grey et al., 2018; McLaughlin et al., 2010; Morgan-Short et al., 2012; Morgan-Short, 2014; Pélissier et al., 2015). In the majority of studies, L2 proficiency was the most important factor determining ERP profiles (Antonicelli and Rastelli, 2022; Caffarra et al., 2015; McLaughlin et al., 2010; Morgan-Short, 2014).

Here we are interested in whether L2 learning stages reflect on the ERPs in simulated participants like in human participants. We do so by taking a monolingual computational

cognitive model of sentence production that has been used to explain ERPs, and extending it to the bilingual case.

Computational models of ERP effects

Several connectionist cognitive models have been proposed to explain the N400 ERP effect in sentence comprehension (see Eddine et al., 2022, for a review). Some of these take the magnitude of change in neural activation as a predictor of the N400 (Rabovsky et al., 2018) while others take the network's prediction error to account for N400 size (Brouwer et al., 2017; Fitz and Chang, 2019; Frank et al., 2015).

While a number of models can potentially explain the N400, the models by Brouwer et al. (2017) and Fitz and Chang (2019) are in addition able to model the P600. Specifically, Fitz and Chang (2019) used Chang's (2002) Dual-path model to show that prediction error corresponds to N400 size and backpropagated error corresponds to P600 size across a wide range of studies, providing support for the hypothesis that ERPs might reflect learning signals. This account of the N400 and P600 is known as the Error Propagation account.

The Dual-path model is a connectionist model of sentence production and syntactic development. The model has two pathways. The first pathway is the sequencing system that learns how words are ordered in a sentence and is based on the Simple Recurrent Network (Elman, 1990). The second pathway is a meaning system that learns how to map messages onto sentences in a target language. Previously, the Dual-path model was used to explain a wide range of sentence production phenomena in a number of different languages (Chang et al., 2006, 2015; Janciauskas and Chang, 2018; Tsoukala et al., 2017, 2021). For our studies, we used a bilingual extension of the Dual-path model (Tsoukala et al., 2021).¹

The present study

We perform a computational modelling experiment to investigate whether simulated L2 learners progress through stages of syntactic learning, and further test the viability of Error Propagation as an account of ERPs. We do this by ascertaining whether a P600 effect can be simulated by the Bilingual Dual-path model, and whether the magnitude of this effect increases in later L2 learning stages. We simulate native speakers of Spanish (L1) who start learning English (L2) from a later age. At every L2 learning stage, we run a subject-verb number agreement experiment similar to one of the experiments in Fitz and Chang (2019), presenting simulated participants with stimuli containing syntactic violations that elicit a P600 in native speakers (Osterhout et al., 2008; Tanner et al., 2013, 2014), and with control sentences without such violations.

We expect to find a simulated P600 effect in the Bilingual Dual-path model, since Fitz and Chang (2019) were able to have the monolingual Dual-path model reproduce N400 and P600 effects for stimuli used in a number of human EEG studies. We further expect N400 and P600 effects to occur and

their magnitude to decrease and increase, respectively, through learning stages, because ERP effects and their magnitude in L2 learners have been shown to be primarily determined by proficiency (Antonicelli and Rastelli, 2022; Caffarra et al., 2015; McLaughlin et al., 2010; Morgan-Short, 2014). We specifically expect the P600 effect to be more pronounced at later learning stages since advanced L2 learners show native-like ERP waveforms for L2 grammatical features (Morgan-Short, 2014). Additionally, we specifically expect the N400 effect to decrease in magnitude at later learning stages, because lexical learning precedes syntactic learning in L2 learners and L2 learners seem to rely on lexical processing early on because of this (McLaughlin et al., 2010).

Methods

To simulate late Spanish-English bilinguals, we trained the Bilingual Dual-path model (Figure 1) to learn Spanish from "infancy" and English as L2 at a later stage. The training input to the model consisted of sentences from two artificial languages (modelled on Spanish and English) that were paired with messages that encoded their meaning. The model learned to express messages as sentences of the target language (Spanish or English) by predicting the next word.

Artificial languages Table 1 shows the different constructions in the artificial languages. Constructions were distributed uniformly in the training input. Taken together, the two artificial languages consisted of 258 lexical items: 121 nouns, 11 adjectives, 6 pronouns, 6 determiners, 12 prepositions, 87 verbs, 7 auxiliary verbs, 6 verb inflectional morphemes, 1 plural noun marker, and the period. The inflectional morphemes were used to generate verbs with simple, progressive and perfect aspect in present or past tense. The plural noun marker was used to generate plural nouns.

The meaning space had 116 concepts and 7 thematic roles. Thematic roles are similar to those from Chang et al. (2006). To provide a simple example, the meaning of "the old lady carves a cake" would be represented as AGENT: LADY; ACTION-LINKING: CARVE; PATIENT: CAKE; AGENT-MODIFIER: OLD. This is implemented by introducing fixed-weight connections between role units and concept units (see Figure 1).

Model configuration and training For our simulations, we modified the Bilingual Dual-path model to resemble the architecture used in Fitz and Chang (2019): Previous word-history and role-history layers were added to the model which kept a running average of the activation of the input layer and role layer, respectively, and were connected to the hidden layer.

As pre-registered², all models used 50 hidden-layer units and 30 compress-layer units. Internal layer units used the logistic activation function; the output layer units used a softmax activation function. Weights were initialized randomly, uniformly between ± 1 . Fixed weights for concept-to-role

¹<https://gitlab.com/yhkhoe/bilingual-dual-path-/tree/ICCM2023>

²The pre-registration can be accessed here: https://aspredicted.org/blind.php?x=CGL_X3R

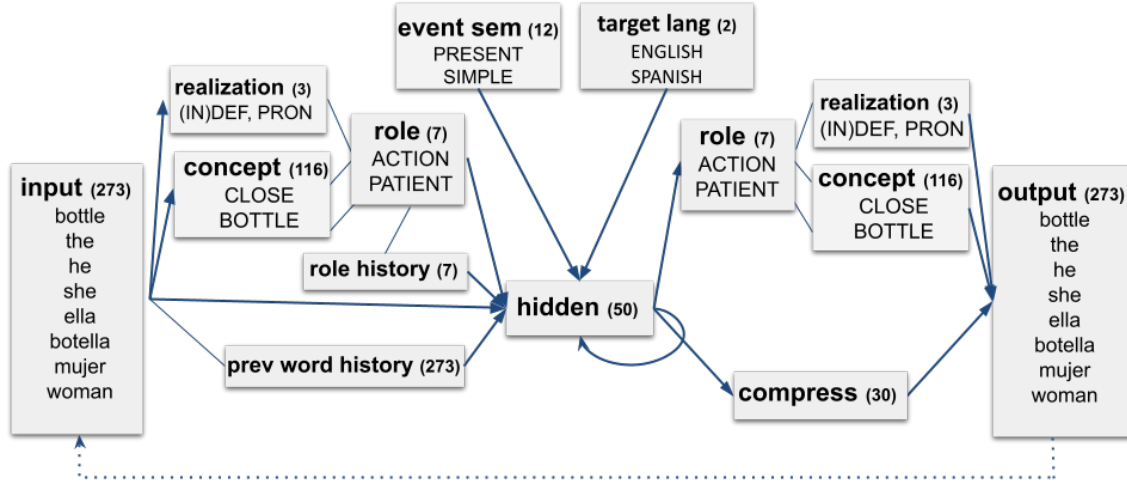


Figure 1: Architecture of the Bilingual Dual-path model. The model learns to map messages onto sentences in different languages by predicting the next word in its input. The sequencing system (lower path) maps from the input through a hidden layer to the output via a compression layer. The meaning system (upper path) uses information about thematic roles, concepts, and the realization of concept (e.g., by a pronoun or with an (in)definite determiner). The number of units per layer are shown in parentheses. Figure adapted from Tsoukala et al. (2021).

connections and realization-to-role connections were set to a value of 6. The concept layer had a set bias of -3 .

As pre-registered, for each of 60 model subjects and for Spanish and English combined, we generated 10,000 unique message-sentence pairs for training and a novel set of 200 message-sentence pairs for testing. The sentences are approximately equally divided over the two languages, where the percentage of Spanish sentences was sampled from a uniform distribution between 48% and 52% and the rest was English. Following Fitz and Chang (2019), the message was excluded from 70% of the training items. Each model first iterated five times over its monolingual Spanish training set, followed by 75 epochs over its bilingual training set. The training set's order was randomized at the beginning of each of these 80 epochs. The model learned by steepest descent backpropagation, with momentum set to 0.9. Initially, the learning rate was set to 0.1, it decreased linearly to 0.02 over the 5 epochs of monolingual training, and then stayed constant during bilingual training.

Model evaluation After each epoch, model accuracy was tested using a 200-sentence test set. The model's L2 English proficiency was evaluated with two measures. First, syntactic accuracy was measured as the percentage of sentences for which all words had the correct part of speech. Second, meaning accuracy was measured as the percentage of syntactically correct sentences that also conveyed the target message without additions. As pre-registered, we excluded the 20 subjects with the lowest meaning accuracy, leaving data from 40 model subjects.

Experimental trials To elicit ERPs, we generated 30 English sentence pairs, each consisting of a control and a violation item. The control was an active transitive sentence

where the verb form agreed with the subject in number. In the violation item, the verb did not agree with subject number. Violations were created by adding or omitting the inflectional marker for singular verbs (-ss), see Table 2.

Model subject differences Weights are initialized randomly, and differed between subjects. The percentage of Spanish versus English (training and testing) sentences varied between subjects, ranging from 48/52 to 52/48. The distribution of constructions is the same for all subjects. Training, testing and experimental trial sentences in the same language with the same constructions can differ between subjects in two ways. Firstly, sentences can differ in content-words resulting in different meaning of sentences. Secondly, singular nouns can differ in definiteness of the article.

Measuring model ERPs After every training epoch, the model was tested on the experimental sentence pairs. As in Fitz and Chang (2019), learning was turned on in the model during processing, but connection weights were reset to the weights of the respective training epoch after each test sentence in order to exclude learning effects during the experiment. The state in which the model encountered each trial was thus the same for all of the sentences.

We measured the prediction error at the output layer and the hidden layer (see Fitz and Chang, 2019, for details). The prediction error of output unit j is the difference between its activation y_j and the target activation t_j , or: $\delta_j = y_j - t_j$, with $y_j \in [0, 1]$ and $t_j \in \{0, 1\}$. This error was backpropagated in the network, as happens during training, to generate error at deeper layers. Error for units connected to the output layer was calculated as shown in Eq. 1, where k indexes the units connected to the output layer with weight w_{kj} , and j references

Table 1: Constructions with English example sentences. In the artificial language modelled on English, inflectional morphemes -prg, -prf and -ss are used for verb conjugations in progressive, perfect, and 3rd-person present simple tense, respectively.

Construction	Example sentence
Animate intransitive	The woman is play -prg
Animate with intransitive	The woman is play -prg with a dog
Inanimate intransitive	The apple is fall -prg
Locative	The boy is walk -prg around the school
Theme-experiencer (active)	The uncle surprise -ss the grandfather
Theme-experiencer (passive)	The grandfather is surprise -prf by the uncle
Transitive (active)	The girl bake -ss a cake
Transitive (passive)	The cake is bake -prf by the girl
Cause-motion	The hostess is put -prg a cactus into the office
Benefactive transitive	The grandmother repair -ss the cup for the girl
State-change	The waiter is fill -prg the cup with water
Locative alternation	The man spray -ss the sink with water

the units that are backpropagating error.

$$\delta_k = y_k(1 - y_k) \sum_{j=1}^n \delta_j w_{kj} \quad y_k \in [0, 1] \quad (1)$$

Error was calculated the same for other layers backpropagating error into the network. The error was collected after the transitive verb where the third-person singular morpheme was present or absent. The simulated N400 and P600 sizes are the sums over $|\delta|$ of the output- and hidden-layer units, respectively. Note that the scales of these two measures are not comparable because the output units, unlike the hidden units, use the softmax activation function and therefore their activations always sum to 1.

Table 2: Example sentences for the experimental trials. The bold morphemes indicate the sentence position where prediction error was measured.

Example sentence	Subject Nr	Agreement
the old lady carve -ss a cake	Singular	Control
the old lady carve a cake	Singular	Violation
the old lady -s carve a cake	Plural	Control
the old lady -s carve -ss a cake	Plural	Violation

Results

Figure 2 displays the proficiency of the model at the start and the end of bilingual training.

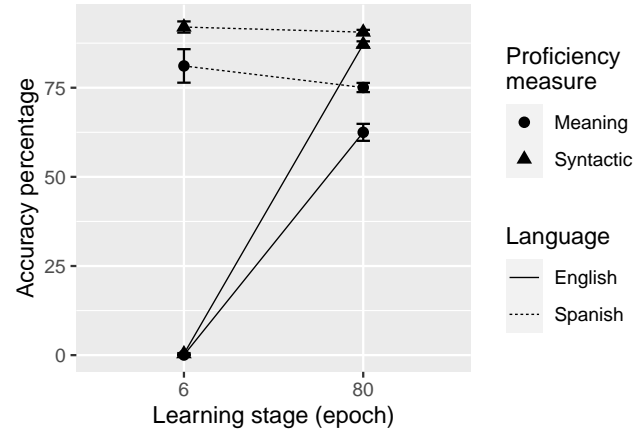


Figure 2: Mean proficiency of model. The syntactic and meaning accuracy are displayed for the first and last epoch of bilingual training. The error bars show the 95% confidence interval.

The mean prediction error over L2 learning stages at the hidden layer and the output layer are displayed in Figure 3, respectively. At the output layer, the mean error (simulating N400) for the VIOLATION items, was 1.89 at the start of bilingual training and increased to 1.93 at epoch 19, whereafter it decreased to 1.33 over the learning epochs. The mean error at the hidden layer (simulating P600) for the VIOLATION condition was 3.30 at the start of bilingual training, and increased over the learning epochs to 12.52. For the CONTROL items, error at both layers was high initially, but decreased to values close to 0 during L2 learning.

Pre-registered analysis

As pre-registered, we analyzed the data from our experiment with a linear mixed-effects model, using the lmer function from the package lme4 (Bates et al., 2015) in R (R Core Team, 2013). The model fits the prediction error from the Bilingual Dual-path model, a numerical value. The regression model³ included the predictors of interest: AGREEMENT, LAYER, LEARNING_STAGE and their interactions. AGREEMENT and LAYER were sum-coded. AGREEMENT levels Control and Violation were coded -1 and +1, respectively. Levels Hidden and Output of LAYER were coded +1 and -1, respectively. The number of L2 training epochs is indicated by the LEARNING_STAGE predictor, which was standardized. We fit random intercepts for model participants, and by-participant random slopes for the three predictors of interest and their interactions. Table 3 reports estimates, 95% confidence intervals,

³The script for the mixed-effects model can be accessed here: https://osf.io/yprjk/?view_only=aee2b8a52819475eb127721931de19ba

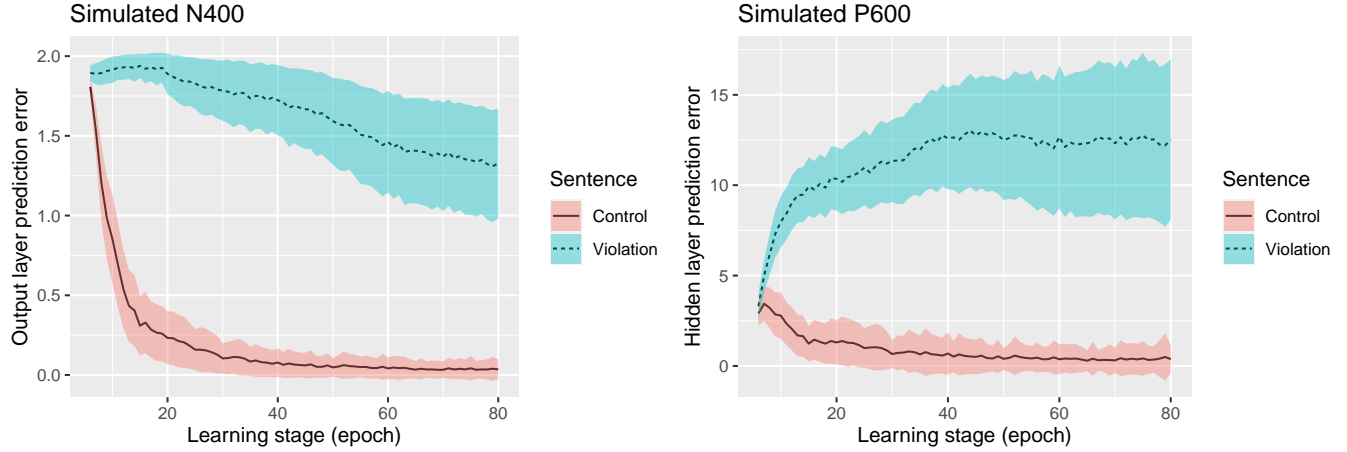


Figure 3: Mean prediction error (averaged over all model subjects) as a function of learning stage, in the output layer (left panel) and in the hidden layer (right panel), for number agreement violation and control items,. Shaded areas represent the 95% CI.

Table 3: Summary of the fixed effects in the linear mixed-effects models.

Predictor	Est.	95% CI	SE	df	<i>t</i> -value	Pr(> <i>t</i>)
Intercept	3.54	[3.32, 3.75]	0.11	40.00	33.84	<0.001
AGREEMENT	3.00	[2.81, 3.20]	0.10	40.05	30.76	<0.001
LAYER	2.61	[2.41, 2.82]	0.10	40.00	26.17	<0.001
LEARNING_STAGE	0.10	[-0.04, 0.24]	0.07	40.17	1.41	0.165
AGREEMENT:LAYER	2.28	[2.10, 2.46]	0.09	40.04	25.49	<0.001
AGREEMENT:LEARNING_STAGE	0.50	[0.36, 0.63]	0.07	40.15	7.31	<0.001
LAYER: LEARNING_STAGE	0.31	[0.19, 0.43]	0.06	40.18	5.08	<0.001
AGREEMENT:LAYER: LEARNING_STAGE	0.49	[0.37, 0.61]	0.06	40.16	8.34	<0.001

standard errors, degrees of freedom, *t*-values and *p*-values.

The positive estimate for the interaction between the predictors AGREEMENT, LAYER and LEARNING_STAGE (Estimate = 0.49, 95% CI = [0.37, 0.61]) indicates that the learning stages affect the two layers' sensitivity to violated sentences differently. The estimate has a confidence interval not including zero, thus there was an effect of the three-way interaction between these predictors. As Figure 3 clearly shows, this interaction is driven by an increasing effect of violation in the hidden layer combined with a decreasing effect of violation in the output layer.

Discussion

In the present work, we investigated whether simulated L2 learners progress through stages of syntactic learning. We used a connectionist model of syntactic development (Chang, 2002) to simulate Spanish-English bilinguals and exposed the model to L2 number-agreement violations at different points in time. Similar to the account in Fitz and Chang (2019), we recorded ERPs in response to these syntactically anomalous sentences from the model. On this account, ERPs are summary signals of brain activity that index the propagation of prediction error during comprehension whose functional role is to support

learning. Prediction error at the output layer was used to model the N400 and the backpropagated prediction error at the hidden layer was used to model the P600. The results of our simulations revealed a clear P600 effect for syntactically anomalous sentences in the L2, as well as a clear N400 effect early in acquisition. We also found that over time the P600 increased as the model became more proficient in the L2 and the N400 decreased over time. These findings are similar to human L2 learners as reported in several ERP studies on second language acquisition (Antonicelli and Rastelli, 2022; Caffarra et al., 2015; McLaughlin et al., 2010; Morgan-Short, 2014). Thus, our results support a theory of stages of syntactic learning in L2 learners where the magnitude of different ERP components changes during acquisition.

In our simulations, monolingual training resulted in optimal network weights for the L1, after which new L2 learning required a considerable amount of further training. At the beginning of L2 learning, the model does not know the English syntax for noun-verb number agreement. Consequently, after seeing the verb, the model activates a variety of candidate words and morphemes, which leads to large prediction error at the lexical output layer, and thus a large-amplitude N400 prediction for both violations and control sentences.

Prediction error at the hidden layer indexing the P600, in contrast, is relatively small because the model has not yet learned the syntax of agreement. As the model gradually acquires agreement, word predictions after the verb become increasingly more accurate because they are more and more driven by learned syntactic knowledge in the hidden layer. When the model is presented with a number agreement violation item, there is now a larger mismatch between the observed violation and the correct word predictions made by the model at this sentence position. Because the correct prediction is due to syntactic knowledge at the hidden layer, the hidden layer gets the majority of the blame when such a mismatch occurs. Thus, the size of the P600 effect increases during syntactic learning. The lexical output layer, on the other hand, gradually receives less blame as the syntax of agreement is acquired deeper in the network, which leads to a decrease in the N400 effect over time.

The error propagation account explains why ERPs elicited by lexical violations (N400) precede ERPs in response to syntactic violations (P600) and this account has been able to reproduce key findings from a considerable number of monolingual ERP studies (Fitz and Chang, 2019). The results presented here on bilingual ERPs, and how they change over development, adds further support for this account. Apart from the error propagation account, the model of Brouwer et al. (2017) can also explain monolingual N400 and P600 effects but it remains to be tested whether this model would be able to simulate ERP effects in bilinguals and the change in size of these effects during second language acquisition. What is unique about the error propagation account is that it can naturally model and explain ERPs in development because on this account ERPs are directly linked to learning. Therefore, the magnitude of ERP effects is expected to change as different pieces of linguistic knowledge are acquired. One limitation of the model is that it currently does not account for differences in the precise onset of the N400 or P600 and that it does not model earlier ERP components such as the early left-anterior negativity (eLAN) which has been elicited in some bilingual studies (Caffarra et al., 2015).

At present, it is unclear to what extent L1–L2 language similarity affects ERP effects in bilinguals. Some studies showed reduced P600 effects, or no P600 effect, for syntactic features that are instantiated differently between languages (Antoncelli and Rastelli, 2022; Liu et al., 2017; Morgan-Short, 2014), while other studies have shown P600 effects for syntactic L2 features regardless of L1–L2 similarity (Caffarra et al., 2015; McLaughlin et al., 2010; Morgan-Short, 2014). In future work, the proposed model will be used to shed more light on the role of language similarity in simulated bilinguals.

References

- Alemán Bañón, J., Fiorentino, R., and Gabriele, A. (2014). Morphosyntactic processing in advanced second language (L2) learners: An event-related potential investigation of the effects of L1–L2 similarity and structural distance. *Second Language Research*, 30(3):275–306.
- Antoncelli, G. and Rastelli, S. (2022). Event-related potentials in the study of L2 sentence processing: A scoping review of the decade 2010–2020. *Language Acquisition*, pages 1–38.
- Bates, D., Mächler, M., Bolker, B., Walker, S., et al. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(i01).
- Bian, J., Zhang, H., and Sun, C. (2021). An ERP study on attraction effects in advanced L2 learners. *Frontiers in Psychology*, 12.
- Bowden, H. W., Steinhauer, K., Sanz, C., and Ullman, M. T. (2013). Native-like brain processing of syntax can be attained by university foreign language learners. *Neuropsychologia*, 51(13):2492–2511.
- Brouwer, H., Crocker, M. W., Venhuizen, N. J., and Hoeks, J. C. (2017). A neurocomputational model of the N400 and the P600 in language processing. *Cognitive Science*, 41:1318–1352.
- Caffarra, S., Molinaro, N., Davidson, D., and Carreiras, M. (2015). Second language syntactic processing revealed through event-related potentials: An empirical review. *Neuroscience & Biobehavioral Reviews*, 51:31–47.
- Chang, F. (2002). Symbolically speaking: A connectionist model of sentence production. *Cognitive Science*, 26(5):609–651.
- Chang, F., Baumann, M., Pappert, S., and Fitz, H. (2015). Do lemmas speak German? a verb position effect in German structural priming. *Cognitive Science*, 39(5):1113–1130.
- Chang, F., Dell, G. S., and Bock, K. (2006). Becoming syntactic. *Psychological Review*, 113(2):234.
- Díaz, B., Erdocia, K., De Menezes, R. F., Mueller, J. L., Sebastián-Gallés, N., and Laka, I. (2016). Electrophysiological correlates of second-language syntactic processes are related to native and second language distance regardless of age of acquisition. *Frontiers in Psychology*, 7:133.
- Eddine, S. N., Brothers, T., and Kuperberg, G. R. (2022). The N400 in silico: A review of computational models. *The Psychology of Learning and Motivation*, page 123.
- Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.
- Esfandiari, L., Nilipour, R., Maftoon, P., and Nejati, V. (2021). Native-like event-related potentials in processing the second language syntax: Late bilinguals. *Caspian Journal of Neurological Sciences*, 7(2):51–59.
- Esfandiari, L., Nilipour, R., Nejati, V., Maftoon, P., and Khosrowabadi, R. (2020). An event-related potential study of second language semantic and syntactic processing: Evidence from the declarative/procedural model. *Basic and Clinical Neuroscience*, 11(6):841.
- Fitz, H. and Chang, F. (2019). Language ERPs reflect learning through prediction error propagation. *Cognitive Psychology*, 111:15–52.

- Frank, S. L., Otten, L. J., Galli, G., and Vigliocco, G. (2015). The ERP response to the amount of information conveyed by words in sentences. *Brain and Language*, 140:1–11.
- Grey, S. (2022). Variability in native and nonnative language: An ERP study of semantic and grammar processing. *Studies in Second Language Acquisition*, pages 1–30.
- Grey, S., Sanz, C., Morgan-Short, K., and Ullman, M. T. (2018). Bilingual and monolingual adults learning an additional language: ERPs reveal differences in syntactic processing. *Bilingualism: Language and Cognition*, 21(5):970–994.
- Janciauskas, M. and Chang, F. (2018). Input and age-dependent variation in second language learning: A connectionist account. *Cognitive Science*, 42:519–554.
- Kutas, M. and Hillyard, S. A. (1980). Reading senseless sentences: Brain potentials reflect semantic incongruity. *Science*, 207(4427):203–205.
- Liu, H., Dunlap, S., Tang, Y., Lu, Y., and Chen, B. (2017). The modulatory role of L1 and L2 morphosyntactic similarity during production of L2 inflected words: An ERP study. *Journal of Neurolinguistics*, 42:109–123.
- McLaughlin, J., Tanner, D., Pitkänen, I., Frenck-Mestre, C., Inoue, K., Valentine, G., and Osterhout, L. (2010). Brain potentials reveal discrete stages of L2 grammatical learning. *Language Learning*, 60:123–150.
- Mickan, A. and Lemhöfer, K. (2020). Tracking syntactic conflict between languages over the course of L2 acquisition: A cross-sectional event-related potential study. *Journal of Cognitive Neuroscience*, 32(5):822–846.
- Morgan-Short, K. (2014). Electrophysiological approaches to understanding second language acquisition: A field reaching its potential. *Annual Review of Applied Linguistics*, 34:15–36.
- Morgan-Short, K., Steinhauer, K., Sanz, C., and Ullman, M. T. (2012). Explicit and implicit second language training differentially affect the achievement of native-like brain activation patterns. *Journal of Cognitive Neuroscience*, 24(4):933–947.
- Nichols, E. S. and Joanisse, M. F. (2019). Individual differences predict erp signatures of second language learning of novel grammatical rules. *Bilingualism: Language and Cognition*, 22(1):78–92.
- Osterhout, L. and Mobley, L. A. (1995). Event-related brain potentials elicited by failure to agree. *Journal of Memory and Language*, 34(6):739–773.
- Osterhout, L., Poliakov, A., Inoue, K., McLaughlin, J., Valentine, G., Pitkanen, I., Frenck-Mestre, C., and Hirschensohn, J. (2008). Second-language learning and changes in the brain. *Journal of Neurolinguistics*, 21(6):509–521.
- Pélissier, M., Krzonowski, J., and Ferragne, E. (2015). Effect of proficiency on subject-verb agreement processing in french learners of english: An erp study. In *Proceedings of the International Conference of Experimental Linguistics, EXLing*.
- R Core Team (2013). Core R: A language and environment for statistical computing. *R Foundation for Statistical Computing, Vienna, Austria*. Online: <http://www.R-project.org>, 201.
- Rabovsky, M., Hansen, S. S., and McClelland, J. L. (2018). Modelling the N400 brain potential as change in a probabilistic representation of meaning. *Nature Human Behaviour*, 2(9):693–705.
- Tanner, D., Inoue, K., and Osterhout, L. (2014). Brain-based individual differences in online L2 grammatical comprehension. *Bilingualism: Language and Cognition*, 17(2):277–293.
- Tanner, D., McLaughlin, J., Herschensohn, J., and Osterhout, L. (2013). Individual differences reveal stages of L2 grammatical acquisition: ERP evidence. *Bilingualism: Language and Cognition*, 16(2):367–382.
- Tsoukala, C., Broersma, M., Van den Bosch, A., and Frank, S. L. (2021). Simulating code-switching using a neural network model of bilingual sentence production. *Computational Brain & Behavior*, 4(1):87–100.
- Tsoukala, C., Frank, S. L., and Broersma, M. (2017). “He’s pregnant”: Simulating the confusing case of gender pronoun errors in L2 English. In *the 39th Annual Meeting of the Cognitive Science Society*, pages 3392–3397. Cognitive Science Society.

Exploring Errors Towards a More Realistic Strategy Model

Shan N. Wang (sxw820@psu.edu)

Frank E. Ritter (fer2@psu.edu)

College of IST, Penn State
University Park, PA 16802 USA

Keywords: error analysis, cognitive modeling

Introduction

We present an example of how to model errors. We first analyze error patterns in a large dataset from a previous study (Ritter et al., 2022), then choose an appropriate error type, and build the corresponding error model.

The task, shown in Figure 1, is to find a single broken component in the BenFranklin Radar system (Ritter et al., 2022). The system is organized into five subsystems. It is implemented as a simulator (MENDS) with two levels of fidelity in Unity. We used the simpler version in this study. For each task, one of the components in the circuit, excluding the power supply, was broken. Participants were asked to find the broken component based on the light and switch conditions. However, participants can make mistakes and replaced the wrong one that is not broken. As an example, for task S2M (red solid squared), participant (PID) 413 ignored switch information and chose S1M (red dotted squared) that had switch off (green circled). This is also the error that we will model in this paper.

Participants in the study (N=111) were taught with an online tutor about the troubleshooting task and about the device and its interface (Ritter et al., 2022). Our definition of errors does not include variations of the strategies nor repetitive or extra steps. Those extra or wrong steps, corrected by participants or not, as long as those do not lead to an incorrect final replacement with feedback from the simulator interface, are not considered as errors, but just part of the problem-solving. We have come up with some strategies that participants may use (Wang & Ritter, 2023), but none of those strategies model errors while we know that participants make errors.

We dig deeper into errors, provides categorization, individual analyses, and an error generation and correction model. The errors here are extracted from the test session from the 111 participants. These are errors not corrected by the participants. Some actions during the fault-finding process can be hard to identify as errors because some actions may be errors or extra steps that are self-corrected, while some may be part of the strategies they use (e.g., a more thorough search that opens unnecessary subsystems). We start understanding human errors in this task as components that were not broken but were replaced.

The all-trays strategy (AllTrays) was developed based on participant P347. Their strategy walks across all trays while deciding the fault, then directly goes to a certain tray to locate the broken component. Components were assigned a number based on their distance from the power supply. The smaller the number, the closer the component to the power supply. The one that has the smallest number among the grey components is the broken one. This strategy ignores front panel information and uses both schematic knowledge and interface information, shown in Figure 2.

Modeling Human Errors

We present an example here. PID 421 matches with All-Trays strategy with trained learning in the test session. With current information, we only know that PID 421 in round 18 of the test session made an error of choosing a grey component that is not in the active path (with the switch off). PID 421 did not check the switches as they were supposed to; this behavior is obvious and can be modeled. To better model the error of participant 421, we examined their mouse clicks on the interface that were recorded using RUI (Kukreja et al.,

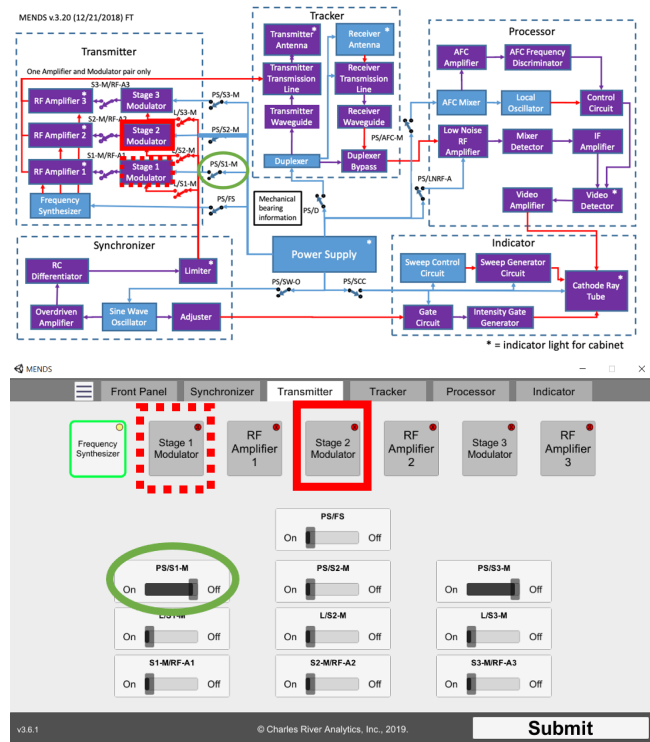


Figure 1: Upper: The BenFranklin Radar schematic.
Lower: The MENDS interface.

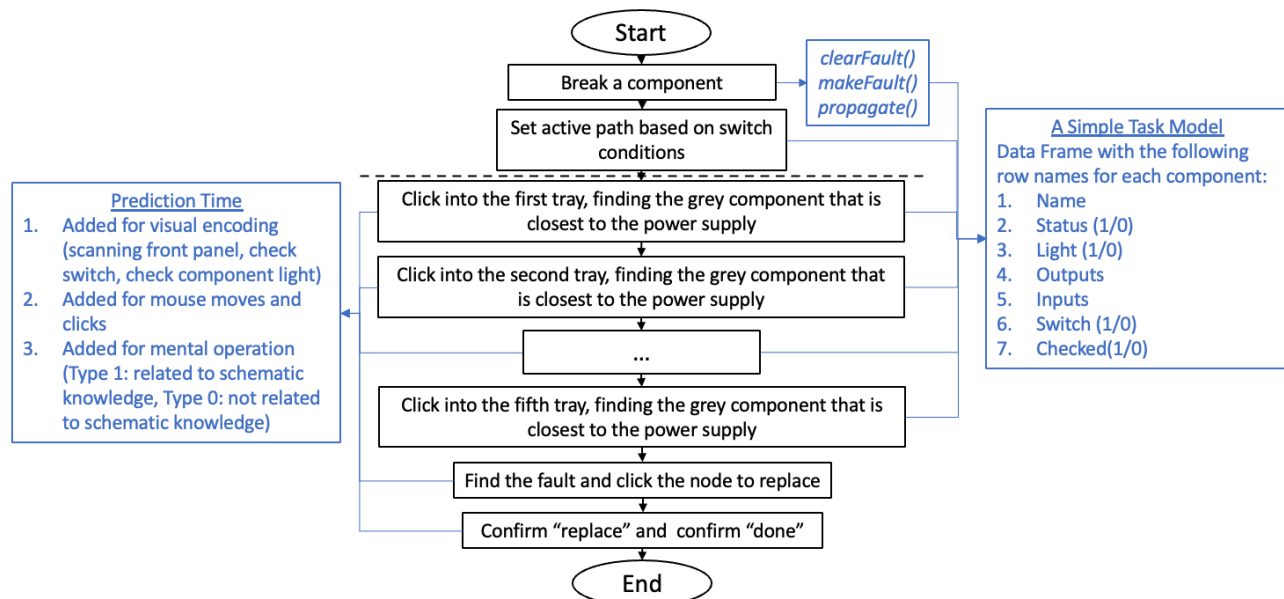


Figure 2: Flowchart of the All-Trays strategy.

2006). We edited our All-Trays strategy model to include error and error correction—ignoring switch information once. This would be better for models to explain where time goes, but those behaviors are not yet carefully explored.

Comparison

We compare the response times of the 20 tasks in the test session to the old All-Tray strategy model, the error model, and the participant 421's performance (Figure 3). The Error model is more of a "glove and hand" shape than the old model for observed time and it also takes more time due to additional steps. The error model may get better if we adjust the parameters. The error model also shows a better power law of learning and correlation with the participant data. The error model (0.65) has higher R^2 than the old model (0.60), showing that our error model indeed catches more of participants' behaviors.

reaching correction and success, from many perspectives and not just in this fault-finding task, it is an unavoidable journey that we all need to be friends with the errors we make and learn from them. Adding error generation and correction into a strategy model indeed increased correlation with human data, but led to a difference in total time.

The errors categorized and modeled are only those not corrected by participants themselves. More errors, as well as more types of errors, exist if we expand and explore the errors that are corrected by participants themselves during the fault-finding process. Further analysis of where time goes can be done with those expanded analysis of errors in the process. Understanding and showing participants' learning and improvements through the analysis is also another exciting further analysis.

Acknowledgement

Thanks to Sarah Ricupero and Mizzah Tocmo for providing feedback to the longer version of the paper.

References

- Kukreja, U., Stevenson, W. E., & Ritter, F. E. (2006). RUI: Recording user input from interfaces under Windows and Mac OS X. *Behavior Research Methods*, 38(4), 656–659. <https://doi.org/10.3758/BF03193898>
- Ritter, F. E., Workman, D., & Wang, S. (2022). Predicting learning in a troubleshooting task using a cognitive architecture-based task analysis. *International Conference on Cognitive Modeling*. 222-223.
- Wang, S. & Ritter, F. E. (2023). Modeling strategy with learning in a complex task. *International Conference on Cognitive Modeling*.

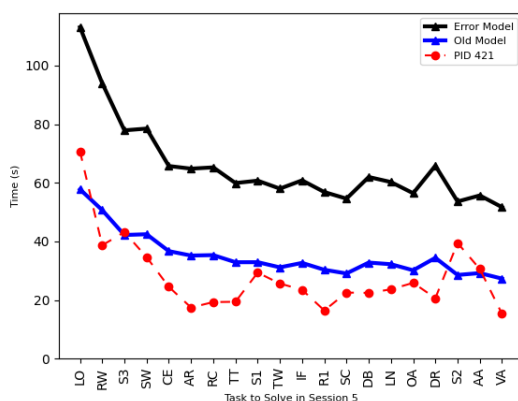


Figure 3: Comparison of response times.

Discussion and Conclusion

The patterns and trends of the errors tell us more about participants' learning and performance improvement. Before

Modeling a Strategy with Learning in a Complex Task

Shan N. Wang (sxw820@psu.edu)

Frank E. Ritter (fer2@psu.edu)

College of IST, Penn State
University Park, PA 16802 USA

Keywords: individual differences; cognitive modeling

Introduction

We used a complex electrical troubleshooting task shown in Figure 1 to study problem solving with learning, the BenFranklin Radar System that consists of 36 components, versus 7 components in the Klingon Laser Bank task used in several previous studies (Friedrich & Ritter, 2020; Ritter & Bibby, 2021). MENDS is a simulator created by Charles River Analytics for the BenFranklin Radar system, shown in the lower figure, used under license. The participants' task is to find the broken component in the circuit. In MENDS, participants can click and open the subsystem (trays) to see the components in each subsystem. They can decide and click the component that they think is the broken component, based on their schematic knowledge, the light and switch conditions. Components without power are in grey.

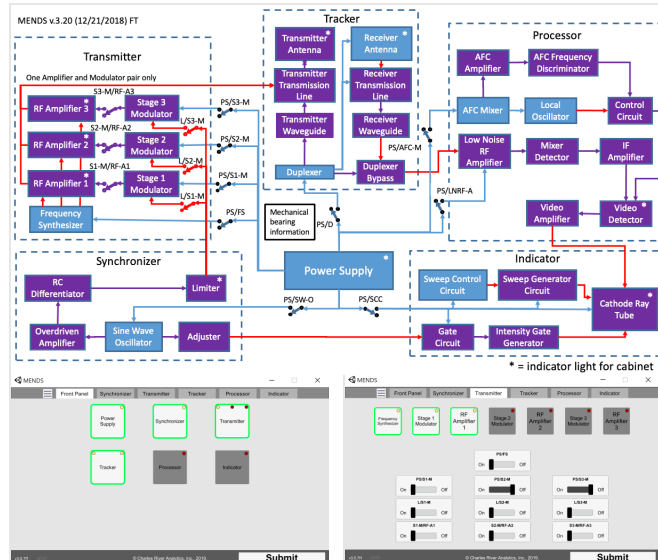


Figure 1: Upper: Schematic for the BenFranklin Radar. Blue lines are power; red lines are signal; purple lines are both. Lower: the MENDS simulator's front panel and one subsystem.

We collected participants' mouse moves and clicks for our modeling and data analysis using the RUI logger (Kukreja et al., 2006). To gather data, a user study was run (Ritter et al., 2022). The goal was to identify strategies and learning with a larger number of participants. After data cleaning, we had 111 participants' data in the test session, where they were asked to finish 20 problems. We collected the component and times the participants clicked on. From the mouse clicks of

the top 6 participants in the test session, we developed and implemented four strategies in the BenFranklin Radar task. Here we present one, the Grey Upstream strategy. The observed time for participants' performance was compared with the predicted time of our strategy models, without and with learning.

Modeling

Figure 2 shows how we built a simple task model for MENDS in Python, also described in (Ritter et al., 2022). The simple task model used a Panda data frame to store and reflect the components' broken status (1: component is fine; 0: broken), light status (1: component has light on; 0: light off), downstream components to give power, upstream components to receive power, switch condition before them (1: switch on; 2: switch off), the number of times the required schematic knowledge have applied by participants.

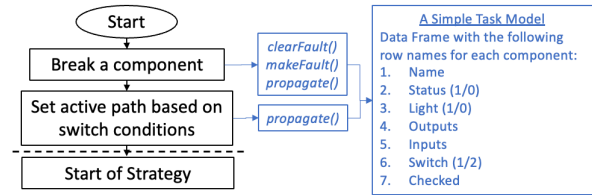


Figure 2. A flowchart for the simple task model and the grey-upstream strategy model. An active path refers to the components receiving power and that are supposed to have lights on.

The Grey Upstream Strategy

The Grey Upstream strategy is one of the four strategies that may be used to identify the broken component. Here we define the strategy in the scope of a task. Those strategies are categorized by four features: their starting point, how the front panel information was used, degree of schematic knowledge used, and degree of interface information used. Variations within one strategy are also possible. All strategies find the correct fault. The grey upstream strategy (GreyUp), shown in Figure 3, is based on participants P324, 420, 451, & 453. The strategy involves two major steps, finding a grey component as a starting point and tracing upstream in the schematic till finding the broken component. To locate the starting point, users click into the first grey tray using light information from the front panel and identify the first grey component by interface order from left to right and up to down within the clicked tray. Starting from the first grey component, participants use their schematic knowledge to trace up until they identify the broken component, which is

the only one with its light off but all its upstream components in the active path are with their lights on.

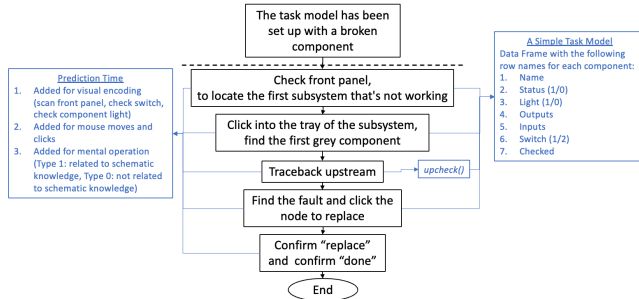


Figure 3. A flowchart of the Grey Upstream strategy, as a technical detail example.

The models take steps to solve the task based on each strategy, and time is added with each type of action. The time for each step depends on the learning level. Under the learning condition, we assume that knowledge is transferred to later tasks and the participants solve the later tasks faster based on the repetitive application of the same knowledge. Time parameters used in the models include visual coding, 0.4 (s); mouse move, 1.1; mouse click, 0.2; mental operation, 1.35; learning rate, 0.4. The learning follows ACT-R and the power law, and the base times are KLM times.

Models learn by modifying the mental operation time. We do this by having mental operation time as a function and not a constant. The function $mental(type, var2)$ gives different times for different types of mental operation and learning levels. The *type* variable includes *type 1* and *type 0*. *Type 1* refers to the mental operation time retrieving a component by using schematic knowledge. *Type 0* refers to any other mental processing not related to schematic knowledge. *Type 1*, retrieving a component, can be faster with learnings which indicated by *var2*. For *type 0*, *var2* is a random number because the time is assumed to be fixed, and no learning happens. For situations that assume no learning, the mental operation function in models uses a constant 1.35 s. When learning, the mental operation time is $1.35 \cdot n^{-l}$, where *n* is the number of times the component has been checked or the number of times the required circuit knowledge has being used. The value of *l* represents the learning rate, 0.4.

Comparison

We trained the models with the previous sessions that participants experienced before we ran the models for tasks in the test session. We compared the predicted time to the observed times of the participants. We consider that participants fit well to a strategy if they have R^2 with a p-value < .05. 14 participants' behaviors match a strategy ($p < .05$; R^2 varied) without learning in the test session. 69 participants' behaviors match a strategy with learning in the test session.

Figure 4 shows the match of PID 413 as an example. PID 413 has R^2 of .498, without learning and an R^2 of .518, with learning. The red dotted line is the observed time from human data; the blue solid line is the predicted time without learning;

the black solid line is predicted time with learning and was trained with previous sessions' faults.

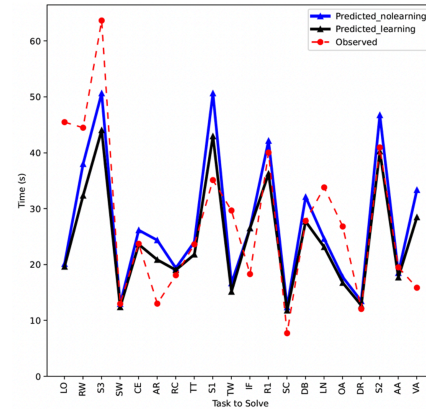


Figure 4: An example of PID 413 as one of the best matches. A comparison of human data, our Grey Upstream Strategy model with learning, and without learning.

Discussion and Conclusion

The strategy that includes learning indeed does better at performance prediction. Also, more strategy models could have been presented. Variations within strategies and strategy switch are not yet modeled. The current strategy models are from the top 6 well-performed participants, while the other participants made much more errors during the fault-finding process. We have not modeled errors, lapses, or changes of strategies within the same session. Modeling those can be our future steps. If our strategy models consider errors, the match between participants and strategies may increase.

Acknowledgements

Thanks to Sarah Ricupero, Mizzah Tocmo, and Rochelle Lorraine Clerkin for providing feedback to the longer version of the paper.

References

- Friedrich, M. B., & Ritter, F. E. (2020). Understanding strategy differences in a fault-finding task. *Cognitive Systems Research*, 59, 133–150.
- Kukreja, U., Stevenson, W. E., & Ritter, F. E. (2006). RUI: Recording user input from interfaces under Windows and Mac OS X. *Behavior Research Methods*, 38(4), 656–659.
- Ritter, F. E., & Bibby, P. (2021). Modeling how and when learning happens in a simple fault-finding task. *Proceedings of the 2001 Fourth International Conference on Cognitive Modeling*, 330–341.
- Ritter, F. E., Workman, D., & Wang, S. (2022). Predicting learning in a troubleshooting task using a cognitive architecture-based task analysis. *International Conference on Cognitive Modeling*, 222–223.

Cognitive and Meta-cognitive Signatures of Memory Retrieval Performance in Spoken Word Learning

Thomas Wilschut, (t.j.wilschut@rug.nl)¹ Florian Sense² Hedderik van Rijn¹

¹Department of Experimental Psychology, University of Groningen; ²Infinite Tactics, LLC

Keywords: Adaptive Learning, Cognitive Modeling, Memory Retrieval, Speech, Structural Equation Modelling, Prosody, Confidence

memory performance—extracted from speech in real time—may be used to effectively inform models of memory retrieval and improve adaptive learning systems.

Background

Cognitive models of memory retrieval aim to capture human learning and forgetting over time. Such models have been applied in learning systems that aid in memorizing information by adapting to the needs of individual learners (e.g., Lindsey, Shroyer, Pashler, & Mozer, 2014). Adaptive learning systems track learning performance to provide personalized feedback or optimize item repetition schedules. The effectiveness of such learning systems critically depends on their ability to use behavioral proxies to estimate the extent to which learners have successfully memorized the materials. The present study examines cognitive and meta-cognitive indicators of memory strength that are present in the learners' recorded speech signal while studying vocabulary items by vocally responding to cues.

In most model-based learning systems, predictions of memory retrieval rely on the accuracy and response latency of retrieval attempts. In this project, we will focus on spoken responses to visually-presented retrieval cues, which contain prosodic speech features (PSFs). PSFs are high-level properties of units of speech such as syllables, words or sentences, and include intonation (pitch variations), loudness and speaking speed. PSFs can carry information that is not conveyed by grammar or vocabulary, such as the emotional state of the speaker, emphasis, or the form of the utterance (e.g., question versus statement/command) (Xu, 2011). A recent study by Goupil and Aucouturier (2021) demonstrated that both the *objective accuracy* of a response, and a speaker's *meta-cognitive* confidence in the response are differentially reflected in speech. In their study, participants were instructed to complete a visual detection task, where they had to verbally choose which word they saw before from a number of alternatives and rate their confidence in the response. The results showed that *some* of the PSFs (speaking speed and pitch) were associated with the subjective confidence in a response, whereas the other PSF (loudness) was associated with objective accuracy.

In a recent study, Wilschut and colleagues (Wilschut, Sense, Scharenborg, & van Rijn, 2022), demonstrated that the above results generalize to a memory retrieval paradigm, and found that using PSFs on the current trial could increase the prediction accuracy for memory retrieval success on future learning trials. In the current project, we extend their work by further investigating the exact way in which PSFs are associated with both cognitive *and* meta-cognitive aspects of memory retrieval. Examining this question is important, as information about cognitive and meta-cognitive indices of

Methods

A total of 40 participants studied 30 Lithuanian-English vocabulary items. The first presentation of an item involved the visual presentation of both Lithuanian cue and the visual presentation of the English translation. Subsequent presentations of the item just showed the visual Lithuanian cue, and the participant was asked to utter the English translation. After this response was recorded, the participant was asked to rate their subjective confidence in the accuracy of the response using a slider-response scale, followed by corrective feedback. Participants cycled through the total list 4 times. At the start of a new cycle, the 30 items were split in to subsets of the first 15 and last 15 items, and both subsets were shuffled. Speech features were extracted from the recorded data after the experiment, and all speech features were standardized within participants.

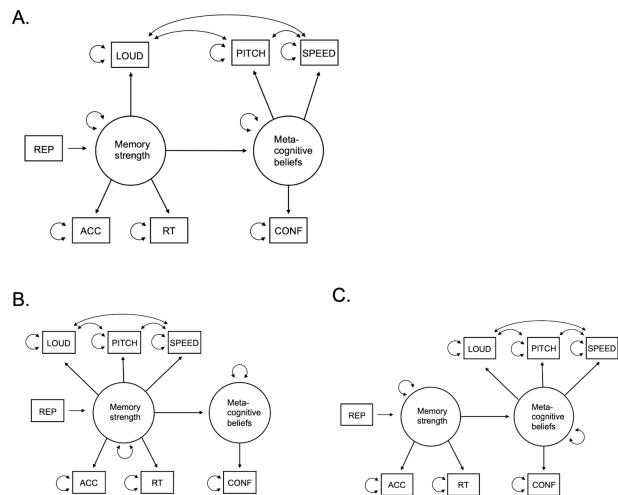


Figure 1: Structural equation models showing alternative possible relationships between latent factors memory strength and meta-cognitive beliefs as measured by accuracy, response times, confidence ratings, and PSFs. **A** shows the hypothesized model, **B** and **C** are alternative models.

To examine the relationship between PSFs and cognitive and meta-cognitive aspects of memory retrieval, we contrasted a hypothesized model to two alternative, competing models using structural equation modeling (Ullman & Bentler, 2012, (SEM)). All three models assume a relationship between latent variables memory strength and a learners' meta-cognitive beliefs about performance, with memory

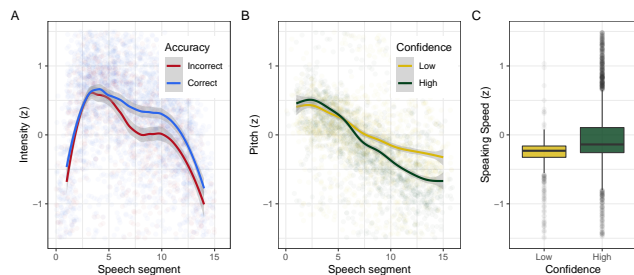


Figure 2: PSFs as a signature of memory retrieval performance. **A** shows that the average loudness over spoken retrieval attempts was higher for correct than for incorrect responses. **B** shows that average pitch slopes were higher for low versus high confidence responses, and **C** shows that speaking speed was lower for low than for high confidence responses. Shaded area's represent 95% confidence intervals.

strength measured by accuracy (ACC) and response times (RT), and meta-cognitive beliefs measured by subjective confidence judgements (CONF; see Figure 1). The hypothesized SEM is shown in Figure 1A, with two alternative models shown in Figure 1B and 1C. In the hypothesized model (A), in line with earlier research, meta-cognitive beliefs are measured by intonation and speaking speed, whereas memory strength is measured by loudness. The alternative models reflect two different underlying relationships: In the first alternative model (B), all PSFs directly reflect memory strength, and not meta-cognitive beliefs. In the second alternative model (C), all PSFs only indirectly measure memory strength, via a speaker's meta-cognitive beliefs about memory performance. We used Vuong's likelihood ratio test (Merkle, You, & Preacher, 2016) to compare the three models.

Results

The correctness of the responses was determined by Google's speech-to-text API, yielding sufficiently high transcription accuracy for the subsequent analyses. Figure 2A shows the average standardized loudness (intensity) over the duration of each utterance, for both correct and incorrect spoken retrieval attempts. Responses were, on average, louder for correct compared to incorrect responses. Figure 2B shows the average standardized pitch, separated for high subjective confidence scores (confidence scores above average for that participant), and low subjective confidence scores (below average for that participant). The figure shows a less negative averaged pitch slope for responses with low confidence than for responses with high confidence. Finally, Figure 2C shows that the average standardized speaking speed for high confidence retrieval attempts was higher than the average standardized speaking speed for low confidence retrieval attempts. These results underline and extend earlier findings, (Goupil & Aucouturier, 2021) and (Wilschut et al., 2022) by demonstrating that both cognitive (accuracy) and meta-cognitive (confidence) markers of memory performance are present in spoken word learning.

To compare the fit of the three SEM models outlined above, we used Vuong's likelihood ratio test. The hypothesized SEM model (A) fits the experimental data significantly better than both alternative models B and C ($z = 7.177$, $p < 0.001$; $z = 2.980$, $p = 0.001$, respectively). This supports the idea that meta-cognitive beliefs about memory retrieval are captured in different PSFs than the objective accuracy of a response.

Conclusion

This study examined which cognitive and meta-cognitive proxies of memory strength are present in the speech signal during spoken retrieval attempts. Participants studied vocabulary items using spoken retrieval practice. The results of the study are twofold. First, we demonstrate that it is possible to extract information about (1) the accuracy of a response and (2) a speaker's subjective confidence in a response from the speech signal. Second, we show that meta-cognitive beliefs about memory performance are measured mainly by variations in pitch and speaking speed, whereas the objective accuracy of a response is mainly measured by its loudness. The results of this study have theoretical and practical relevance. They contribute to a better understanding of the relationship between prosodic speech variations and (meta)memory processes and could facilitate the development of speech analyses as a new tool to explore open questions in learning research (e.g., about a learner's confidence in their responses). Second, as they demonstrate that the speech signal contains relevant information about memory retrieval performance, they may have important implications for the further development of models of memory retrieval used in adaptive learning systems. For example, extracting information about a speaker's confidence from the speech signal in real time may allow for improvement of predictions of future retrieval success—without the learner having to make explicit confidence judgments after each learning trial.

References

- Goupil, L., & Aucouturier, J.-J. (2021). Distinct signatures of subjective confidence and objective accuracy in speech prosody. *Cognition*, 212, 104661.
- Lindsey, R. V., Shroyer, J. D., Pashler, H., & Mozer, M. C. (2014). Improving students' long-term knowledge retention through personalized review. *Psychological science*, 25(3), 639–647.
- Merkle, E. C., You, D., & Preacher, K. J. (2016). Testing nonnested structural equation models. *Psychological Methods*, 21(2), 151.
- Ullman, J. B., & Bentler, P. M. (2012). Structural equation modeling. *Handbook of Psychology, Second Edition*, 2.
- Wilschut, T., Sense, F., Scharenborg, O., & van Rijn, H. (2022). Beyond responding fast or slow: Improving cognitive models of memory retrieval using prosodic speech features. *Paper presented at In-Person MathPsych/ICCM 2022, Via mathpsych.org/presentation/858*.
- Xu, Y. (2011). Speech prosody: A methodological review. *Journal of Speech Sciences*, 1(1), 85–115.

Long Road Ahead: Lessons Learned from the (Soon to Be) Longest Running Cognitive Model

Siyu Wu (sfw5621@psu.edu)¹, Amir Bagherzadeh (abb6024@psu.edu)²,
Frank E. Ritter (frank.ritter@psu.edu)¹, and Farnaz Tehranchi (farnaz.tehranchi@psu.edu)³

¹ College of Information Sciences and Technology

² Department of Industrial and Manufacturing Engineering

³ School of Engineering Design and Innovation
Penn State, University Park, PA, 16802, USA

Abstract

We present a cognitive model that plays a video game of driving a bus for a long time. The model was built using the ACT-R cognitive architecture and an extension to support perceptual-motor knowledge of how to interact with the environment (VisiTor and ACT-R). Our extension includes bitmap-level eyes and robot hands. We ran the model for a long time, over 4 hours on the way from Tucson to Las Vegas. We employed a design approach based on the ADDIE model to create different knowledge representations and actions; the model's predictions can be matched to some aspects of human behavior on the fine details regarding the number of course corrections and average speed and learning rate. However, it does not exhibit the same level of fatigue as human behavior. This contrasts with the way humans typically perform such long tasks. This model shows that (a) perception opens up new interfaces and provides a very accessible testbed for examining further aspects of behavior and (b) adding components of human behavior that remain missing from ACT-R can now be included.

Keywords: Cognition Computational Modeling; ACT-R; Driver Model

Introduction

Cognitive architectures can be used to develop cognitive models of various psychological phenomena and tasks (Newell, 1990). In addition, cognitive architectures afford procedures and structures that align with human behavior, such as reaction times, error rates, and fMRI results (Anderson, 2007; Laird, 2019).

ACT-R is one kind of cognitive architecture realized as software, through which we can construct models that can store, retrieve, and process knowledge, as well as explain and predict performance (Anderson, 1996; Bothell, 2017). Modelers and researchers have used ACT-R to create a variety of models, from models that only contain cognition-level activities (e.g., Tower of Hanoi) to models that contain comprehensive perception and motor behaviors (e.g., Fleetwood & Byrne, 2002; Tehranchi & Ritter, 2018).

In particular, ways to implement perceptual and motor behaviors can be classified into several categories based on

how directly they interact with a task (Ritter et al., 2000). Perhaps the most commonly seen approach is models that interact with modified interfaces (e.g., Anderson & Douglass, 2001; Byrne et al., 2010).

Another important approach is to have models interact with unmodified tasks that users can see, using simulated eyes and hands (e.g., Bagherzade & Tehranchi, 2022; Ritter et al., 2006). Previous studies used ACT-R to build cognitive models and compare the model behavior with human behavior, finding that ACT-R's model behavior at the cognition level is more consistent with human behavior than at the perceptual motor level (e.g., Ritter et al., 2006; Schwartz, Tehranchi, & Ritter, 2020). However, extending ACT-R to model further aspects of vision and motor behavior on uninstrumented interfaces can be an important future direction (Ritter et al., 2019; Laird, 2019; Pew & Mavor, 2007).

For example, Schwartz et al., (2020) built a model using ACT-R in conjunction with an extended vision and motor management tool (JSegMan) to play Penn and Teller's *Desert Bus* video game (e.g., <https://desertbus.org/>; Parkin, 2013). However, the model's behavior shows discrepancies with human behavior, and one of them is that the model can only run the bus for less than 20 minutes, which is a much shorter time than a human can. Table 1 lists some of the discrepancies between their model and human behavior.

Table 1. Limitations of the Schwartz et al. (2020) model.

1.	Did not start the simulation
2.	Did not drive for more than 20 min.
3.	Did not make the first turn around
4.	Did not make the second turn around (which might be different, after 16 hours of driving)

Our research expands upon the work of Schwartz et al. (2020) by using the previous ACT-R model and revising

its perceptual motor components to enable real-time control of a driving task. Our study presents a model that possesses two relatively novel capabilities for cognitive models: it can perform long-term tasks lasting up to 4 hours, and it can do so while interacting with an interface that was not specifically designed for models. Ultimately, the models will be able to play the video game *Desert Bus* for a much longer period of time, essentially, indefinitely.

Using the ADDIE (Morrison et al., 2010) framework as our design approach, we improved the knowledge representation and actions of the model. We also added a new function for the extended hand. Although the model successfully completed the task longer, its behavior on this task revealed limitations in the ACT-R model, which we identify and attempt to address.

Components and Theoretical Foundations

We now explain our architecture and the perceptual interface to interact with the interface. We then describe the simulation that the model interacts with.

The Architecture of Cognition

ACT-R is a cognitive architecture and a theory of simulating and understanding human cognition (Anderson, 2007; Ritter, Tehranchi, & Oury, 2019). Its theory is embodied in the ACT-R software, through which we can construct models that can store, retrieve, and process knowledge, as well as explain and predict performance (Bothell, 2017).

There are currently two kinds of knowledge representations in ACT-R, declarative and procedural knowledge. Declarative knowledge consists of chunks of memory (e.g., apple is a kind of fruit), while procedural knowledge performs basic operations, moves data among buffers, and identifies the next instructions to be executed (e.g., to submit your answer, you have to click the submit bottom). When the model is driving a bus in a first-person perspective, these pieces of information will contain information such as what visual items presented to look at and what tasks to do next.

ACT-R is not complete, like all models. In this work we extend it to include new types of interaction knowledge and the capability to interact with all tasks that have a computer interface that is represented with a screen and that can be interacted with a keyboard and a mouse.

The Architecture of Interaction

Models interact with the world through their visual and motor systems. The interaction includes processing visual items presented (visual systems), pressing keys, and moving and clicking the mouse (motor systems).

Specifically, the visual system holds chunks of information about an object's location in the "where" buffer and chunks of information about objects in the visual scene in the "what" buffer. A central production system can reason about and lead to behavior based on these chunks. For

example, the driving model may move forward or steer based on the position data retrieved from the visual buffer (Ritter et al., 2019).

Models can interact with the simulation, but the approach we will use is to use the screen's bitmap directly to find objects. Motor output can be put on the USB bus and appear as if a user at the keyboard typed characters or moved the mouse. In Table 2, we list previous models' history of interaction using this approach.

Table 2: Previous models history of interaction.

Name of model	Interaction tool	Reference
Eyes and Hands	ESegman	(Tehranchi & Ritter, 2017)
Biased coin	JSegman	(Tehranchi & Ritter, 2020)
Spreadsheet	JSegman	(Tehranchi, 2020)
Desert Bus 1	JSegman	(Schwartz et al., 2020)
Heads and Tails	VisiTor	(Bagherzadeh & Tehranchi, 2022)
Desert Bus 2	VisiTor	(this paper)

VisiTor (Bagherzadeh & Tehranchi, 2022) is a Python software package stored on a public GitHub that has been developed to provide simulated hands and eyes. It is comprised of two types of functions—motor and visual. The visual functions include "whatIsOnScreen", which checks if certain visual patterns are present in the environment, "whereIs", which locates a pattern within a defined module, and "getMouseLocation", which retrieves the mouse's location. The motor functions consist of "click", which imitates a single mouse click, "Keypress," which replicates the pressing a key, "moveCursorTo", which emulates mouse movement to a specific screen location, and "moveCursorToPattern", which replicates mouse movement to a specific visual pattern.

The Simulation

Penn and Teller created the video game *Desert Bus* with the intention of making a statement about video games. The game is deliberately monotonous and lengthy, with the player driving a bus in real-time at a maximum speed of 45 mph from Tucson, AZ to Las Vegas, NV. Each leg takes at least eight hours, and the bus continuously drifts to the right. If the player swerves off the road, the engine will stall, and they will need to start over from Tucson. The game has no virtual passengers or other cars on the road. Once the player completes the eight-hour journey, the screen fades to black, and they return to the starting point to play again indefinitely. At night, the road is dark. Figure 1 provides a screenshot of the game available through Steam (there are other versions available now).



Figure 1: A Screenshot of the driver's view at approximately 10 min. into the game, oversteered.

This game offers the player a first-person view as they carry out tasks, and the surroundings change dynamically based on their actions. The specific edition that we use was created by Dinosaur Games and released by Gearbox Software, based on the unreleased "Smoke and Mirrors" Sega CD game. The game's driving environment, Desert Bus, was obtained from Steam (https://store.steampowered.com/app/638110/Desert_Bus_VR/) and can be downloaded for free on Windows machines. There were no alterations made to the game to support the model.

Desert Bus Model

To extend the amount of time the models could drive the bus, we created a more sophisticated model than Schwartz et al. (2020). We also explain the extensions to ACT-R's perceptual-motor system (VisiTor) to support this model, and then explain the details of the model.

Extending ACT-R 7 With VisiTor

This model is built in the latest version of ACT-R, ACT-R 7. It includes a Perceptual-Motor module (Bothell, 2011) that provides models with direct access to interfaces built in Macintosh Common Lisp (MCL). Therefore, by modifying modules, researchers can refine ACT-R 7 models to produce more complex behavior with neurologically compatible mechanisms. However, the current ACT-R PM module enabled models to interact only with MCL interfaces built with a window type provided with ACT-R, which limits their ability to interact with interfaces not created in that tool, such as *Desert Bus*.

To allow ACT-R 7 to access uninstrumented interfaces, a potential solution is to use VisiTor (Bagherzadeh & Tehranchi, 2022). It can simulate the user's visual attention (vision) as well as their use of a mouse and keyboard (motor). VisiTor functions as a vision manager tool that receives motor commands from the ACT-R PM module and sends them to the environment through an Emacs/slime link. By using this tool, ACT-R can engage with any environment while maintaining operations that are as similar as

possible to those of the user. Additionally, VisiTor's capabilities can be expanded by incorporating modules into the tool.

ACT-R instructs VisiTor to scan the screen for particular pixel patterns that activate a production rule to initiate the program. Once VisiTor detects the start pattern, it sends a signal to ACT-R to begin running. Subsequently, ACT-R activates an "if-then" production rule that directs VisiTor to hold down the "W" key which starts the bus and accelerates when the start pattern is located in the visual environment. ACT-R then requests VisiTor to use the simulated hands to maintain pressure on the "W" key, effectively holding the bus's accelerator down. When VisiTor observes that the "right border of the road" objects deviate more than 200 pixels from the center of the road (approx. 5 degrees for someone 1.5 feet from the display), it signals to execute the steering production rule that specifies to hold down the "W" and "A" key until the car returns to the center of the road when the right edge of the road appears in the designated environment.

To undertake this task, VisiTor required a few minor extensions. It needed to simplify the process of describing visual objects and incorporate a range of visual objects. Furthermore, it had to transfer motor commands with a variable duration to maintain a keystroke. To support the novel task of driving a desert bus indefinitely, we implemented the "longpresskey" feature to VisiTor. This functionality enables the simulation of key-pressing actions, with the option of defining a duration of time to hold the key. (There are numerous other ways to implement this motor output, and we are also exploring those.)

The Driving Task

The tasks in Drive the Bus can be seen as occurring over three sections. (a) The player starts the game and starts driving the bus. (b) The player drives the bus from Tucson to Las Vegas. (c) After arriving in Las Vegas, the bus appears again at the end of an eight-hour stretch of road and starts again in an endless way.

This study reports the work of having the model do task (b), drive the bus from Tucson to Las Vegas. Tasks (a), and (c) will be reported later.

Driver model

Our objective was to redesign the model to make it do the long hours of driving. We employed the ADDIE (Morrison et al., 2010) framework for developing tutors (which we are familiar with) to create the model. ADDIE is a popular instructional design framework that can be adapted to create cognitive computational models. The ADDIE model consists of five stages: analysis, design, development, implementation, and evaluation.

In our analysis phase, the modeler gathers information about the target simulation environment, task objectives, and constraints. Based on the analysis, the modeler creates

a plan for the model in the design phase, which includes the overall structure, content, and development strategies.

The development phase involves creating and refining the knowledge components and extended eyes and hands functions, such as declarative memories, production rules, visual patterns, and functions that will be utilized in VisiTor. Once the components analyzed are complete, the implementation phase involves delivering the model to the intended simulation environment.

Finally, in the evaluation phase, the modeler collects feedback and data to assess the effectiveness of the model and make improvements as needed. Using the ADDIE framework can help ensure that the computational cognition model is designed with the task in mind and are effective in achieving the desired simulation outcomes. It also encourages more intermediate products and buy-in from stakeholders and reflection, similar to the risk-driven spiral model (Pew & Mavor, 2007)

We consider two important pieces in the models' design. The first is how to represent the necessary knowledge for the model to be able to perform the task, and the other is the steps the model will perform to complete the task.

To start the model creation process, the task and simulation environment is analyzed by examining the game interface, 2 human subjects' keystrokes, and interkey intervals, as well as the visual cues and triggered actions. A list of declarative knowledge chunks representing visual cues and keypresses is formulated, such as "push the key to move forward," and a set of production rules representing the sequential actions that are triggered, such as "if the deviation of the bus exceeds 200 pixels, steer left". For example, we had two research assistants one Saturday afternoon literally drive the bus from Tucson to Las Vegas and attempt to record their behavior.

Additionally, supportive functions in VisiTor are developed by including a long-key-press operator. Then, the model is tested in the simulation, and necessary implementations are made, such as redefining visual cues.

Finally, the ACT-R output data are analyzed, and the model's performance is evaluated. The information collected from this evaluation is used to guide the iterative development of the model.

Below we describe the model we have written for this task. The following is a detailed explanation of how the model control loop is built, as well as the model's knowledge representations, actions to perform, and capabilities and functions used via the interaction architecture and VisiTor.

Control Loop

Figure 2 shows a flowchart of the mechanism underlying the model's control loop. It uses the visual buffer and simulated eyes to attend to and harvest the two visual objects, and then use the "whereis" function of VisiTor to encode

the screen-x locations of the two objects. The model will then subtract the value of screen-x, and decide to steer left if the deviation is over 200 pixels.

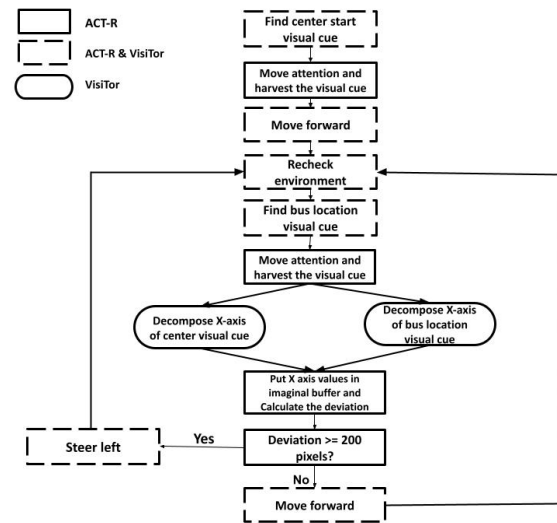


Figure 2: The control loop of the model.

Knowledge Representation

The model has two types of chunks, and a total of 12 declarative memories, which are working memories that tells the model to make the action based on the visual cues it saw. The first chunk is named "drive" and has two slots, "strategy" and "state", with state having parameters as object items. Another chunk type is "encoding", which has slots for the screen-x locations of the two visual cues and a deviation slot.

Actions to Perform

This model uses an explicit goal state to control the model. It contains 13 production rules. Table 3 list the high-level descriptions of the steps the model performs and the corresponding production rules.

Table 3 indicates that the model begins by investigating the simulation environment to locate and collect the visual cue necessary to initiate gameplay. It then utilizes the manual buffer to maintain forward motion by holding down the key. While doing so, the model continuously evaluates the environment to identify and gather visual cues related to the bus's location. It then breaks down the x-axis of the center line and bus location and computes the difference between the two values. If this difference is above 200 pixels, the model will hold down the key to turn the bus left; otherwise, it will just continue moving forward.

Table 3: High level description of the steps and the production rules that have been used in the model X.

High level descriptions of steps	Corresponding production rules
1. When it detects a start visual cue, attend it, and press the “W” key using the manual buffer	Go PerceiveEnvironment Move-attention Ahead
2. Clear the visual buffer and attend to the bus location	Recheck-environment Danger Finding-danger Move-attention-danger
3. Calculate the bus deviation from the center lane	Where-is-danger Where-is-center Calculate-deviation
4. Use the manual buffer by pressing “w” if the deviation is less than 200 pixels	Consider-ahead
5. Clear the manual buffer if the deviation > 200 pixels. Using the manual buffer, align the bus by pressing the key for 6 seconds.	Consider-steer
6. After that, clear the visual buffer, Repeat steps 1,2,3,4,5	Loop back to perceive-environment

Demonstration Observations

The experiment involved running a model to assess its performance and collect ACT-R output data. The model was found to be capable of running for hours in the long term. The declarative memories and production rules that were developed proved to be successful in meeting the needs of the simulation. Additionally, a new feature was incorporated into VisiTor, enabling the bus to accelerate with “w” held down.

However, if the driving speed exceeded the game setting, the ACT-R perceptual motor module in conjunction with simulated eyes and hands may not be able to identify, harvest, and process the location deviation as quickly as required.

As seen in Table 4, The ACT-R output data revealed that the model had an average time of 0.235 s to find a visual cue using the Visicon in conjunction with simulated eyes, an average time of 0.05 s to move attention to the visual cue, and an average time of 0.2 s to decompose the visual cue location and place it into the imaginal buffer. The total decision-making time of the model in gauging the deviation and making the next action decision of punching the keys was 0.9 s. This reflection time would be efficient for the model to identify danger cues and steer back to the road

when the bus was driving at a maximum speed of 45 mph. One the order of minutes, this behavior can be compared to human behavior. On the order of hours we will see the model will outperform humans. This allows the model to accomplish a task in driving the bus that surpasses human capability, as it does not experience fatigue or mistakes (Gunzelmann, Moore, Salvucci, & Gluck, 2011).

Table 4: The output script of the ACT-R model that shows buffers, fired productions, and VisiTor commands with time stamps.

```
CL--USER> (run 10)
0.000 GOAL SET-BUFFER-CHUNK GOAL GOER NIL
0.000 VISION SET-BUFFER-CHUNK VISUAL-LOCATION CHUNK0 NIL
0.050 PROCEDURAL PRODUCTION-FIRED GO
Ready to go
0.100 PROCEDURAL PRODUCTION-FIRED PERCEIVE-ENVIRONMENT
0.150 PROCEDURAL PRODUCTION-FIRED MOVE-ATTENTION
0.150 VISION SET-BUFFER-CHUNK VISUAL-LOCATION CHUNK0
0.200 PROCEDURAL PRODUCTION-FIRED AHEAD
hihi
continuouspress a key!
0.200 MOTOR PUNCH HAND RIGHT FINGER INDEX
0.235 VISION SET-BUFFER-CHUNK VISUAL CHUNK2
0.250 PROCEDURAL PRODUCTION-FIRED RECHECK-ENVIRONMENT
0.285 VISION SET-BUFFER-CHUNK VISUAL CHUNK3
0.300 PROCEDURAL PRODUCTION-FIRED DANGER
0.350 PROCEDURAL PRODUCTION-FIRED FINDING-DANGER
0.350 VISION SET-BUFFER-CHUNK VISUAL-LOCATION CHUNK1
0.485 VISION SET-BUFFER-CHUNK VISUAL CHUNK4
0.535 PROCEDURAL PRODUCTION-FIRED MOVE-ATTENTION-DANGER
0.585 PROCEDURAL PRODUCTION-FIRED WHEREISDANGER
0.620 VISION SET-BUFFER-CHUNK VISUAL CHUNKS
0.185 IMAGINAL SET-BUFFER-CHUNK-FROM-SPEC IMAGINAL
0.835 PROCEDURAL PRODUCTION-FIRED WHEREISCENTER
1.035 IMAGINAL SET-BUFFER-CHUNK-FROM-SPEC IMAGINAL
1.085 PROCEDURAL PRODUCTION-FIRED CALCULATE-DEVIATION
1.135 PROCEDURAL PRODUCTION-FIRED CONSIDER-STEER
```

In comparison to the driving the bus model created by Schwartz et al. (2020), there was a significant improvement in the accuracy of identifying visual cues in our model, as well as the ability to drive for a longer period of time. This model's better performance can be attributed to the following reasons.

To begin with, the ADDIE framework is a suitable choice for building the model for the current task because it helps in creating declarative memories, production rules, and control loop mechanism that closely resemble human driving behavior. It is important to distinguish between human driving behavior and human behavior. Human driving behavior pertains to how drivers use visual cues, such as the center line and bus location, to gauge deviation and make steering decisions. Human behavior is also determined by other psychophysiological factors, such as fatigue and decreasing correction rate, which are not included in this model yet and will be discussed separately. In this model, a superior control mechanism was implemented that replicates human driving behavior and allows for better long-term bus driving performance.

Furthermore, the integration of VisiTor into ACT-R 7 leads to enhanced coordination between perceptual and motor behavior. The entire process of ACT-R sending a request to prompt VisiTor to search the screen for the bus location visual cue, extract the location of the bus and center line, calculate the deviation, and decide on the next action can be completed in just 0.9 s. This time represents

a significant improvement compared to the previous model, where JSegman was used in combination with ACT-R 6. According to Schwartz et al. (2020), the average time required to just match the visual template was already 6.01 s. Additionally, VisiTor's extensibility allows for the creation of new functions that support the specific requirements of the task, thereby considerably enhancing the model's performance. The long key press function, which is incorporated in this model, effectively enables the model to complete long-term tasks successfully by preventing ACT-R key presses from being interpreted as immediate press and release actions.

Nevertheless, a key factor that affects the model's performance is its limited ability to simulate activities in a dynamically changing environment. During gameplay, the environment undergoes dynamic changes, and at the near four-hour mark, the game environment shifts from a daytime mode to a nighttime mode, accompanied by a complete alteration in the visual pattern of the visual display. Although the PM module with simulated eyes can adapt to minor environmental changes, such as changes in road position or decorations along the roadside, the Visicon and VisiTor are not yet equipped to recognize an entirely different environment.

Discussion and Conclusion

The aim of this study was to employ ACT-R 7.X and its architecture of interaction to successfully complete a demanding cognitive modeling task of driving. The average run time for the model was one hour, with the longest run time lasting four hours until the gaming environment transitioned into night mode.

Instead of altering the game environment to accommodate ACT-R's MCL interfaces, we utilized the perceptual motor module of ACT-R 7 along with the vision and motor management software VisiTor to enable the model to play on the uninstrumented game interface.

We captured the model's gameplay and examined the ACT-R output, which demonstrated that the coordination

between motor and vision using ACT-R 7 and VisiTor was highly effective, taking less than one second to steer the bus back into the safe range of the road. This was only feasible if the car's speed was not more than 45 mph (The speed limit in this game is 45 mph). We anticipate that if the bus speed is higher, then a shorter transport signal will be necessary for communication between ACT-R and VisiTor.

We have found that the superior human behavior model has implied limitations in the ACT-R perceptual-motor modules.

There are also limitations in the central modules (production rules). These limitations include the lack of consideration for physiological factors such as fatigue or decreasing correction rates over time. In a study by Schwartz et al. (2020), it was suggested that incorporating physiology with ACT-R could make the model more realistic. We agree with this point and plan to add that in our future work. This new approach can help with testing the compound effects of fatigue and learning rate on our model.

ACT-R + VisiTor playing *Drive the Bus* provides an excellent platform for studying the interaction of vision, attention, errors, and fatigue. It is a more naturalistic task than the PsychoMotor Vigilance task (PVT, Dinges & Powell, 1985). We can now explore an existing fatigue model (Gunzelmann, Gross, Gluck, & Dinges, 2009), and examine fatigued driving (e.g., Gunzelmann, Moore, Salvucci, & Gluck, 2011), visual attention, the need for micuration, and modeling the details of interaction.

We could gain understanding about how long-term and repetitive physical activities, like driving a bus for an extended period, affect human performance. It remains to be seen if this task is more like the PVT or like motor control (Bolkhovsky, Ritter, Chon, Qin, 2018). This task would also allow us to determine whether psychological factors could potentially harm or the increasing of learning rate due to the practice would enhance driving skills. We could also introduce additional variables, such as caffeine, to examine their combined impact.

References

- Anderson, J. R. (1996). ACT: A simple theory of complex cognition. *American Psychologist*, 51(4), 355.
- Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York, NY: Oxford University Press.
- Anderson, J. R., & Douglass, S. (2001). Tower of Hanoi: Evidence for the cost of goal retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 27, 1331–1346. <https://doi.org/10.1037/0278-7393.27.6.1331>
- Bagherzadeh, A., & Tehranchi, F. (2022). Comparing cognitive, cognitive instance-based, and reinforcement learning models in an interactive task. *Proceedings of ICCM, The 20th International Conference on Cognitive Modeling*. 1-7.
- Bolkhovskiy, J. B., Ritter, F. E., Chon, K. H., & Qin, M. (2018). Performance trends during sleep deprivation. *Aerospace Medicine and Human Performance*, 89(7), 626-633(8).
- Bothell, D. (2017). ACT-R 7 reference manual. Available at actr.psy.cmu.edu/wordpress/wpcontent/themes/ACT-R/actr7/reference-manual.pdf, Accessed February 2023.
- Byrne, M. D., O'Malley, M. K., Gallagher, M. A., Purkayastha, S. N., Howie, N., & Huegel, J. C. (2010). A preliminary ACT-R model of a continuous motor task. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 54(13), 1037–1041. <https://doi.org/10.1177/154193121005401308>
- Dinges, D. I., & Powell, J. W. (1985). Microcomputer analysis of performance on a portable, simple visual RT task sustained operations. *Behavior Research Methods, Instrumentation, and Computers*, 17, 652–655.
- Fleetwood, M. D., & Byrne, M. D. (2002). Modeling icon search in ACT-R/PM. *Cognitive Systems Research*, 3(1), 25–33. [https://doi.org/10.1016/S1389-0417\(01\)00041-9](https://doi.org/10.1016/S1389-0417(01)00041-9)
- Gunzelmann, G., Gross, J. B., Gluck, K. A., & Dinges, D. F. (2009). Sleep deprivation and sustained attention performance: Integrating mathematical and cognitive modeling. *Cognitive Science*, 33(5), 880-910.
- Gunzelmann, G., Moore, L. R., Salvucci, D. D., & Gluck, K. A. (2011). Sleep loss and driver performance: Quantitative predictions with zero free parameters. *Cognitive Systems Research*, 12, 154-163.
- Laird, J. E. (2019). *The Soar cognitive architecture*. Cambridge, MA, MIT Press.
- Morrison, G. R., Ross, S. M., Kahlman, H. K., & Kemp, J. E. (2010). *Designing effective instruction* (6th Edition ed.). Hoboken, NJ: John Wiley & Sons.
- Pew, R. W., & Mavor, A. S. (Eds.). (2007). *Human-system integration in the system development process: A new look*. Washington, DC: National Academy Press. books.nap.edu/catalog/11893, checked May 2019.
- Newell, A. (1990). *Unified Theories of Cognition*. Harvard University Press.
- Ritter, F. E., Baxter, G. D., Jones, G., & Young, R. M. (2000). Supporting cognitive models as users. *ACM Transactions on Computer-Human Interaction*, 7(2), 141–173. <https://doi.org/10.1145/353485.353486>
- Ritter, F. E., Tehranchi, F., & Oury, J. D. (2019). ACT-R: A cognitive architecture for modeling cognition. *WIREs Cognitive Science*, 10(3), e1488. <https://doi.org/10.1002/wcs.1488>
- Ritter, F. E., Van Rooy, D., Amant, R. St., & Simpson, K. (2006). Providing user models direct access to interfaces: An exploratory study of a simple interface with implications for HRI and HCI. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 36(3), 592–601. <https://doi.org/10.1109/TSMCA.2005.853482>
- Schwartz, D. M., Tehranchi, F., & Ritter, F. E. (2020). Drive the bus: Extending JSegMan to drive a virtual long-range bus. In *Proceedings of the 18th International Conference on Cognitive Modeling (ICCM 2020)*. 241-246.
- Tehranchi, F., & Ritter, F. E. (2017). An eyes and hands model for cognitive architectures to interact with user interfaces. In *MAICS* (pp. 15-20).
- Tehranchi, F., & Ritter, F. E. (2018). Modeling visual search in interactive graphic interfaces: Adding visual pattern matching algorithms to ACT-R. In *Proceedings of ICCM-2018-16th International Conference on Cognitive Modeling* (pp. 162-167)
- Tehranchi, F. (2020). An Eyes and Hands Model: Extending Visual and Motor Modules for Cognitive Architectures (Doctoral dissertation, The Pennsylvania State University).
- Wilson, D., & Sicart, M. (2010). Now it's personal: On abusive game design. *Proceedings of the International Academic Conference on the Future of Game Design and Technology*, 40–47. <https://doi.org/10.1145/1920778.1920785>

The Cognitive Substrates of Model-Based Learning: An Integrative Declarative-Procedural Model

Yuxue C. Yang (chery@uw.edu)

Department of Psychology, University of Washington, Seattle, WA 98105, USA

Andrea Stocco (stocco@uw.edu)

Department of Psychology, University of Washington, Seattle, WA 98105, USA

Abstract

Understanding the fundamental cognitive process of decision-making is crucial for developing appropriate cognitive models. Two main planning-based approaches have been used to investigate learning in complex decision-making tasks: one using model-based (MB) reinforcement learning, an extension of reinforcement learning that includes high-level planning, and the other using instance-based learning (IBL), based on episodic memories of previous interactions. In this paper, we attempt to reconcile the two approaches by using ACT-R to implement a cognitively plausible substrate for the planning component of MB-RL. We review the MF and MB learning approaches in reinforcement learning and discuss their roles in decision-making strategies. Within the ACT-R framework, we propose a promising model that incorporates memory retrieval in MB planning, offering a cognitively plausible approach to the planning component of MB-RL. Our combined model successfully replicates well-known findings in the literature, including developmental reliance on memory and response time variations between common and rare options. Finally, our model naturally accounts for the balance of memory and RL depending on the relative cost of each. We argue for the superiority of our cognitive model and address the significance of this study for understanding the brain and computational processes underpinning decision-making strategies, as well as for applications in artificial intelligence and decision-making modeling.

Keywords: Decision-making, Reinforcement Learning, Model-Based Learning, Instance-Based Learning, Cognitive architecture

Introduction

Decision making is a fundamental ability of human cognition. Extensive research has been conducted on the mechanisms of experiential decision making in humans and animals. The predominant view is that, under simple circumstances, decisions are well characterized by *model-free* (MF) reinforcement learning (RL). In MF RL, the decision maker is an agent, and the available options are actions that the agent can apply to an environment. The agent typically uses temporal difference (TD) methods, such as Q-learning, to improve the estimates of future rewards associated with each action.

The MF paradigm has been extremely successful at explaining both behavioral and neural data in animal and human experiments (Niv, 2009). Most decisions, however, are not made within the simplified boundaries of laboratory experiments. This is particularly true in the case of humans, who interact with complex, non-stationary environments.

To deal with more complex situations, researchers have borrowed the concept of *model-based* RL (MBRL), an extension of RL that includes additional memory structures to explicitly store changes in the environment following an

action from the agent. The MB approach involves the construction and use of an internal representation of the environment, which allows for flexible and goal-directed decision-making. MBRL is a heterogeneous collection of methods, some of which include explicit replay of previous experiences (Sutton, 1991) while others are purely planning algorithms (Glascher et al, 2008).

Cognitive Substrates of MB

Despite much research, it is still unclear what cognitive processes underlie MB learning; some authors refer to it as "planning," while others link it to memory. Additionally, there is evidence that the MB and MF strategies are frequently combined, but there are no established standards for figuring out the best combination of these two approaches, especially in the context of cognitive literature.

Doll et al. (2015) provide critical insights into the interplay between MB and MF reinforcement learning approaches in decision-making strategies. They pointed out that the brain's multiple memory systems, specifically the declarative and procedural memory systems, serve as crucial substrates for distinct decision systems. Declarative memory, which involves conscious recollection of facts and events, is associated with MB learning as it enables the construction of mental models of the environment and the planning of actions based on simulating potential action outcomes. In contrast, procedural memory, which entails learning habits and skills, is linked to MF learning, where decisions are guided by learned associations between actions and outcomes. Doll et al.'s findings further provide insights into how these two memory systems are dependent on each other in learning decision-making strategies, highlighting the complex and adaptive nature of human cognition.

Instance-Based Learning

When dealing with decisions in complex tasks, a radical alternative to MB is the hypothesis that humans only rely on memories of previous interactions. Perhaps the most promising part of this framework is the Instance-Based Learning (IBL) theory. Gonzalez, Lerch, and Lebiere (2003) pioneered this line of inquiry with their IBL framework, which integrates elements of both MB and MF learning approaches. They present a cognitive model within ACT-R to explain how people make decisions in dynamic environments. They proposed that humans memorize specific *instances* of their interactions, such as the action taken and the associated outcomes, and use these memories

to inform future decisions. When confronted with a new decision, individuals retrieve the most typical instance from memory and use this instance's actions and outcomes to guide their current choice. This process is influenced by the perceived utility of past actions and the similarity between the current situation and stored instances. As individuals accumulate more experiences from the environment, their decision-making processes become more refined and better aligned with the changing environment.

MF RL, IBL both rely on associations between actions and outcomes without explicitly constructing a model of the environment. On the other hand, IBL can also be connected to MB in RL, particularly in the context of the ACT-R framework. In MB learning, agents plan actions by simulating the consequences of different choices. While IBL does not explicitly build a complete model of the environment, it relies on memory retrieval and the evaluation of previously encountered situations to inform decision-making. This aspect of IBL aligns with the cognitive processes in the MB approach, such as working memory and planning. Unlike MBRL, IBL makes specific predictions about which cognitive and neural resources are used. For instance, while RL traditionally forecasts reward processing and relies on procedural brain networks, IBL explicitly associates with memory circuits."

Present Study

In this paper, we aim to integrate the classical approach to MB with the insights gathered from the IBL use of long-term memory to guide decision-making. We propose an integrated cognitive model that relies on declarative long-term memory to implement MB learning, and uses ACT-R's declarative model to give cognitively plausible implementation of these operations. We argue for the superiority of our cognitive model over the traditional RL model and discuss the implications of this research for understanding the neural and computational mechanisms related to cost-benefit evaluation underlying decision-making strategies, as well as for applications in areas such as artificial intelligence and decision-making modeling.

Methods

Dataset

Nussenbaum et al. (2020) conducted an online experiment using the Markov two-stage task paradigm to replicate the main findings from Potter et al. (2017) and Decker et al. (2016) that MB learning increased as age increased. The de-identified behavioral data is obtained from the Open Science Framework by Nussenbaum et al., 2021: <https://osf.io/we89v/>. A total of 151 participants (fifty children; fifty adolescents, and fifty-one adults) were included in this study. The computational model in ACT-R was developed using a similar paradigm, but with abstract stimuli (A1, B1, B2, etc.) rather than a spaceship and alien stimuli. All participants were healthy adults with no

neurodevelopmental or neuropsychiatric disorders. The experimental protocol, subject recruitment procedures, and consent to share de-identified information were approved by the Institutional Review Board at Washington University.

The Two-Stage Task

In the Markov two-stage task (Figure 1), participants are presented with a series of trial screens, referred to as "states" and indicated as *A*, *B*, and *C*. Each state contains two options, indicated as *A1* and *A2*; *B1* and *B2*; *C1* and *C2*. Participants are asked to select one of the two options using the keyboard left or right. They always start in state *A* and, depending on the option they choose, will transition to state *B* or *C*. Once arriving at the end state *D*, participants would receive feedback informing whether they obtained a reward or not. Their choices in the second state determine their chance of receiving a reward. As the experiment progresses, participants learn the probabilistic nature of the transitions and rewards, updating their decision-making strategies accordingly. This learning process is one of the key aspects that researchers examine in the two-stage Markov decision task.

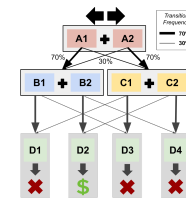


Fig 1: Markov Two-Stage Task Paradigm. Black arrows indicate the state transition probability, the thick line is 70% and the thin line is 30%. One of the final states is associated with a reward, and the probability of receiving a reward changes slowly across the experiment following a random walk with a mean of 0 and a standard deviation of 0.025.

The probability of a state transition from selecting one option in the first state to a specific second-stage state is predetermined, according to 70% of the time, and this state transition probability remained consistent throughout the entire experiment. For example, selecting *A1* has a 70% chance of resulting in the *B* state, referred to as the *common* state frequency, and a 30% chance of resulting in the *C* state, referred to as the *rare* state frequency. Similarly, selecting *A2* will have a 70% chance of resulting in the *C* state (*common*) and a 30% chance of resulting in the *B* state (*rare*). To promote continuous learning, the probability of receiving a reward for selecting the sequence of options slowly changes on each trial following Gaussian random walks ($M = 0$, $SD = 0.025$).

The task consists of two phases: a learning phase and a choice phase. Each stage is divided into three blocks. In the first two blocks of the learning phase ($N = 20$), participants are free to explore the transition probability between states by randomly selecting one of the two options, but no reward is given in the end. After a short break, participants are able to collect rewards at the end state with a slowly changing probability. This block is designed to allow participants to potentially learn the relative value of a sequence of choices

by experiencing the outcomes. The first two blocks of the choice phase are identical to the learning phase, but the last reward block consists of $N = 201$ trials. Participants must rely on the information they learned during the learning phase to make their selections.

MB and MF Patterns in the Two Stage Task

The two-stage task was developed to separate the contributions of MB and MF learning to decision-making. To understand how this is possible, one must consider two factors. The first is that, after feedback is delivered, the values of the actions that led to it would be updated accordingly. This means that the probability of repeating the same initial action in the same trial, indicated as the *Stay Probability*, would change. The second consideration is that MF and MB would update the values of the actions in different, and sometimes *opposite* ways. MF learning is blind to the circumstances that lead to the reward and would simply increase the value of the preceding actions. The MB learning system, by contrast, has access to information about the state transition probabilities and can update the values of actions based on them.

Specifically, if a reward was delivered after a first-stage action that led to a *rare* state transition (that is, one with a 30% chance of happening), the MB system would prefer to increase the value of the *opposite* action, since that has a greater likelihood of leading to the rewarded state. This, in turn, would lead to a decrease, rather than an increase, in the stay probability. In other words, while the stay probability is only affected by reward in MF, the stay probability shows an interaction of reward and transition frequency in MB, with the reward having opposite effects on actions that lead to common or rare transitions. Figure 2 illustrates the prototypical behavior of three RL models in this task.

Multiple empirical studies revealed that human participants demonstrated a mixture of MF and MB that combined elements of both MF and MB strategies (Gläscher et al., 2008; Daw et al., 2011; Otto et al., 2013). By analyzing the probability of staying with the same first-stage option as a function of reward and transition frequency, researchers can infer the extent to which participants rely on MF, MB, or hybrid learning strategies in the two-stage task.

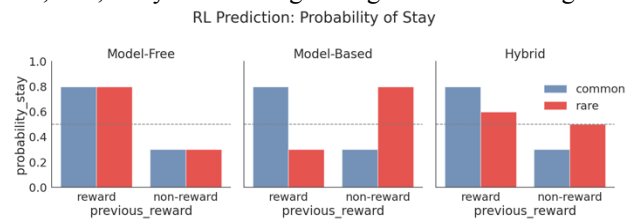


Fig 2. The canonical RL prediction of the probability of staying as a function of reward and transition frequency. (Left) Model-Free (Middle) Model-Based (Right) Hybrid

Computational Models

In the next section, we will illustrate two different model implementations capable of performing this task. The first is a “pure” RL model that combines MF and MB components

and includes no specific substrate for the MB component. This is the de-facto implementation that is commonly used across multiple research papers that have used this task (Daw et al., 2011; Doll et al., 2015; Kool et al., 2016; Gershman & Daw, 2017; Weissengruber et al., 2019). The second is a hybrid model that integrates ACT-R's declarative model within the MB component. The latter implementation aims to develop a more cognitively plausible and robust framework that can better simulate human cognitive processes in a dynamic and noisy environment.

Pure RL Model

MF Component. The central idea of MF learning is to gradually update the value of actions by calculating the difference between expected and actual rewards gained from the environment. The expected value of state-action is calculated based on the SARSA temporal difference algorithm using Eq 1 (Daw et al., 2011), where a denotes the learning rate, r denotes the immediate reward received after taking action a in state s ; the discount factor, denoted by gamma (γ), determines the importance of future rewards compared to immediate rewards; and (s, a) refers to state-action at current state, (s', a') refers to state-action at the next state.

$$Q(s', a') = Q(s, a) + \alpha \delta_{i,t}$$

$$\text{Where } \delta_{i,t} = r + \gamma Q(s'_t, a'_t) - Q(s_t, a_t) \quad (1)$$

MB Component. Unlike MF, which learns by trial-and-error, MB Learning is based on the idea that agents could create a mental representation of the environment and plan accordingly. Here, “planning” means simulating various future trajectories. This is achieved by predicting the consequences of possible actions and then using these predictions to select optimal actions.

The Q -value in MBRL is calculated and updated based on the Bellman equation (Eq 2). Similarly, the parameter r denotes the immediate reward received after taking action in state s ; the discount factor, denoted by gamma (γ), determines the importance of future rewards compared to immediate rewards. $\sum s'$ is a sum over all possible next states, and $\max[Q]$ represents the maximum Q -value over all possible actions a in the next state s' . $P(s'|s,a)$ is the transition probability that agents have knowledge about the dynamics of the environment. In this formulation, the transition probabilities are fixed (0.3/0.7) and directly provided as part of the model's knowledge.

Using this equation, the Q values are gradually updated, converging to the optimal Q which is the expected reward for all states and all action pairs under the best policy.

$$Q(s, a) = r + \gamma \sum_{s'} P(s' | s, a) \cdot \max[Q(s', a)] \quad (2)$$

Combination of MF and MB Components. Empirical findings suggest that human subjects tend to adopt hybrid approaches rather than pure MF or MB learning (Daw et al., 2011; Decker et al., 2016; Otto et al., 2013). They also imply that the interaction between these learning strategies

is crucial in determining decision-making behavior, with the balance between the two being influenced by the task's state and transition structures.

One common approach to combining MF and MB learning approaches is to use the weighted sum of the MB and MF value estimates to make decisions. A weight parameter w determines the relative importance of the MB and MF estimates. In these models, the agent starts by using a MB strategy to plan its actions, but as it gathers more information about the environment, it shifts towards a MF strategy. This allows the agent to quickly adapt to changes in the environment while still being able to plan its actions efficiently.

In pure MF learning, the weight parameter is set to 0, while in pure MB learning, it is set to 1. In a hybrid model, the weight parameter can be set to a value between 0 and 1, depending on the task and the available information. The weight parameter can be learned through a process called "weight adjustment," where the agent adjusts the weight based on the performance of the current strategy. This allows the agent to adapt to the changes in the environment and find the optimal balance between exploration and exploitation. However, the neurobiological meaning of this parameter is less clear, especially from the perspective of decision-making and its underlying cognitive processes.

Hybrid ACT-R RL Model

MF Component. The MF component of the hybrid model is identical to the MF component of the pure RL model. In turn, this component is also broadly consistent with ACT-R's procedural knowledge module, which also uses RL to learn stimulus-response associations in the form of procedural rules. Thus, we employed the SARSA MF framework mentioned above as a substitute for ACT-R's procedural module.

MB Component. RL-MB learning is creating a model of the environment, which allows the agent to plan its actions by simulating the consequences of different choices. Critically, it depends on the knowledge about transition probability, that is, how likely it is to move from the initial state to the next, given a particular action. In the pure RL model, these probabilities are directly provided to the model. However, this assumption may not fully capture the nature of learning and cognitive processes in the task. This knowledge is not simply given, but actively updated and accumulated by agents from the environment through their interactions with the external world.

Thus, in the hybrid model, the MB component encodes its knowledge of the environment as episodic long-term memories. Much like in the IBL approach, the model retrieves and inspects these traces as part of its planning process. The model's long-term memory was developed using PyACTUp (Morrison, 2019), a Python implementation of ACT-R's declarative system. We argue that our ACT-R model performs as well as RL models in simulating canonical MB behavioral patterns, and even more importantly, it provides a plausible cognitive

framework to understand how our brain represents the planning process.

Specifically, as the agent observes the two options, a two-step process of planning begins. The agent tends to retrieve the most likely (most frequent and recent) subsequent states given two possible actions ('f' and 'k') from the declarative memories. Based on the retrieved next state s' , the MB Q value of two possible state 2 actions is calculated based on the Eq 3.

$$Q(s, a) = P(s'_1|s, a)max[Q(s'_1, a)] + P(s'_2|s, a)max[Q(s'_2, a)] \quad (3)$$

Additionally, unlike RL, our declarative model estimates the transition frequency based on prior memories of trials. It samples the memories about state transitions and calculates the probability of *state1-state2* given an action. That is, among the retrieved chunk samples, the estimated transition probability, $P(s'|s, a)$, is calculated by the number of first state (A) to second state (e.g. B or C) divided by the total number of sampled memories given a particular action a . Here, the number of sampling times is fixed at an arbitrary number of 20. Although we did not investigate further how the number of sampling counts affects decision-making, it is reasonable to hypothesize that a larger n suggests a more accurate assessment of transition frequency and may indicate a greater effort in cognitive control planning or an individual's greater working memory (WM) capacity. After estimating $P(s'|s, a)$, the Q -values are computed as in Eq 3.

This method has a significant advantage in approximating realistic transition frequencies for decision-making processes because it uses declarative memory to make educated estimates about the frequency of state-action pairs. In contrast to conventional MBRL algorithms, which typically assume a fixed 0.3/0.7 transition frequency as the basis for Q -value calculations, our ACT-R model incorporating declarative memory provides a more realistic representation of human cognitive processes. Because it is built upon a reliable model of memory, the estimates made by the agent reflect some of the distortions and fallacies of humans.

At the end of each trial, the agent forms a new episodic memory of the interaction, containing the states and actions taken. A new chunk, consisting of 5 slots: 'stage', 'current_state', 'next_state', 'response', and 'reward' is created and merged into existing Long-Term memory.

This two-step planning at stage 1 allows the agent to retrieve the most rewarding action sequence based on prior learning experience, taking reward and state transition frequency into account. The probability of a sequence being retrieved depends on the corresponding memory's base level activation B_i , which is computed as shown in Eq. 4:

$$A_i = B_i + \epsilon \text{ where } B_i = \ln\left(\sum_{j=1}^n t_j^{-d}\right) \quad (4)$$

This equation describes the activation of chunks calculated with a base-level learning function (B_i) and random noise (ϵ), which reflects the recency of previous retrievals. The base level activation B_i captures the history of

use of a particular chunk, taking into account both the frequency and the recency of a particular memory i being retrieved at time t . The calculation is based on the idea of time decay, where more recent uses contribute more to activation than less recent ones. The decay rate is an individual-level parameter (Sense et al 2016) that determines how quickly the contribution of past memories diminishes over time. A higher decay rate leads to more rapid forgetting, emphasizing the role of recent interactions, while a lower decay rate allows the influence of older interactions to persist longer.

In addition to base-level activation, noise plays a pivotal role in the ACT-R's declarative memory framework. Noise, denoted by ϵ , adds a level of randomness to the activation formula, which lets brain processes be different and hard to predict. This lets the model better represent the wide range of actions seen in actual data. When noise is added, chunks with less activity are more likely to be chosen. This encourages exploration and could lead to the discovery of better tactics.

Moreover, ACT-R provides a way to connect a memory's activation with the time it takes to remember it, thus making predictions in the response time domain as well as the accuracy domain (shown in Eq 5), where A_i represents the activation of the memory item, and F is the latency factor parameter in ACT-R (:lf) reflecting the speed of retrieving processes. A fixed cognitive temporal cost is applied in addition to memory retrieval time. This model thus encapsulates the dynamics of memory retrieval, indicating that items with higher activation levels (i.e., more frequently or recently accessed) tend to be retrieved more quickly.

$$\text{Response Time}_i = Fe^{-A_i} + \text{fixed cost} \quad (5)$$

The state2 planning is simpler than state1, since it consists of only one-step planning, with more restricted information. After attending to and encoding the state2 stimuli into the working memory, the agent sends a retrieval request asking for a state2-stimulus chunk that matches the state1 slot value. For example, if an agent chose to leave at state 1 and end up at state B. The state2 plan is searching any chunk that contains the current state and is equal to B. If the retrieved memory has a reward greater than 0, then choose this action as the state 2 response, otherwise, choose the alternative action.

Similarly, the most frequent and recent *state2-stimulus* is retrieved based on the activation calculation, allowing the agent to choose the most available memories (s' , a') by observing action-outcome associations from prior trials. It's worth pointing out that our model doesn't depend on the blending mechanism typically associated with the IBL module. Instead, it leverages more general memory mechanisms, similar to those found in ACT-R's declarative system.

Optimization with Maximum Log-Likelihood

Both of the models were fitted to each individual using maximum log-likelihood approach, which is a standard

method used to estimate individual subjects' data in cognitive modeling research (Yang & Stocco 2021). This method involves calculating the log-likelihood function, which measures the goodness of fit between the observed data and the hypothesized probability distribution. Using the softmax choice equation, as shown in Eq 6, we calculated the probability of selecting a specific response (either left or right) given the model parameters to estimate individual subjects' data using the log-likelihood approach.

$$P(a | s) = \frac{e^{\beta \cdot Q(s, a)}}{\sum_{a'} e^{\beta \cdot Q(s, a')}} \quad (6)$$

We adopted a simpler parameter estimation pipeline, as in Decker et al. 2016; Potter et al. 2017; Nussenbaum et al., 2021. Three Q-learning parameters (α , β and λ , w), and two memory relevant parameters (temperature and decay) are fit to each individual subject. α is the learning rate in Q-Learning, β is the free-parameter fit to each subject's choice that scales the Q-value. λ is the reward discount parameter. Higher values of w indicate greater recruitment of a MF, while higher values of w indicate more use of a MB learning strategy. Temperature and decay parameters are two memory relevant parameters that describe how noisy a memory is, and how fast a memory is forgotten across time.

Then, the log-likelihood function is calculated for each individual subject using the observed data. Finally, the maximum likelihood estimation technique is used to estimate the parameters of the distribution that best fit the individual subject's data.

To estimate the optimized parameter for each individual subject, we computed the probability of the two states' responses and calculated the log-likelihood of each subject's performance using the SciPy library (Virtanen et al., 2020). This process is repeated for each subject 10 times in the sample, resulting in estimates of individual subjects' data based on the highest log-likelihood value among 10 optimizations.

Results

As shown in Figure 3, our ACT-R hybrid model is able to replicate the canonical response switch patterns in MF, MB and hybrid in RL. Furthermore, our ACT-R model is capable of predicting response time, whereas most RL models fail to do so.

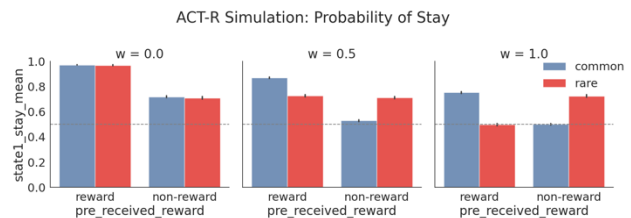


Fig 3. ACT-R simulation: The averaged probability of stay by previous reward and transition frequency, with various degrees of MB Learning ($w = 0, 0.5, 1$). $\alpha = .8$, $\beta = 5$, $\lambda = 0.6$, temperature = 0.2, decay = 0.5, and each agent simulated 201 trials 100 times. The blue bar denotes common transitions of previous

trials, while the red bar denotes rare transitions of previous trials; x-axis represents the outcome of previous trial, reward or non-reward; y-axis is the simulated mean stay probability; error bar represents the standard error of means.

When it comes to parameter estimation, our ACT-R model largely agrees with the RL models. We found a significant positive correlation between RL and ACT-R Hybrid optimized parameters (α : $r = 0.24$, $p = 0.0025$; β : $r = 0.27$, $p < 0.001$, λ : $r = 0.31$, $p < 0.001$) and, moreover, the maximum log-likelihood value is positively correlated ($r = 0.86$, $p < 0.001$) between two models.

We also examined the effect of the decay and temperature parameters of the ACT-R Hybrid Model on the probability of staying. Figure 4 demonstrates how the mean probability of staying changes as a function of decay, and temperature. As we expect, with lower decay, the agent is better at recalling the state-action association, and in turn, such an accurate estimation of state-action frequency encourages the usage of the MB learning strategy. On the other hand, high decay suggests worse memory recall, leaving agents no choice but to rely on the MF learning strategy.

Effects of Long-Term Memory on Decision Times

According to empirical data from Nussenbaum et al., (2020), it takes longer for participants to respond if a previous trial is rarer than common, as shown in Figure 5 (Left). Figure 5(Right) illustrates the simulated mean response time of common vs. rare trials in the hybrid model. This result is explained by the memory mechanisms in ACT-R, whereby the memory of a more common event has higher activation, which makes it easier and faster to recall. In contrast, this effect is not immediately predicted by the classic, pure RL model, primarily because RL doesn't incorporate any mechanisms to forecast response times.

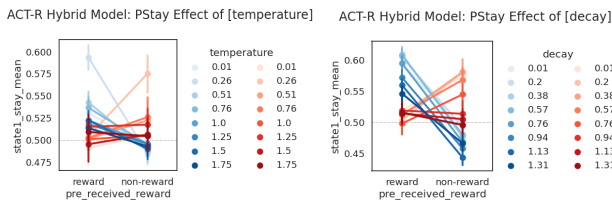


Fig 4. The simulated (ACT-R Hybrid Model) mean probability of stay as a function of two parameters: temperature (left) and decay (right). The temperature parameter ranges from 0.01 to 1.75, simulating 200 trials 100 times. The decay parameter ranges from 0.01 to 1.4, simulating 200 trials 100 times. Blue denotes previous common transition frequency trials, and red color denotes rare ones. The shades of color denote parameter magnitude, where small value is in shallow shade, and large value in dark shade. x-axis represents the outcome of previous trial, reward or non-reward; y-axis is the simulated mean stay probability; error bar represents the standard error of means.

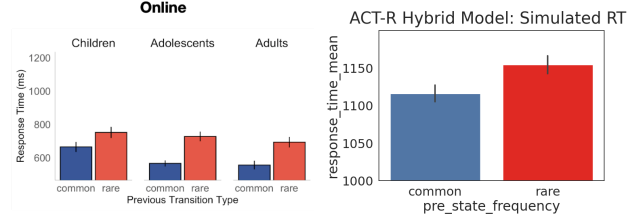


Fig 5. (Left) Empirical data of response time (ms) by transition type for three age groups from Nussenbaum et al., (2020). (Right) Simulated response time (ms) of ACT-R Hybrid Model by state transition frequency. $\alpha = .8$, $\beta = 5$, $\lambda = 0.6$, temperature = 0.2, decay = 0.5, $l_f = 0.63$ and simulated 201 trials 100 times. The black line represents the standard error of the mean. Blue denotes previous common transition frequency trials, and red color denotes rare ones; y-axis is the simulated mean response time in ms; error bar represents the standard error of means

Capturing Individual Developmental Differences

The individually-fitted parameters also provide insight into the developmental data observed in Nussenbaum et al. (2020). Specifically, the authors showed that the use of the MB component increases with age. In the pure RL model, this effect can only be explained by altering the weight parameter according to age. When examining the individual parameters of the hybrid models, however, we found a significant negative correlation between the optimized decay rate d (Eq 4) and the age of each subject ($r = -0.21$, $p < 0.01$), as shown in Figure 6, suggesting that younger subjects have a higher decay rate than older subjects. This is in line with the development of cognitive abilities in children, and provides an explanation for why, as people get older, they tend to rely more on declarative memory (MB strategy) in recreating a mental model of the environment.

Correlation of optimized parameters [age] vs. [param_value.actr]:

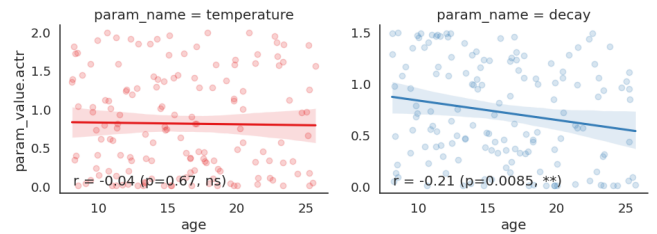


Fig 6. The correlation of age and optimized memory relevant parameters from the ACT-R Hybrid Model fit to each subject (N = 151) is shown above. Each subject was optimized 10 times with random initial parameter seeds. There is a negative correlation between age and the memory decay parameter ($r = -0.21$, $p < 0.01$), but no correlation between age and temperature. X-axis represents subject's age, and y-axis denotes the optimized parameter values for both temperature and decay, extracted from ACT-R model

Discussion

In this paper, we have explored the cognitive underpinnings of decision-making and learning in complex environments, focusing on the interplay between model-based (MB) and model-free (MF) reinforcement learning (RL) strategies.

The declarative framework in ACT-R provides a cognitive representation of how individuals adapt to unstable environments by leveraging memory sampling and incorporating this inherent uncertainty into human decision-making. We propose that MBRL is closely tied to declarative memory, with individuals relying on memory for previous episodic traces when making plans. To test this idea, we implemented an ACT-R model that incorporates declarative memory and procedural Q learning in the decision-making process. Our results provide compelling evidence for the relationship between memory resources and the mixture of MB and MF strategies.

By establishing this connection between our results and the developmental data from Nussenbaum et al. (2020), we provide further evidence for the importance of declarative memory in the decision-making process and highlight the developmental changes that occur in the balance between MB and MF strategies. These findings contribute to a more comprehensive understanding of the cognitive foundations of MBRL and the factors that influence the balance between MB and MF strategies across different age groups. Ultimately, this knowledge may inform the development of age-appropriate interventions and strategies to improve decision-making outcomes throughout the lifespan.

We offer a unique perspective on decision-making in dynamic environments, incorporating elements of both the MF and MB RL approaches and, most importantly, providing a plausible cognitive framework for planning. This synthesis of reinforcement learning principles within the ACT-R framework provides valuable insights into the cognitive mechanisms underlying human decision-making and adaptation. Our work underscores the necessity of considering the contributions of multiple memory systems when investigating the balance between MB and MF approaches to decision-making.

References

- Daw, N. D., Gershman, S. J., Seymour, B., Dayan, P., & Dolan, R. J. (2011). Model-Based influences on humans' choices and striatal prediction errors. *Neuron*, 69(6), 1204–1215. <https://doi.org/10.1016/j.neuron.2011.02.027>
- Doll, B. B., Shohamy, D., & Daw, N. D. (2015). Multiple memory systems as substrates for multiple decision systems. *Neurobiology of Learning and Memory*, 117, 4–13. <https://doi.org/10.1016/j.nlm.2014.04.014>
- Gershman, S. J., & Daw, N. D. (2017). Reinforcement learning and episodic memory in humans and animals: An integrative framework. *Annual Review of Psychology*, 68(1), 101–128. <https://doi.org/10.1146/annurev-psych-122414-033625>
- Gläscher, J., Hampton, A. N., & O'Doherty, J. P. (2008). Determining a role for ventromedial prefrontal cortex in encoding action-based value signals during reward-related decision making. *Cerebral Cortex*, 19(2), 483–495. <https://doi.org/10.1093/cercor/bhn098>
- Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science*, 27(4), 591–635. https://doi.org/10.1207/s15516709cog2704_2
- Kool, W., Cushman, F. A., & Gershman, S. J. (2016). When does model-based control pay off? *PLOS Computational Biology*, 12(8), e1005090. <https://doi.org/10.1371/journal.pcbi.1005090>
- Kotseruba, I., & Tsotsos, J. K. (2018). 40 years of cognitive architectures: Core cognitive abilities and practical applications. *Artificial Intelligence Review*, 53(1), 17–94. <https://doi.org/10.1007/s10462-018-9646-y>
- Morrison D. (2019). A lightweight Python implementation of a subset of the ACT-R cognitive architecture's Declarative Memory. <https://pypi.org/project/pyactup>
- Niv, Y. (2009). Reinforcement learning in the brain. *Journal of Mathematical Psychology*, 53(3), 139–154. <https://doi.org/10.1016/j.jmp.2008.12.005>
- Nussenbaum, K., Scheuplein, M., Phaneuf, C. V., Evans, M. D., & Hartley, C. A. (2020). Moving developmental research online: Comparing in-lab and web-based studies of model-based reinforcement learning. *Collabra: Psychology*, 6(1). <https://doi.org/10.1525/collabra.17213>
- Otto, A. R., Gershman, S. J., Markman, A. B., & Daw, N. D. (2013). The curse of planning. *Psychological Science*, 24(5), 751–761. <https://doi.org/10.1177/0956797612463080>
- Stocco, A., Prat, C. S., & Graham, L. K. (2021). Individual differences in reward-based learning predict fluid reasoning abilities. *Cognitive Science*, 45(2). <https://doi.org/10.1111/cogs.12941>
- Sutton, R. S. (1991). Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4), 160–163.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>
- Weissengruber, S., Lee, S. W., O'Doherty, J. P., & Ruff, C. C. (2019). Neurostimulation reveals context-dependent arbitration between model-based and model-free reinforcement learning. *Cerebral Cortex*, 29(11), 4850–4862. <https://doi.org/10.1093/cercor/bhz019>
- Yang, Y. C., Karmol, A. M., & Stocco, A. (2021). Core cognitive mechanisms underlying syntactic priming: A comparison of three alternative models. *Frontiers in Psychology*, 12. <https://doi.org/10.3389/fpsyg.2021.662345>
- Yang Y. C., Sibert C and Stocco A (2023). Strategy from Structure: Individual Preferences in Decision Making Strategies Adaptively Reflect Differences in Brain Network Connectivity. (In Prep)
- Yang Y. C., Stocco A (2022). Allocating Mental Effort in Cognitive Tasks: A Model of Motivation in the ACT-R Cognitive Architecture. (Under Review)

A Diffusion Model Decomposition of the Unit-Decade Compatibility Effect in Two-digit Number Comparison

Bella E. Zapata (bella.zapata@go.tarleton.edu)

Thomas J. Faulkenberry (faulkenberry@tarleton.edu)

Department of Psychological Sciences, Tarleton State University, Stephenville, TX 76402 USA

Abstract

Mechanisms of number processing have been of interest to cognitive psychologists for many years. There are multiple competing theories to explain how people form mental representations of two-digit numbers. Nuerk et al. (2001) proposed a decomposed representation, where the decade and unit digits are processed separately. Primary evidence came from the unit-decade compatibility effect, where comparisons when both unit and decade digits obey the same order relation (e.g., 23 versus 55, where both $2 < 5$ and $3 < 5$) are faster than trials where the order of digit relations is opposite (e.g., 27 versus 55, where $2 < 5$ but $7 > 5$). In this study, we used mathematical modeling to perform a decomposition of the unit-decade compatibility effect. We analyzed data from 53 adult observers, each of whom completed a two-digit number comparison task. Each observer's distribution of RTs (split by compatibility condition) was fit to a diffusion model. We used the EZ-diffusion method (Wagenmakers et al., 2007) to obtain estimates of drift rate and nondecision time for each design cell. The estimates were then compared with a Bayesian paired samples *t*-test. As expected, compatible trials were faster than incompatible trials. This mean effect manifested almost entirely in the drift rate, which was smaller for incompatible trials than for compatible trials. Critically, the nondecision time did not differ between conditions. This implies that the unit-decade compatibility effect is due entirely to decision-related processes (e.g., stimulus information uptake) but not auxiliary nondecision processes (e.g., encoding, motor preparation, etc.). This work helps to shed light on the locus of the unit decade compatibility effect, and more broadly, on the nature of decomposed processing in numerical cognition.

Keywords: Number comparison; diffusion modeling; EZ-diffusion; Bayesian hypothesis testing.

Introduction

Mental representations of numbers are studied by investigating the various behavioral patterns obtained in cognitive tasks. In the case of symbolic numbers, one popular task is a numerical comparison task, where participants complete a number of trials on which they quickly judge whether each presented number is greater than (or less than) a fixed comparison standard (e.g., 5). A classic finding (Moyer & Landauer, 1967) is the *numerical distance effect*, where the response time increases as the numerical distance between the stimulus number and the comparison standard decreases. For example, people typically respond “larger” faster when 9 is presented than when 6 is presented; one classic explanation for this is that the internal magnitude representation is inherently imprecise and variable (i.e., “fuzzy”), and increasing the distance between to-be-compared numbers reduces their rep-

resentational overlap, resulting in a faster decision (Verguts, Fias, & Stevens, 2005).

In the context of two-digit numbers, we observe similar phenomena. Indeed, Hinrichs, Yurko, and Hu (1981) found a numerical distance effect for two-digit number comparison; participants' response times decreased as the numerical distance from the to-be-compared number increased from the comparison standard 55. Dehaene, Dupoux, and Mehler (1990) observed a similar result. In both cases, the observed numerical distance effect was taken as evidence of a *holistic* representation of two-digit numbers, where the two separate digits (the decade and unit digits) in the two-digit number stimulus are merged into a single representational unit.

Despite this simple explanation for the observed numerical comparison behavior, increasing evidence has pointed to a *decomposed* representation of two-digit numbers. Primary evidence for the decomposed account comes from Nuerk, Weger, and Willmes (2001), who observed a *unit-decade compatibility effect* in two-digit number comparison. That is, when the decade and unit digits of one number were both smaller (or both larger) than both digits of the other number (i.e., unit-decade *compatible*), response times were faster than if the digits were unit-decade *incompatible* with each other. For example, the comparison 23 versus 55 would be considered unit-decade compatible, whereas the comparison 27 versus 55 would be considered unit-decade incompatible (see Figure 1). In the former, the individual comparisons for each digit are in the same ordinal relationship to the digits in the comparison standard. That is, both the decade (2) and unit (3) are *less than* the corresponding digits from the standard 55. In the latter, the comparisons are reversed; in this case, the decade comparison is *less than* (i.e., $2 < 5$), but the unit comparison is *greater than* (i.e., $7 > 5$). The presence of the unit-decade compatibility effect indicates that people make obligatory comparisons of the both decade and unit digits when comparing two-digit numbers, even though (remarkably) the decision can be made entirely by comparing the decade digits alone.

Since the original discovery of the unit-decade compatibility effect, a number of studies have further confirmed the presence of decomposed processing in two-digit number comparison. For example, while Nuerk et al. (2001) based their conclusion on a comparison task where pairs of two-digit numbers were presented to be compared, Moeller,

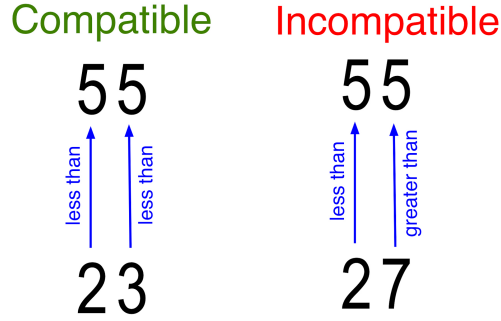


Figure 1: An illustration of unit-decade compatibility (compatible versus incompatible) in two-digit number comparison.

Nuerk, and Willmes (2009) observed a unit-decade compatibility effect for a comparison task using fixed standards (53 and 57). In such trials, the decision can be made entirely by comparing the decade digits alone, but Moeller et al. (2009) demonstrated that parallel and separate comparisons of the decade and unit digits occur still. Further probing of these decomposed processing signatures has indicated that the size and nature of the unit-decade compatibility effect can be manipulated by introducing variations in stimulus properties (Macizo & Herrera, 2011) or task instructions (Reynvoet, Notebaert, & Van den Bussche, 2011; Faulkenberry, Cruise, & Shaki, 2017, 2018).

Additionally, several researchers have investigated the unit-decade compatibility effect in the context of a general theory of numerical cognition (Verguts et al., 2005; Verguts & De Moor, 2005), which proposes that such compatibility effects occur due to competition between parallel and partially active responses. This response-competition account has been successful in explaining a variety of phenomena in numerical cognition, including the numerical distance effect (Erb, Moher, Song, & Sobel, 2018; Faulkenberry, 2016), SNARC effect (Gevers, Verguts, Reynvoet, Caessens, & Fias, 2006; Faulkenberry, 2014), decomposed process in fraction comparisons (Faulkenberry, Montgomery, & Tennes, 2015), and the size-congruity effect (Faulkenberry, Cruise, Lavro, & Shaki, 2016; Sobel, Puri, & Faulkenberry, 2016; Sobel, Puri, Faulkenberry, & Dague, 2017). One common thread between these studies is that they provide converging evidence that the timecourse of numerical compatibility effects tends to reflect *late interaction* (i.e., decision-related effects) rather than *early interaction* (i.e., encoding/perceptual effects).

In the present study, we employed *diffusion modeling* (Ratcliff, Smith, Brown, & McKoon, 2016) to decompose participants' response times into components which can provide separate measures of early and late processes in two-digit number comparison. Most of the past studies cited above have relied on the common technique of collapsing the distribution of response times observed in each design cell of an experiment (usually using the mean). Given that response times typically exhibit a positive skew, collapsing each cell to

a single mean results in a loss of information about the shape of each participant's distribution of response times. Additionally, information about errors is typically lost, because analyses of mean response times is typically performed only on the correct trials.

Diffusion modeling

To account for both response speed and response accuracy, Ratcliff (1978) proposed the diffusion model, which mathematically characterizes a distribution of observed response times as the collection of stopping times for a noisy random walk with drift toward a fixed boundary. Though the full diffusion model has seven parameters, for the purposes of this paper we will focus on three: drift rate v , boundary separation a , and nondecision time T_{er} . The drift rate v is the rate at which stimulus information is accumulated toward either the upper or lower response boundary; as an index of cognitive processing, it represents task difficulty. Boundary separation a represents the separation between the lower and upper response boundaries and indexes the amount of information required to be accumulated before triggering a response. Finally, the nondecision time T_{er} is the part of the total response time that is not related to the accumulation of noisy information; that is, it represents other non-cognitive processes, such as perceptual and/or encoding processes, as well as motor responses. Variations in these parameters can index individual differences, task demands, or instructions. For example, variations in boundary separation a can describe the impulsivity of a participant. Smaller values of a would represent sporadic or impulsive decision makers, whereas larger values of a would represent more conservative or careful decision makers.

Though various methods exist to fit full diffusion models to observed data (Wabersich & Vandekerckhove, 2014), we elected to use the *EZ diffusion model* of Wagenmakers, Van Der Maas, and Grasman (2007). The EZ diffusion model provides a series of closed-form equations to estimate v , a , and T_{er} directly from the descriptive statistics of a set of observed response times. To use the EZ diffusion model, one needs to provide the mean response time (MRT), the variance of the response times (VRT), and the proportion of trials which represent a correct response (P_c). The first step is to input the VRT and P_c into the following equation to estimate drift rate:

$$v = \frac{\text{sign}\left(P_c - \frac{1}{2}\right)}{10} \left[\frac{\text{logit}(P_c) \left(P_c^2 \text{logit}(P_c) - P_c \text{logit}(P_c) + P_c - \frac{1}{2} \right)}{VRT} \right]^{\frac{1}{4}}.$$

Note that in cases where no errors were made (i.e., $P_c = 1$, which produces the undefined term $\text{logit}(1)$), we applied an edge correction

$$P_c = 1 - \frac{1}{2n},$$

where n is the number of trials. Next, boundary separation a is calculated directly from the obtained drift rate v and P_c :

$$a = \frac{0.01 \cdot \text{logit}(P_c)}{v}.$$

Finally, nondecision time T_{er} is calculated by subtracting from the mean response time:

$$T_{er} = MRT - \left(\frac{a}{2v} \right) \cdot \frac{1 - \exp(-100va)}{1 + \exp(-100va)}.$$

The present study

In the present study, we applied the EZ diffusion model to a set of response time from a two-digit number comparison task. Our primary goal was to investigate the effect of unit-decade compatibility on the diffusion model parameters. If the locus of the unit-decade compatibility effect is in late, response-related stages, then the primary diffusion parameter impacted by compatibility should be the drift rate v . On the other hand, if the locus of the unit-decade compatibility effect is early in nature, then the primary diffusion parameter impacted by compatibility should be the nondecision time T_{er} .

Because we may possibly observe null effects on one or more diffusion model parameters, we employed Bayesian hypothesis testing (Wagenmakers, 2007; Faulkenberry, Ly, & Wagenmakers, 2020), which allowed us to assess the evidence for either the alternative hypothesis \mathcal{H}_1 or the null hypothesis \mathcal{H}_0 . Instead of relying solely on the p -value, which gives the likelihood of the observed data (or more extreme) under \mathcal{H}_0 only, we also computed the *Bayes factor*

$$BF_{10} = \frac{p(\text{data} | \mathcal{H}_1)}{p(\text{data} | \mathcal{H}_0)},$$

which provides a continuous index of the ability of each model to predict the observed data. Additionally, we computed posterior probability of the winning model. Assuming prior odds of 1:1, the posterior probability of \mathcal{H}_1 can be computed directly from the Bayes factor as

$$\Pr(\mathcal{H}_1 | \text{data}) = \frac{BF_{10}}{1 + BF_{10}}.$$

In cases where \mathcal{H}_0 is the winning model, the calculation of $\Pr(\mathcal{H}_0 | \text{data})$ proceeds by replacing BF_{10} with BF_{01} .

Method

Participants

For this study we used an existing unpublished dataset from Cipora et al. (2022), which we downloaded from <https://osf.io/pm4zt>. The dataset includes 53 students from Loughborough University (43 females, mean age = 23.2 years, age range 18 to 29 years) who completed a two-digit number comparison task in a single session.

Design and procedure

The design and procedure were fully described in the dataset's metadata, available at <https://osf.io/dpjm2>. We describe it briefly here for convenience. During the task, two-digit number pairs were presented with vertical separation

around a centrally presented fixation cross. The stimuli were presented in 24-point white bolded Courier New font on a black background. Stimuli remained on the screen until a response was recorded or a maximum of 3000 ms elapsed. Each trial was followed by an intertrial interval of 500 ms. Participants that were assigned to groups of 6 where they worked individually on assigned laptops. Before the task began, each participant was given 12 practice trials, where they were instructed on the basic guidelines of the task as well as given the instruction to press T for an upper number stimulus response and V for lower number stimulus response. All participants used a standard QWERTY keyboard. After the practice trials, there were 4 blocks of 60 trials per participant, and participants were given the option to rest after each block. The trial order was completely randomized for each participant with compatible and incompatible trials mixed randomly throughout each block.

Results

Participants completed a total of 12,720 trials. We then removed 17 trials that were less than 200 ms and an additional 76 trials that exceeded 3000 ms, leaving a total of 12,627 trials for analysis (retaining 99.3% of the original trials). There were a total of 587 error trials (total error rate = 4.6%). Now we will describe the general data processing workflow, which consisted of the following steps:

1. First, we separated the observed responses into 106 design cells, formed by crossing 53 participants by 2 trial types (compatible, incompatible);
2. Next, we extracted summaries of the distributions of response times in each cell in two different ways. First, we used the traditional method of collapsing each distribution to a mean response time. Next, we used the EZ diffusion equations to transform the observed response times and proportions correct in each of the design cells into 3 diffusion model parameter estimates: drift rate v , boundary separation a , and nondecision time T_{er} . Because each participant completed trials in both the compatible and incompatible conditions, this resulted in 6 parameter estimates for each participant: 3 for the compatible trials and 3 for the incompatible trials. In total, this yielded $6 \times 53 = 318$ EZ diffusion parameter estimates. Note that since instructions did not vary between compatible and incompatible trials, we assumed that boundary separation a remained constant between conditions. Thus, once we estimated a for compatible trials, we subsequently used that value of a in our computation of nondecision time T_{er} for incompatible trials.
3. Finally, we submitted the mean response times and the collection of diffusion model parameters to a Bayesian paired samples t -test (Rouder, Speckman, Sun, Morey, & Iverson, 2009). For each test, we assessed the predictive adequacy of two competing models on the unit-decade compatibility

effect δ : a null model $\mathcal{H}_0 : \delta = 0$ versus a two-sided alternative model $\mathcal{H}_1 : \delta \sim \text{Cauchy}(1/\sqrt{2})$.

All raw data processing was done with R version 4.3.0 (R Core Team, 2023), and all hypothesis tests were performed using JASP version 0.17.1 (JASP Team, 2023).

Mean response times

As expected, we observed the typical unit-decade compatibility effect on mean response times. Response times were longer for incompatible trials ($M = 720$ ms, $SD = 142$ ms) than for compatible trials ($M = 674$ ms, $SD = 131$ ms), $t(52) = 10.2$, $p < 0.001$, $BF_{10} = 1.5 \times 10^{11}$, $\Pr(\mathcal{H}_1 | \text{data}) > 0.999$ (see Figure 2).

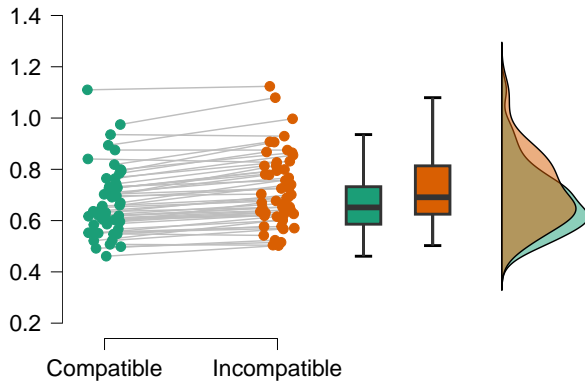


Figure 2: A raincloud plot illustrating the effect of unit-decade compatibility on mean response time (sec.).

In Figure 3, we see the distribution of response time differences between compatible trials and incompatible trials. The mean difference was observed to be 46 ms; in terms of standardized effect size δ , this was a large effect, with a 95% credible interval of (0.99, 1.76).

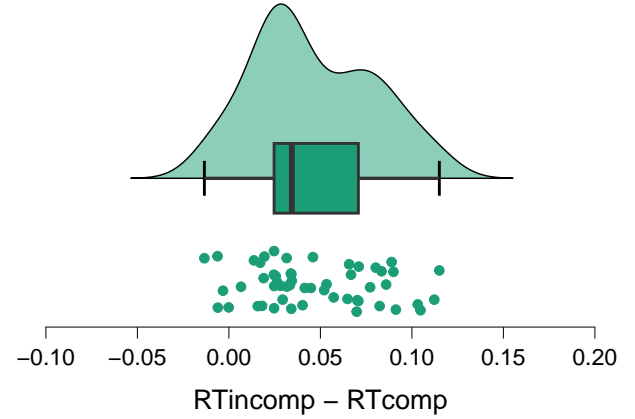


Figure 3: A raincloud difference plot illustrating the distribution of observed differences in mean response times between compatible and incompatible trials.

Drift rate v

We observed a smaller drift rate v for incompatible trials ($M = 0.219$, $SD = 0.058$) compared to compatible trials ($M = 0.269$, $SD = 0.058$), $t(52) = -9.5$, $p < 0.001$, $BF_{10} = 1.2 \times 10^{10}$, $\Pr(\mathcal{H}_1 | \text{data}) > 0.999$ (see Figure 4).

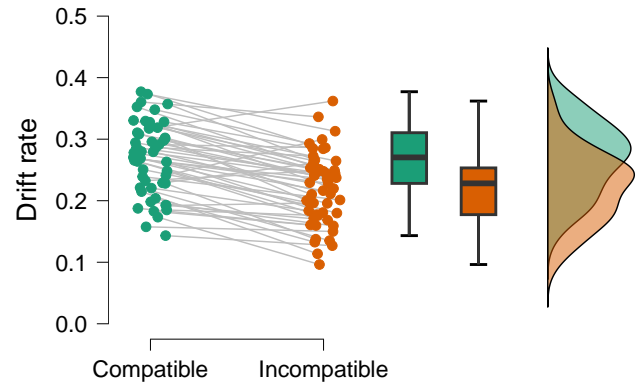


Figure 4: A raincloud plot illustrating the effect of unit-decade compatibility on drift rate v .

In Figure 5, we see the distribution of raw drift rate differences between compatible trials and incompatible trials. The mean difference was observed to be -0.050; in terms of standardized effect size δ , this was a very large negative effect, with a 95% credible interval of (-1.637, -0.899). Thus, the rate of information accumulation was greatly reduced on incompatible trials, implying that the effect of unit-decade compatibility persists in the decision time component.

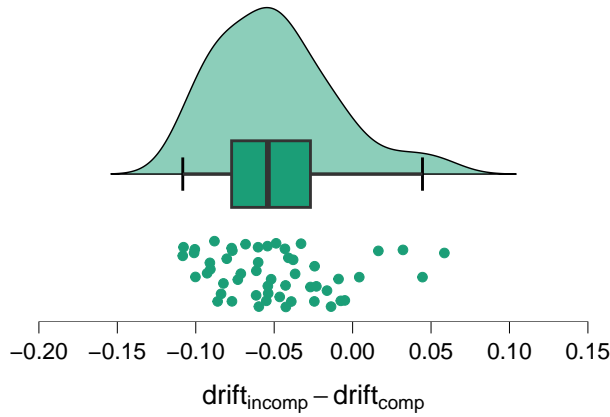


Figure 5: A raincloud difference plot illustrating the distribution of observed differences in drift rate v between compatible and incompatible trials.

Nondecision time T_{er}

Critically, there was *no difference* in nondecision time T_{er} between compatible trials ($M = 392$ ms, $SD = 71$ ms) and incompatible trials ($M = 394$ ms, $SD = 81$ ms), $t(52) = 0.4$, $p = 0.703$, $BF_{01} = 6.2$, $\Pr(\mathcal{H}_0 \mid \text{data}) = 0.861$ (see Figure 6).

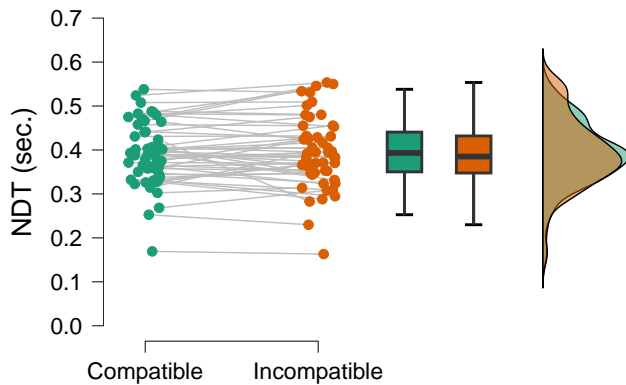


Figure 6: A raincloud plot illustrating the effect of unit-decade compatibility on nondecision time T_{er} .

In Figure 7, we see the distribution of raw drift rate differences between compatible trials and incompatible trials. However, as $\mathcal{H}_0 : \delta = 0$ is the best fitting model, there is no need to compute a credible interval for δ . While there is a large effect of unit-decade compatibility on drift rates, there is no such effect on nondecision time. Thus, we can conclude that the unit-decade compatibility effect is isolated to decision processes, not encoding or response processes.

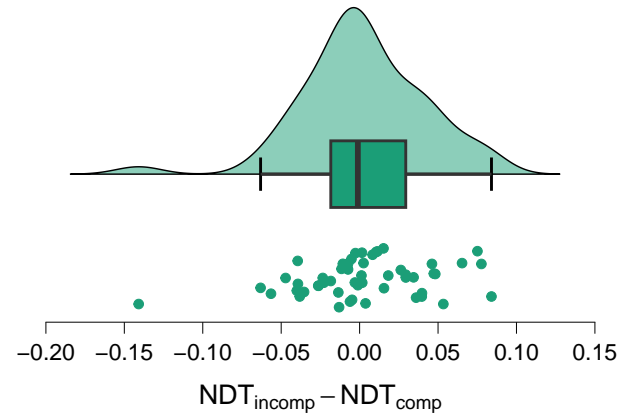


Figure 7: A raincloud difference plot illustrating the distribution of observed differences in nondecision time T_{er} between compatible and incompatible trials.

Discussion

In the present study, we performed a diffusion model decomposition of the response times observed in a two-digit number comparison task. As expected, we found a reliable increase in mean response times on unit-decade incompatible trials. Applying the EZ diffusion model to our observed response times and accuracies, we found that this increase in mean response time was driven completely by a large decrease in *drift rate* v . Remarkably, we found positive evidence for a null effect on nondecision time T_{er} . These results imply that the unit-decade compatibility effect is isolated completely to the decision component of the response times. Thus, it appears that the unit-decade compatibility effect is due to decision-level competition between parallel and partially active representations of the individual digits, not early perceptual processing. This gives initial support for a late-interaction account of the unit-decade compatibility effect, which is in line with a recent computational model for numerical cognition (Verguts et al., 2005; Gevers et al., 2006).

Acknowledgments

This research was supported by an NREUP grant from the Mathematical Association of America awarded to TJF.

References

- Cipora, K., Faulkenberry, T. J., Bahnmueller, J., Connolly, H., Bowman, K., Moeller, K., & Nuerk, H.-C. (2022). *Prevalence of cognitive phenomena - comparison of four methods*. OSF. Retrieved from osf.io/9y6m4
- Dehaene, S., Dupoux, E., & Mehler, J. (1990). Is numerical comparison digital? Analogical and symbolic effects in two-digit number comparison. *Journal of Experimental Psychology: Human Perception and Performance*, 16(3), 626–641. doi: 10.1037/0096-1523.16.3.626
- Erb, C. D., Moher, J., Song, J.-H., & Sobel, D. M. (2018). Numerical cognition in action: Reaching be-

- havior reveals numerical distance effects in 5- to 6-year-olds. *Journal of Numerical Cognition*, 4(2), 286–296. doi: 10.5964/jnc.v4i2.122
- Faulkenberry, T. J. (2014). Hand movements reflect competitive processing in numerical cognition. *Canadian Journal of Experimental Psychology*, 68(3), 147–151. doi: 10.1037/cep0000021
- Faulkenberry, T. J. (2016). Testing a direct mapping versus competition account of response dynamics in number comparison. *Journal of Cognitive Psychology*. doi: 10.1080/20445911.2016.1191504
- Faulkenberry, T. J., Cruise, A., Lavro, D., & Shaki, S. (2016). Response trajectories capture the continuous dynamics of the size congruity effect. *Acta Psychologica*, 163, 114–123. doi: 10.1016/j.actpsy.2015.11.010
- Faulkenberry, T. J., Cruise, A., & Shaki, S. (2017). Reversing the manual digit bias in two-digit number comparison. *Experimental Psychology*, 64(3), 191–204. doi: 10.1027/1618-3169/a000365
- Faulkenberry, T. J., Cruise, A., & Shaki, S. (2018). Task instructions modulate unit–decade binding in two-digit number representation. *Psychological Research*, 84(2), 424–439. doi: 10.1007/s00426-018-1057-9
- Faulkenberry, T. J., Ly, A., & Wagenmakers, E.-J. (2020). Bayesian inference in numerical cognition: A tutorial using JASP. *Journal of Numerical Cognition*, 6(2), 231–259. doi: 10.5964/jnc.v6i2.288
- Faulkenberry, T. J., Montgomery, S. A., & Tennes, S.-A. N. (2015). Response trajectories reveal the temporal dynamics of fraction representations. *Acta Psychologica*, 159, 100–107. doi: 10.1016/j.actpsy.2015.05.013
- Gevers, W., Verguts, T., Reynvoet, B., Caessens, B., & Fias, W. (2006). Numbers and space: A computational model of the SNARC effect. *Journal of Experimental Psychology: Human Perception and Performance*, 32(1), 32–44. doi: 10.1037/0096-1523.32.1.32
- Hinrichs, J. V., Yurko, D. S., & Hu, J. (1981). Two-digit number comparison: Use of place information. *Journal of Experimental Psychology: Human Perception and Performance*, 7(4), 890–901. doi: 10.1037/0096-1523.7.4.890
- JASP Team. (2023). *JASP (Version 0.17.1)[Computer software]*. Retrieved from <https://jasp-stats.org/>
- Macizo, P., & Herrera, A. (2011). Cognitive control in number processing: Evidence from the unit–decade compatibility effect. *Acta Psychologica*, 136(1), 112–118. doi: 10.1016/j.actpsy.2010.10.008
- Moeller, K., Nuerk, H.-C., & Willmes, K. (2009). Internal number magnitude representation is not holistic, either. *European Journal of Cognitive Psychology*, 21(5), 672–685. doi: 10.1080/09541440802311899
- Moyer, R. S., & Landauer, T. K. (1967). Time required for judgements of numerical inequality. *Nature*, 215(5109), 1519–1520. doi: 10.1038/2151519a0
- Nuerk, H.-C., Weger, U., & Willmes, K. (2001). Decade breaks in the mental number line? Putting the tens and units back in different bins. *Cognition*, 82(1), B25–B33. doi: 10.1016/s0010-0277(01)00142-1
- R Core Team. (2023). *R: A language and environment for statistical computing [Computer software manual]*. Vienna, Austria. Retrieved from <https://www.R-project.org/>
- Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, 85(2), 59–108. doi: 10.1037/0033-295x.85.2.59
- Ratcliff, R., Smith, P. L., Brown, S. D., & McKoon, G. (2016). Diffusion decision model: Current issues and history. *Trends in Cognitive Sciences*, 20(4), 260–281. doi: 10.1016/j.tics.2016.01.007
- Reynvoet, B., Notebaert, K., & Van den Bussche, E. (2011). The processing of two-digit numbers depends on task instructions. *Zeitschrift für Psychologie*, 219(1), 37–41. doi: 10.1027/2151-2604/a000044
- Rouder, J. N., Speckman, P. L., Sun, D., Morey, R. D., & Iverson, G. (2009). Bayesian t tests for accepting and rejecting the null hypothesis. *Psychonomic Bulletin & Review*, 16(2), 225–237. doi: 10.3758/pbr.16.2.225
- Sobel, K. V., Puri, A. M., & Faulkenberry, T. J. (2016, Apr). Bottom-up and top-down attentional contributions to the size congruity effect. *Attention, Perception, & Psychophysics*, 78(5), 1324–1336. doi: 10.3758/s13414-016-1098-3
- Sobel, K. V., Puri, A. M., Faulkenberry, T. J., & Dague, T. D. (2017). Visual search for conjunctions of physical and numerical size shows that they are processed independently. *Journal of Experimental Psychology: Human Perception and Performance*, 43(3), 444–453. doi: 10.1037/xhp0000323
- Verguts, T., & De Moor, W. (2005). Two-digit comparison: Decomposed, holistic, or hybrid? *Experimental Psychology*, 52(3), 195–200. doi: 10.1027/1618-3169.52.3.195
- Verguts, T., Fias, W., & Stevens, M. (2005, feb). A model of exact small-number representation. *Psychonomic Bulletin & Review*, 12(1), 66–80. doi: 10.3758/bf03196349
- Wabersich, D., & Vandekerckhove, J. (2014). The RWiener package: an r package providing distribution functions for the wiener diffusion model. *The R Journal*, 6(1), 49. doi: 10.32614/rj-2014-005
- Wagenmakers, E.-J. (2007). A practical solution to the pervasive problems of p values. *Psychonomic Bulletin & Review*, 14(5), 779–804. doi: 10.3758/bf03194105
- Wagenmakers, E.-J., Van Der Maas, H., & Grasman, R. P. P. P. (2007). An EZ-diffusion model for response time and accuracy. *Psychonomic Bulletin & Review*, 14(1), 3–22. doi: 10.3758/bf03194023