# Proceedings of ICCM 2022

20<sup>th</sup> International Conference on Cognitive Modelling<sup>1</sup>

Edited by

Terrence C. Stewart

<sup>1</sup> Co-located with the 55<sup>th</sup> Annual Meeting of the Society for Mathematical Psychology and held both online and in Toronto, Canada

# 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 but not limited to connectionism, symbolic modelling, dynamical systems, Bayesian modelling, and cognitive architectures. Research topics can range from low-level perception to high-level reasoning. In 2022, ICCM was jointly held with MathPsych, the annual meeting of the Society for Mathematical Psychology. Both events were held in a hybrid manner; first there was a purely online event running from July 11 to July 15, 2022, and then there was an inperson event held in Toronto, Canada from July 23 to July 27.

# Acknowledgements

We would like to thank the Society for Mathematical Psychology (SMP) for their tremendous work in developing tools to allow these conferences to function in an online venue. In particular, the MathPsych Conference Chair (Joachim Vandekerckhove) guided the development of a custom 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. (2022). Title of the paper. In Stewart, T. C. (Ed.). *Proceedings of the 20th International Conference on Cognitive Modelling* (pp. 6-12). University Park, PA: Applied Cognitive Science Lab, Penn State.

ISBN-13: 978-0-9985082-6-9, published by the Applied Cognitive Science Lab, Penn State.

(C) Copyright 2022 retained by the authors

Proceedings of the 20th International Conference on Cognitive Modelling (ICCM 2022)

# **Conference Committees**

# **General and Program Chair**

Terrence C. Stewart

National Research Council Canada

# **Program Committee**

Madeleine Bartlett **Edward Cranford** Michael Furlong Theodros Haile Alex Kelly Jong Wook Kim Othalia Larue Peter Lindes Miki Matsumuro **David Peebles** Sarah Ricupero Frank Ritter Dario Salvucci Robert St. Amant James Chapman Treyens Natarajan Vaidyanathan Marieke van Vugt Robert L. West Yuxue Yang Martin K.-C. Yeh

University of Waterloo **Carnegie Mellon University** University of Waterloo University of Washington **Carleton University OpenAI** Wright State University University of Michigan **Ritsumeikan University** University of Huddersfield Pennsylvania State University Pennsylvania State University **Drexel University** US Army Research Lab University of Washington University of Waterloo University of Groningen **Carleton University** University of Washington Pennsylvania State University

# Table of Contents

Comparing Cognitive, Cognitive Instance-Based, and Reinforcement Learning Models in an Interactive Task
Response time modeling provides stable and mechanistically interpretable measures of individual differences in behavioral pattern separation
Biologically-Plausible Memory for Continuous-Time Reinforcement Learning
Fast Online Reinforcement Learning with Biologically-Based State Representations20 Madeleine Bartlett, Terrence C. Stewart, Jeff Orchard
Towards a Method for Evaluating Convergence Across Modeling Frameworks27 Lorraine Borghetti, Christopher R. Fisher, Joseph W. Houpt, Leslie M. Blaha, Glenn Gunzelmann, Christopher Stevens
Leveraging cognitive models for the wisdom of crowds in sequential decision tasks
Clarifying system 1 & 2 through the common model of cognition40 Brendan Conway-Smith, Robert L. West
Combining machine learning and cognitive models for adaptive phishing training46 Edward A. Cranford, Shahin Jabbari, Han-Ching Ou, Milind Tambe, Cleotilde Gonzalez, Christian Lebiere
Gamma power as an index of sustained attention in simulated vigilance tasks
Modeling short-term fatigue decrements in the successive/simultaneous discrimination task59 <i>Taylor Curley, Megan B. Morris</i>
Using a Cognitive Architecture to consider antiblackness in design and development of AI systems
Argumentation-based Reasoning guided by Chunk Activation in ACT-R73 <i>Emmanuelle Dietz</i>
Modeling of Multi-Defender Collaboration in a Cyber-Security Scenario81

Yinuo Du, Palvi Aggarwal, Kuldeep Singh, Cleotilde Gonzalez

Combining EEG and a cognitive model to infer the time course of game play
On the limits of spreading activation in ACT-R: Predictions and testability
A comparison of quantum and multinomial processing tree models of the interference effect
Informational Trade-offs of Learning from Expert Demonstration
Action sequencing, timing, and chunking in Space Fortress109 Pierre G. Gianferrara, John R. Anderson
Estimating ACT-R Declarative Memory Parameters Using a Drift Diffusion Model112 <i>Gillian Grennan, Andrea Stocco</i>
Learning to expect change: Volatility during early experience alters reward expectations in a model of interval timing119 <i>Nora C. Harhen, Aaron M. Bornstein</i>
Multitasking while driving: central bottleneck or problem state interference?125 Moritz Held, Jochem Rieger, Jelmer Borst
Obstacles to the skill-based approach: Why is skill reuse so difficult for cognitive architectures?
Individual Differences and Levels of Analysis in Computational Models of Coordination134 <i>Alexander R. Hough, Ion Juvina</i>
A quantum walk framework for multialternative decision making141 Jiaqi Huang, Jerome Busemeyer
Using GOMS to Model Individual Differences in a Competence Assessment Task143 Hadeel Ismail, Peter C-H. Cheng
Cognitive and Motivational Effects in Peer-Assisted Learning

Proceedings of the 20th International Conference on Cognitive Modelling (ICCM 2022)

Understanding adversarial decisions for different probing-action costs in a deception game via cognitive modeling
Reverse-Engineering of Boolean Concepts: A Benchmark Analysis
A computational cognitive theory of temporal reasoning
Evolving understandable cognitive models
Towards a computational model of a dynamic feeling of knowing183 <i>Justin Li, Bryce Boyle</i>
Do Models of Syllogistic Reasoning extend to Generalized Quantifiers?
Modeling optimal arousal by integrating basic cognitive components
Estimating phonological awareness with interactive cognitive models: Feasibility study manipulating participants' auditory characteristics
Minerva-Q: A multiple-trace memory system for reinforcement learning
Modeling prominence constraints for German pronouns as weighted retrieval cues216 Umesh Patil, Petra B. Schumacher
Predicting Learning in the Troubleshooting Task using a Cognitive Architecture-Based Task Analysis
Discontinuities in function learning224 Roderick Yang Terng Seow, John R. Anderson
Exploring Multitasking Strategies in an ACT-R model of a Complex Piloting Task226 Garrett Swan, Christopher A. Stevens, Christopher R. Fisher, Samantha Klosterman
Specificity of the Jumping-to-Conclusion Bias in Social Anxiety: An Account using the Bayesian Computational Modelling Approach232 Nicole Tan, Yiyun Shou, Junwen Chen, Bruce Christensen
Predicting algorithmic complexity for individuals240

Sara Todorovikj, Daniel Brand, Marco Ragni

Learning linguistic reference biases in the PRIMs cognitive architecture
Using deep neural networks for modeling representational spaces: the prevalence and impact of rarely-firing nodes
Modelling Expert and Novice Programming Strategies using Python-ACTR251 Maria Vorobeva, Robert West, Kasia Muldner
Beyond Responding Fast or Slow: Improving Cognitive Models of Memory Retrieval using Prosodic Speech Features258 Thomas Wilschut, Florian Sense, Odette Scharenborg, Hedderik van Rijn
A Model of Motivation and Effort Allocation in the ACT-R Cognitive Architecture261 <i>Yuxue C. Yang, Andrea Stocco</i>

# Comparing Cognitive, Cognitive Instance-Based, and Reinforcement Learning Models in an Interactive Task

#### Amir Bagherzadehkhorasani (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, Technology and Professional Programs, Penn State, University Park, PA 16802 USA

#### Abstract

This work tries to answer fundamental questions of learning bias in cognitive models, how decision-making strategies in different cognitive models vary and why. Using a biased coin in a coin flip game, we study the number of trials it takes for each cognitive model to learn the asymmetry in the coin. Also, we investigate how the model behaves knowing the asymmetry. A web-based game is designed to simulate coin flipping to collect the models' data. The most common approaches to model the decision-making process are used for this study. Cognitive architectures such as ACT-R and PyIBL with the capability of learning and making decisions are used and compared. Also, we consider Reinforcement Learning with different decision-making strategies such as Thompson Sampling, Boltzmann Exploration, and Epsilon Greedy algorithm. All developed models interact with the task environment and complete the task. To facilitate the interaction between the models and the game's interface, we developed a new tool called VisiTor. VisiTor grants cognitive models the ability to gain information and execute actions in dynamic environments. The results show models are capable of replicating human's main decision-making strategies: matching and maximizing.

**Keywords:** ACT-R; cognitive modeling; reinforcement learning; instance-based learning; binary choice experiments; decision-making

#### Introduction

The most commonly used method to study human decisionmaking procedures consists of observing human performance in a choice task and proceeding with developing a cognitive model. These models emulate human behavior (Cassimatis, Bello, & Langley, 2008); Erev et al. (2005) discussed the learning process with immediate feedback, which consists of different processes such as the tradeoffs of adaptation and maximization in repeated choice tasks. They proposed a Reinforcement Learning model alongside the cognitive strategies to consider the payoff variability and other deviations. Janssen et al. (2012) also utilized ACT-R to study the effect of the reward value. They suggested a new approach to determine the reward that is experienced in the environment. Lebiere et al. (2007) used Instance-Based Learning (IBL) to demonstrate that a binary choice problem with immediate feedback does not always lead to payoff maximization. One of the factors that limits the studies to explore more complicated choice tasks is the restricted cognitive models' capability to interact with task environments. ACT-R is a hybrid cognitive architecture that is consisted of a set of programmable information-processing mechanisms. These mechanisms are used to predict and explain human behavior, including cognition and interaction with the environment (J. R. Anderson et al., 2004; Ritter, Tehranchi, & Oury, 2018; Tehranchi & Ritter, 2018a). Ever since the emergence of ACR-R in 1998 (John R Anderson, Bothell, Lebiere, & Matessa, 1998), several researchers have utilized ACT-R capabilities to simulate human interaction and cognition while performing a specific task (Cao, Ho, & He, 2018; Gray, Schoelles, & Sims, 2005; Hope, Schoelles, & Gray, 2014). ACT-R models typically interact with the world through ACT-R's device interface, an abstract representation of the world based on a simulated Lisp environment provided with ACT-R or by instrumenting interfaces. However, these interactions are limited to being applied to an unmodified ACT-R environment in special windows provided by ACT-R. In other words, if the environment that a model is interacting with is subject to change, the model will not be able to work properly. PyIBL is a Python implementation of a subset of Instance-Based Learning Theory (Gonzalez, Lerch, & Lebiere, 2003). PyIBL does not have a built-in capability to interact with any environment.

Inspired by JSegMan, SegMan, and ACT-CV (Halbrügge, 2013; St. Amant, Riedel, Ritter, & Reifers, 2005; Tehranchi & Ritter, 2018b), we developed VisiTor (Vision+Motor) that generates the required interactions in dynamic task environments. VisiTor simulates users' visual attention (vision) and use of a mouse and keyboard (Motor). This tool allows ACT-R and PyIBL to interact with any environment while keeping the operations similar to users as close as possible and its capabilities are expandable.

# Probability Learning and Decision-Making in Psychology Literature

Unknown bias effects on decision-making and prediction of the next outcome using a binary choice prediction task have been studied before. Bilda, Gero, and Sun (2006) conducted a simulation modeling bias for a pitch in baseball. Altmann and Burns (2005) studied the effect of streaks in coin flips on the prediction of the next toss. In binary choice experiments, participants are asked on each trial to predict the outcome of an event such as a coin flip. The outcome is usually biased towards one of the choices, and participants are not informed of the bias. Altmann et al. (2005) claimed that participants tend to adapt their behavior to the relative reward accordingly instead of maximizing the expected reward. In another word, they try to "match" rather than "maximize." In matching, participants' choices would reflect bias in the coin, while in maximizing, the participants would maximize the reward by choosing the option with a higher probability of success. Assume in a coin flip game that the ratio of head and tail is 3 to 1. While matching, Participants predict heads on roughly 75% of trials by the end of a session. Whereas in the optimal strategy, one should choose head 100 percent of the trials to maximize the number of wins, once they detect the bias. This aligns with (Vulkan, 2000) results. Such behavior is paradoxical because matching results in less reward receipt than maximizing. This is because participants cannot know when a given location or response option will be rewarded, even if they are aware of the overall reward rate.

The effects of age in the strategy taken after learning the bias has been a subject of conflict among Probability Learning studies. The ratio of school-age children demonstrating matching strategy is similar to the ratio of adults using the matching strategy (Brackbill & Bravos, 1962; Derks & Paclisanu, 1967). Also, Younger children (ages 3-5 years) demonstrate maximizing more than older children (Plate, Fulvio, Shutts, Green, & Pollak, 2018). While Moran III and McCullers (1979) have found that adults maximize rewards more effectively than children. Recently, Plate et al. (2018) conducted a comprehensive study on adults and children. They compared their results to 4 different decision-making models: Random model, Matching model, Maximizing model and a combination of the last two (Combination model). Most adults and children's results matched the Combination model based on their study, suggesting participants exhibited matching behavior at the outset of the experiment and then crossed over to maximizing in the experiment. All participants who did not crossover from matching to maximizing were the best fit by the probability Matching model and are sensitive to the underlying probabilities. In summary, all researchers agree that people can identify the bias if the bias is significant enough. However, how they react to the bias is still a subject of discussion.

#### Probability Learning and Decision-Making in Artificial Intelligence Literature

Probability Learning and decision-making models are not following the same strategy when taking an action. Most of Reinforcement Learning models learn the outcome distribution of each action by using posterior distribution over the outcome of each action. These models seek to find the best possible action for each scenario (Zhu, 2018). On the other hand, cognitive models do not necessarily look for the best action. Instead, they try to simulate human behaviors in the same scenario, regardless of the optimality of choice (Lebiere et al., 2007). In line with psychology literature, cognitive models have different strategies for decisionmaking. Reinforcement Learning models and cognitive models are capable of imitating both matching and maximizing decision-making strategies. Agents, developed using Reinforcement Learning (RL), interact with a task environment and generally learn to maximize their rewards (Sutton & Barto, 2018). These agents discover which actions to take to generate the highest possible rewards.

The Reinforcement Learning model discovers the right set of actions to take by trial and error. By observing the result after each instance, the model learns the outcome distribution of the actions. There are two important components in learning the outcome distribution of each choice: (a) how to update the outcome distribution based on the action taken, and (b) what action to take. The reward function is the cornerstone of the learning aspect of RL models. It maps each action to the outcome. The environment's characteristics, such as the delay between taking an action and observing the outcome and the possible outcome distributions, can affect how the reward function is defined (Guo, 2017). Deep Reinforcement Learning models replace the reward function with a Neural Network and let the model determine the best reward function (Li, 2017).

Decision-making strategies such as the Greedy algorithm results in maximizing, Boltzmann Exploration results in matching, and Thompson Sampling (Thompson, 1933) results in the combination of matching and maximizing behavior.

Due to this limitation of greedy algorithms, several methods have been developed to add exploration through randomly perturbing actions that a greedy algorithm would select (Dabney, Ostrovski, & Barreto, 2020; Masadeh, Wang, & Kamal, 2018; Tokic, 2010). These methods are called *Dithering*. The most basic Dithering method is called *Epsilon Greedy Exploration*. This method applies the greedy action with probability  $1 - \varepsilon$  and otherwise selects an action uniformly at random. Although this type of exploration improves the performance of the greedy algorithm, it wastes resources by trying all the actions, even those that are unlikely to generate a better reward than what we already have. For example, half of the exploration is wasted by trying action 2. This issue gets worse as the number of actions increases.

Thompson Sampling was introduced more than 80 years ago (Thompson, 1933). This method provides an alternative to dithering that more intelligently allocates the exploration effort. In this method, a Beta distribution with ( $\alpha = 1, \beta = 1$ ) is initially assumed for each action. At each instance, we sample from each action's distribution. Whichever action gives us the largest sample value will be chosen. After the action is taken and the result is observed. If it is a success,  $\alpha$  is increased by one. Otherwise,  $\beta$  is increased by one. This process will be repeated each time an action needs to be taken.

Boltzmann Exploration utilizes a similar strategy of decision-making to ACT-R. The actions are taken stochastically. Initially, the reward for all actions is assumed to be equal. At each trial, the probability of taking an action i, is calculated as follows:

$$P_i = \frac{e^{U_i/T}}{\sum_{i \in m} e^{U_i/T}}$$

where *m* is the set of all actions. The action is going to be taken based on a discrete distribution with probabilities calculated from this equation. After each trial, the expected rewards for all actions are updated. This equation indicates that as the chance of actions being taken is proportional to the values of  $U_i/T$ .

The parameter T is known as temperature. It controls the randomness of the action. The higher the value of T, the more randomness happens in action selection.

#### Methodology

In this study, we considered a simple coin flip game. Every round, participants and models choose either head or tail. If their choice matches the game's choice, the result is winning the round and a message "Match" will show up. If the choices do not match, the result is losing that round and a message "Wrong" will show up. The probabilities of the computer choosing head or tail are not equal. In 70 percent of the occurrences, head will appear, and the tail will appear in 30 percent of the trials. This game is an online browser game and is written in C# and was first used by (Tehranchi & Ritter, 2020) to study the number of trials needed for ACT-R to match the probability of the biased coin. Figure 1 shows a screenshot of the game when a user starts playing. The models' data of interactions with the game and their final decisions are saved.



Figure 1: The game environment. It consists of a feedback area (where the "Ready" sign is shown) and visual objects (Head and Tail buttons). The feedback area will be changed based on the result of the played round. It shows "Match" if the choice is correct, and it shows "Wrong" if the choice is incorrect.

The Coin flip game is played in an interactive environment. In this environment, the interface is susceptible to change if the user provides input. Every time a model chooses what to play for that round and clicks on the buttons. The feedback area is going to change. Also, the model continues to work even if the location of the environment window is moved. And because of that, the visual module implemented in ACT-R cannot be used for this task.

#### The ACT-R Model

ACT-R's actions that play the game can be broken down into several subtasks. Each subtask consists of some production rules that ACT-R uses to play the Coin flip game. Every visual object on display will be represented by a set of unique features for the visual module. Chucks are created by these features that provide declarative memory, the representation of the visual scene by the vision module. These chucks are visual location and visual object types. Production rules' constraints can match the chunks. The model consists of 3 visual objects, "Ready", "Heads", and "Tail", in addition to 15 declarative memory chunks and 12 production rules. Production rules execute shifting attention, finding the ready icon to start, choosing heads or tail, clicking, finding the visual feedback, and assigning reward value to the results. The reward value for matching is 6 and in case of wrong, the reward value of zero is assigned. The Utility Value and the sub-symbolic computations parameter are set to true. The value of temperature varies based on the decision-making strategy we are trying to replicate. All the other parameters are set to default.

#### Starting the Game

The model looks for the visual object "ready". If it finds it, the model is ready to choose an action. At this point, the reward (Utility) function for all the choices is equal.

#### **Taking an Action**

Based on the reward function, the model will take an action, retrieves the visual object corresponding to that action, moves the cursor to the visual location, and clicks. Each action has a probability of being taken. The probability for action i is calculated using the Boltzmann Equation:

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

Where the summation j is over all the productions which currently meet the conditions required. ACT-R multiplies the temperature value (T in Boltzmann Equation) by the square root of two.

After the model decides what action to take, the model needs to find the visual object corresponding to that action and select (click) it. For this task, VisiTor first finds the location of the visual objects on the screen by Template Matching capability of the OpenCV Python library. The templates are predefined and saved as an image. Then, VisiTor will save that visual object as an image. In order to assure than VisiTor is robust to rescaling and size, different sizes of the template are checked. Then, VisiTor moves the mouse to the location of that object and clicks.

#### Looking for Feedback

After taking an action, the model expects feedback. The model tries to find which of the feedback visual objects is shown on the screen. The model first retrieves them into the memory module and then utilizes VisiTor. VisiTor search for the feedback visual object that is appearing on the screen.

#### **Updating the Reward Function**

Based on the feedback, the model updates the reward function for the action taken in the last step.

$$U_{i}(n) = U_{i}(n-1) + \alpha [R_{i}(n) - U_{i}(n-1)]$$

Where:

- $\alpha$  is the learning parameter
- $R_i(n)$  is the effective reward value given to production i for its n<sup>th</sup> usage
- $U_i(0)$  is the initial utility value for production i

Then the model goes back to the *Taking an action* section and repeats the whole process.

Similar to ACT-R, RL models follow the same set of actions to play the game. The only difference is how the reward function is updated and the decision-making strategy. Here, we tried Epsilon Greedy, Boltzmann Exploration, Thompson Sampling, and PyIBL to analyze the differences. In this section, we will elaborate on the decision-making process of each model and what decision-making strategy they utilize in the coin flip game.

#### **PyIBL Model**

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 activation, base level activation, weights, utilities, noise, and temperature.

#### Activation

A fundamental part of retrieving an instance from the PyIBL model's memory is computing the activation of that instance. The value of the activation is based on (a) how frequently and recently it has been experienced by the model and (b) how well it matches the attributes of what is to be retrieved. The activation is calculated based on the following formula:

Where:

 $A_i = B_i + \epsilon_i$ 

- $A_i$ : Activation of chunk *i*. It is also called "match score"  $M_i$
- *B<sub>i</sub>*: 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
- $\epsilon_i$ : A noise value

#### **Base Level**

The base-level activation,  $B_i$ , describes the frequency and recency of the chunk i. Its value depends upon the decay parameter of Memory, d. The base level activation is computed using the amount of time that has elapsed since each of the past appearances of i, which in the following are denoted as the various  $t_{ij}$ .

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

#### **Activation Noise**

The activation noise,  $\epsilon_i$ , 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.

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

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

$$BV = \sum_{i \in m} \frac{w_i}{\sum_{j \in m} w_j} u_i$$

Lastly, the action with the largest blending value is taken. If the outcome is already represented by a chunk, the base level activation will be updated. If not, a chunk will be created to represent the outcome in the next blending equation.

#### **Deep Reinforcement Learning**

First, a Neural Network predicts the outcome of each action based on the instances the model has seen so far. After each trial, the outcome is observed. Using the observed outcome, the model tries to tune the Neural Network parameters to predict the outcome more accurately. The loss function for the Reinforcement Learning model is defined as follows:

$$\mathbf{L} = \mathbf{E}[(\mathbf{U}(\mathbf{s},\mathbf{a};\,\boldsymbol{\theta}_k) - \mathbf{U}(\mathbf{s},\mathbf{a}))^2]$$

Where the first term is the Neural Network predicted reward function and the second term is the actual reward observed by the model.  $\theta$  represents the Neural Network parameter.

To take an action, the model predicts the reward value for all actions. The reward values are important for all decision-making strategies. Different actions might be taken depending on what decision-making strategy is used. Figures 2 and 3 show the flow chart of how Epsilon Greedy and Thompson Sampling play the coin flip game.

#### Results

The reward value that is assigned to match or wrong visual objects is an important factor in models' convergence. In case of a small difference between the reward of a match and wrong, all models fail to learn bias and fail to show any decision-making strategy. With a proper choice of reward value, all the models show that they are capable of learning the bias in less than 200 trials. Both ACT-R and PyIBL are capable of implementing matching and maximizing decisionmaking strategies. Figure 4 shows the effect of temperature on the decision-making strategy of the PyIBL model. For figure 4.a, the temperature value was set to one and for figure 4.b, the temperature was set to 14. For small values of temperature, the PyIBL model will choose the maximizing strategy. As the value of temperature increases, the decisionmaking strategy move towards matching. If the value of the temperature is set too high, the PyIBL agent will decide completely random (i.e., 50 percent of the time, the PyIBL model chooses head, and 50 percent of the time, it chooses tail).

Figure 5 shows the effect of temperature on the decisionmaking strategy of the ACT-R model. Figure 5.a displays the proportion of head and tail chosen by ACT-R if the temperature is set to 3. In Figure 5.b, the temperature was set to 0.5.



Figure 2: The flowchart for the Epsilon Greedy algorithm contains five processes and one conditional operation. In each step, with the probability of Epsilon, the model takes a random action. Otherwise, it will select the action with the highest predicted utility value.



Figure 3: Thompson Sampling Flowchart contains four processes. The beta distribution is assigned to each choice. At the beginning of the training, all parameters are equal to 1. Meaning that the model assumes that parameters are all likely to generate an optimum result. At each step, samples are taken from each action distribution. The biggest sample determines what action should be taken. Then based on the

result, the distribution of the action taken is updated.

Similar to PyIBL, smaller values of temperature will result in maximization and as the value of temperature increases, the randomness of choices will increase. ACT-R shows more sensitivity to the value of the temperature in comparison to PyIBL. Meaning smaller changes in the value of temperature in ACT-R will result in more noticeable shifts in decision-making strategy.

Figure 6 shows the Epsilon Greedy maximizes the utility by only taking actions with the highest reward. This is exactly what is expected from this model. The Deep Reinforcement Learning with Epsilon Greedy decision-making strategy is designed to maximize the reward. The experiment shows that the maximizing behavior of the model has started after the fourth trial (where the reward value of the head became larger than the tail). A bad sequence of random occurrences might result in the model taking the wrong action as the maximizing action and may not be able to recover.



Figure 4: Probability of choosing Head (Blue) and Tail (Orange) over 200 trials by PyIBL in the case of (a) maximizing and (b) matching with different temperatures.



Figure 5: Probability of choosing Head (Blue) and Tail (Orange) over 200 trials by ACT-R in the case of (a) maximizing and matching (b) using different temperatures.

Figure 7 suggests that Thompson Sampling started with matching and then maximized after gaining confidence that the estimated reward value of the head is larger than the tail. Figure 8 shows the decision-making by Deep Reinforcement Learning with Boltzmann Exploration. With the right value of temperature, this model can imitate both matching and maximizing behaviors. In summary, all the models that utilized the Boltzmann Equation in their action taking (decision-making) strategy (ACT-R, PyIBL and Deep Reinforcement Learning), are capable of both matching and maximizing. Epsilon Greedy decision-making strategy always results in maximizing. Thompson Sampling first matches the probability of the coin and when it is confident in the reward of the head is greater than the tail, it starts to maximize the reward by choosing heads.



Figure 6: Probability of choosing Head (Blue) and Tail (Orange) over 200 trials by Deep Reinforcement Learning with Epsilon Greedy decision-making strategy.





#### Conclusion

We analyzed different models such as ACT-R, PyIBL, Reinforcement Learning with the Epsilon Greedy algorithm, Boltzmann Exploration, and Thompson Sampling decisionmaking strategies. We studied the models' capabilities to learn the bias and how they take an action. A web-based biased coin flip game was developed where models can interact and predict the next coin flip's result.

The outcome of the coin flip game (i.e., match or wrong visual objects) will be shown in the game environment. We introduced VisiTor, a Python-based tool that can facilitate the interaction between different models and task environments in any programming languages.

The models utilized two main strategies to choose what action to take. They can "Maximize," meaning they can select the choice they believe has the highest probability of success and maximize their outcome. Or they "Match" the probability of the actions. We showed that among the well-known cognitive architectures and algorithms, ACT-R, PyIBL and Deep Reinforcement Learning with Boltzmann Exploration are capable of imitating all decision-making strategies by setting the right set of parameters.



Figure 8: Probability of choosing Head (Blue) and Tail (Orange) over 200 trials by Deep Reinforcement Learning with Boltzmann Exploration.

Epsilon Greedy and Thompson Sampling tend to "Maximize" before the 200<sup>th</sup> trial. However, Thompson Sampling tends to "Match" at the beginning and then it will "Maximize" the reward. The behavioral studies in this area believe people are using the same set of strategies. However, which strategy is used in what situations is still a topic of conflict. A more systematic study needs to be conducted to show under what circumstances people tend to minimize or maximize. Based on the result, we will be able to see which model can simulate human behavior and under what circumstances.

#### **Future Works**

In order to determine which model is behaving closest to humans, a study needs to be conducted to analyze human behavior. Models that utilize the Boltzmann Equation in their decision-making strategy, can be tuned to Match or Maximize. Behavioral studies in this area indicate mixed results and may vary case by case. A systematic review is needed in this area to categorize the results and analyze the reason behind the mixed results that are reported by the studies previously done to analyze human behavior. This experiment needs to be conducted to determine if humans tend to match, maximize, or combination of both.

Also, currently, visual objects need to be predefined for VisiTor. A possible extension for VisiTor is to further extend its capabilities by having the model define the visual objects based on the human eye movement data. Users tend to pay closer attention to the visual objects they utilize to play. In the next version of VisiTor, we plan to have the model detect visual objects based on the eye-tracking data.

#### Acknowledgments

The Pennsylvania State University supports this work. We would like to thank Dan Bothell, Don Morrison, and Genevieve Gordon for their assistance.

#### References

- Altmann, E. M., & Burns, B. D. (2005). Streak biases in decision making: Data and a memory model. *Cognitive Systems Research*, 6(1), 5-16.
- 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.
- Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language*, 38(4), 341-380.
- Brackbill, Y., & Bravos, A. (1962). Supplementary report: The utility of correctly predicting infrequent events. *Journal of Experimental Psychology*, 64(6), 648.
- Cao, S., Ho, A., & He, J. (2018). Modeling and predicting mobile phone touchscreen transcription typing using an integrated cognitive architecture. *International Journal* of Human–Computer Interaction, 34(6), 544-556.
- Cassimatis, N. L., Bello, P., & Langley, P. (2008). Ability, breadth, and parsimony in computational models of higher-order cognition. *Cognitive Science*, *32*(8), 1304-1322.
- Dabney, W., Ostrovski, G., & Barreto, A. (2020). Temporally-extended {\epsilon}-greedy exploration. *arXiv preprint arXiv:2006.01782*.
- Derks, P. L., & Paclisanu, M. I. (1967). Simple strategies in binary prediction by children and adults. *Journal of Experimental Psychology*, 73(2), 278.
- Erev, I., & Barron, G. (2005). On adaptation, maximization, and reinforcement learning among cognitive strategies. *Psychological Review*, *112*(4), 912.
- Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instancebased learning in dynamic decision making. *Cognitive Science*, 27(4), 591-635.
- Gray, W. D., Schoelles, M. J., & Sims, C. R. (2005). Adapting to the task environment: Explorations in expected value. *Cognitive Systems Research*, 6(1), 27-40.
- Guo, X. (2017). Deep learning and reward design for reinforcement learning.
- Halbrügge, M. (2013). ACT-CV: Bridging the gap between cognitive models and the outer world. *Grundlagen und Anwendungen der Mensch-Maschine-Interaktion*, 10, 205-210.
- Hope, R. M., Schoelles, M. J., & Gray, W. D. (2014). Simplifying the interaction between cognitive models and task environments with the JSON Network Interface. *Behavior Research Methods*, 46(4), 1007-1012.
- Janssen, C. P., & Gray, W. D. (2012). When, what, and how much to reward in reinforcement learning-based models of cognition. *Cognitive Science*, *36*(2), 333-358.
- 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., Gonzalez, C., & Martin, M. (2007). *Instance*based decision making model of repeated binary choice. Retrieved from
- Li, Y. (2017). Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*.
- Masadeh, A. e., Wang, Z., & Kamal, A. E. (2018). Convergence-based exploration algorithm for reinforcement learning. *Electrical and Computer Engineering Technical Reports and White Papers, 1.*
- Moran III, J. D., & McCullers, J. C. (1979). Reward and number of choices in children's probability learning: An attempt to reconcile conflicting findings. *Journal of Experimental Child Psychology*, 27(3), 527-532.
- Plate, R. C., Fulvio, J. M., Shutts, K., Green, C. S., & Pollak, S. D. (2018). Probability learning: Changes in behavior across time and development. *Child development*, 89(1), 205-218.
- Ritter, F. E., Tehranchi, F., & Oury, J. D. (2018). ACT-R: A cognitive architecture for modeling cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, e1488.
- St. Amant, R., Riedel, M. O., Ritter, F. E., & Reifers, A. (2005). *Image processing in cognitive models with* SegMan (Vol. 4). Mahwah, NJ: Erlbaum.
- Tehranchi, F., & Ritter, F. E. (2018a). Modeling visual search in interactive graphic interfaces: Adding visual pattern matching algorithms to ACT-R. Paper presented at the Proceedings of ICCM-2018-16th International Conference on Cognitive Modeling.
- Tehranchi, F., & Ritter, F. E. (2018b). Using Java to provide cognitive models with a more universal way to interact with graphical user interfaces. Paper presented at the 2018 International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction and Behavior Representation in Modeling and Simulation, BRiMS 2018.
- Tehranchi, F., & Ritter, F. E. (2020). *Extending JSegMan to Interact with a Biased Coin Task and a Spreadsheet Task.* Paper presented at the 17th International Conference on Cognitive Modeling, ICCM 2019.
- Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4), 285-294.
- Tokic, M. (2010). Adaptive ε-greedy exploration in reinforcement learning based on value differences. Paper presented at the Annual Conference on Artificial Intelligence.
- Vulkan, N. (2000). An economist's perspective on probability matching. *Journal of economic surveys*, 14(1), 101-118.
- Zhu, J. (2018). *Probabilistic Machine Learning: Models, Algorithms and a Programming Library.* Paper presented at the IJCAI.

### **Response Time Modeling Provides Stable and Mechanistically Interpretable Measures of Individual Differences in Behavioral Pattern Separation**

Nidhi V. Banavar (nbanavar@uci.edu) Aaron M. Bornstein (aaron.bornstein@uci.edu) Department of Cognitive Sciences, University of California, Irvine Irvine, CA 92617 USA

#### Abstract

The incredible specificity and fidelity of human memory encoding is thought to be supported by a process known as pattern separation (Marr, 1971). Behaviorally, this is typically inferred via performance in the Mnemonic Similarity Task (MST; (Stark, Kirwan, & Stark, 2019)), an object recog-nition task with added similar "lure" images, from which a key metric, the Lure Discrimination Index (LDI) is calculated. Supported by an extensive literature validating its predictive power, this measure is gaining increasing use as a diagnostic of cognitive decline and neurological dysfunction. It is however unclear the exact mechanism through which this behavioral measure of pattern separation reflects the underlying neural computations. In particular, choices alone cannot in principle distinguish the degree to which a given behavior results from signal-based discrimination of the object in question (i.e. the putative separated patterns) versus a more general tendency to inhibit or excite responses (e.g. response caution). Here, we distinguish these potentially co-contributing factors by modeling response times using a sequential sampling framework that identifies independent contributions to choices made by signalnoise discrimination and response thresholding. Across two independent datasets encompassing a lifespan sample (total N = 307, ages 8-89), we find evidence that both factors reliably contribute to response behavior, but that signal discrimination is both more strongly correlated with Lure and Foil discrimination and more stable within-individual than response thresholding, suggesting that this model-derived parameter may be a more specific and reliable measure of the underlying trait of interest in studies of pattern separation.

**Keywords:** memory and discrimination; evidence accumulation; recognition

#### Introduction

How do individuals encode objects in memory, and how does the distinctiveness of encoding affect behavioral expressions of recognition? These functions are thought to be supported by a process known as pattern separation, whereby similar sensory or latent input patterns are projected into higherdimensional space to create highly distinct patterns that support later discrimination among fine degrees of difference (Stark et al., 2019). Traditionally, this process has been attributed to the hippocampus, a critical brain structure for learning and memory (Long, Lee, & Kuhl, 2016; Marr, 1971; Stark et al., 2019). Computational models predict that the more distinct object representations are (i.e. the "better" an individual is at pattern separating), the better an individual will be able to *discriminate* between objects that were seen previously and those that weren't. In particular, people who are better at pattern separating should be less susceptible to interference when novel objects are similar to the previously seen objects.

The most widely used behavioral measure of pattern separation, known as the Lure Discrimination Index (LDI), stems from the 3AFC *Mnemonic Similarity Task* (MST), a modified object recognition task (Stark et al., 2019). In the typical version of this experiment, individuals first complete a learning phase where they study a collection of object pictures. Then, during the recognition phase, individuals see a series of objects of one of three types: *repeats*, or objects they had seen before during learning; *lures*, which vary in degrees of similarity to the repeats; and *foils*, which are totally new objects never seen before in the experiment. Thus responses on these three trials can be analyzed to quantify how sensitively an individual discriminates between what they have, and have not seen before. This measure, the LDI, has been shown to correlate with standard behavioral and physical measures of cognitive decline and neurological dysfunction (Stark et al., 2019).

It is however an open question as to what aspects of recognition memory behavior are measured by the LDI. Specifically, it is unclear to what degree LDI solely reflects the actual "separation" of the underlying memory representations (in Signal Detection Theory terms, the separation between signal and noise distributions), versus more general response selection processes (e.g. the threshold for response execution). To the extent that LDI is indeed a measure consistent with hippocampal pattern separation, we would predict the latter: that it would correspond with an increase in signal to noise ratio (Long et al., 2016).

Sequential sampling models of response time provide an excellent method to assess these separable influences on recognition memory. This family of models, specifically the Linear Ballistica Accumulator which we use in this paper, robustly distinguishes separable contributions to behavior of both signal-noise separation (as *drift rate*) and response execution (as *threshold/boundary* or *starting point*) (Brown & Heathcote, 2008).

Here, we model response times to examine the relationship between LDI and components of the recognition memory process. We find evidence for both processes contributing to measured LDI, examine their relative contributions to choices, and assess their ability to predict behavior out-ofsample. Our results support the suggestion that LDI can be decomposed to isolate a stable, separable signal-based measure of memory discrimination. This measure may further improve the reliability and precision with which clinical practitioners can assess a key transdiagnostic process underlying a wide array of disorders and neurological conditions.

#### Methods

#### **Data and Experiments**

We model two data sets of individuals that completed the Mnemonic Similarity Task (MST). In this task, participants initially completed an "encoding" phase where they categorized unique objects as either belonging indoors or outdoors. They were also told that they would be subsequently tested on their memory of these objects.

Then, participants made a sequence of recognition choices during the "test" phase where they identified each object as either a repeat (seen before during the encoding phase), lure (similar to an object seen during encoding), or foil (a brand new object). Participants saw  $\frac{1}{3}$  repeated objects,  $\frac{1}{3}$  lures, and  $\frac{1}{3}$  foils. There was no feedback after each choice (i.e. participants were not informed if their choice was accurate or not) and subjects had up to 10s to make a choice. The presentation order was fully randomized.

**Experiment 1** We model n = 223 adult subjects (ages 18 – 89, median = 41, 141 female). Subjects saw 128 trials during the encoding phase, and made 192 recognition judgements during the test phase. The data was collected in two modalities: online via Amazon mTurk (n = 173) and in person (n = 72).

**Experiment 2** We model n = 84 subjects (ages 8 - 25, median = 15, 53 female). Subjects saw 64 trials during the encoding phase, and made 96 recognition judgements during the test phase. The data was all collected online via Amazon mTurk. All participant ages in Experiment 2 were verified using photographs of government-issued identification cards.

#### **Choice Behavior Measures**

To quantify memory discriminability, we compute the Lure Discrimination Index (LDI) as in (Stark et al., 2019).

$$LDI = P(Lure Response | Lure Trial)$$

$$- P(Lure Response | Foil Trial)$$
(1)

The LDI provides a sensitive measure of how reliably an individual distinguishes object photographs that were seen during the encoding phase from similar 'lures' presented during the test phase. This measure is typically interpreted as robust in that the more distinctly an individual encodes a previously seen object, the less they will subject to interference from *both* similar lures and unrelated foils. We further compute an individual's Recognition Score (RS), which quantifies how well someone remembers previously seen objects:

$$RS = P(Repeat Response | Repeat Trial) - P(Repeat Response | Foil Trial)$$
(2)

#### **Response Time Modeling**

We model response times (RT) using a Linear Ballistic Accumulator model (LBA) (Brown & Heathcote, 2008). The LBA is a powerful sequential sampling model that differs from other sequential sampling models in the following critical ways: a) it can fit *n* responses (*n*AFC), b) it assumes that evidence in favor of each alternative is accrued independently, and c) that evidence accumulation itself is linear and noiseless. The LBA does remarkably well in fitting response times and recovers standard patterns in RT data (Brown & Heath-cote, 2008).

We use the R package *rtdists* (Singmann et al., 2018) to implement the LBA. We adhere to the assumptions of the most simple LBA in that we allow each individual to have the same starting point bias (*A*), evidence boundary (*b*, with b > A), and non-decision time ( $t_0$ ). However, we allow for the drift rates to vary by accumulator (3 accumulators for 3 response types) and apply the scaling constraint that all drift rates must sum to 1 (i.e.  $\sum_{i=1}^{3} v_i = 1$ ). Drift rates are drawn from a Normal distribution which has a common standard deviation (*sv*) across all three accumulators. We use Maximum Likelihood Estimation (MLE) to fit all parameters to individual subjects.

#### Results

In *Experiment 1*, we excluded a total of 20 subjects (13 for below chance accuracy, 7 for LDI scores below zero) resulting in a total of 255 subjects with valid data. In *Experiment 2*, we excluded a total of 10 subjects (5 for below chance accuracy, 5 for LDI scores below zero) resulting in a total of 74 subjects with valid data.

#### **Choice Behavior**

In *Experiment 1*, individuals chose the correct response 71% of the time. They were most often correct on *Repeat* trials (40% of correct responses) and *Foil* trials (38%), followed by *Lure* trials (22%). In *Experiment 2*, individuals also chose the correct response 71% of the time. They were most often correct on *Repeat* trials (39% of correct responses) and *Foil* trials (38%), followed by *Lure* trials (23%). LDIs were comparable across experiments (*median*<sub>E1,E2</sub> = 0.37(.3), Figure 1). Recognition scores were similarly comparable (*median*<sub>E1</sub> = 0.78(.16), *median*<sub>E2</sub> = 0.78(.19)).

#### **Response Time Modeling**

In *Experiment 1*, median (IQR) RTs for each response type were as follows: *Repeat* = 1.14(0.42), *Lure* = 1.29(0.47), and *Foil* = 1.16(0.46). In *Experiment 2*, median (IQR) RTs for each response type were as follows: *Repeat* = 1.07(0.43), *Lure* = 1.29(0.43), and *Foil* = 1.12(0.45).

Our LBA parameter inferences are presented in Table 1.

Both experiments show the highest median drift rate on the Repeat accumulator, followed by the Foil accumulator, and lastly the Lure accumulator. Both experiments show that subjects have the same median *response caution*, which is often defined as the difference between the boundary and starting point (b - A, median = 0.28).

We next confirmed qualitatively that our model had good descriptive adequacy. To do this, we overlaid predicted RT quantiles on observed RT quantiles



Figure 1: Lure Discrimination Indices for both experiments median(IOR) = 0.37(0.3).

Parameter	Exp. 1	Exp. 2
Starting Point (A)	0.42(0.21)	0.45(0.28)
Boundary ( <i>b</i> )	0.70(0.21)	0.73(0.27)
Non Decision Time $(t_0)$	0.45(0.22)	0.39(0.20)
Drift: v <sub>Repeat</sub>	0.39(0.12)	0.39(0.13)
Drift: v <sub>Lure</sub>	0.26(0.15)	0.27(0.13)
Drift: v <sub>Foil</sub>	0.36(0.06)	0.34(0.06)
Drift: Standard Deviation	0.24(0.32)	0.24(0.27)

Table 1: Maximum Likelihood Estimates (*median*(*IQR*)) for LBA parameters for both experiments. We fit a total of 6 parameters and the seventh, drift rate for the Foil accumulator is  $v_{Foil} = 1 - v_{Repeat} - v_{Lure}$ .

(10%, 30%, 50%, 70%, 90%). We present an example of subject fits across ages and correct/incorrect responses in Figure 2, noting that most subjects were qualitatively well fit by the data.

#### **Relating LBA to MST**

As our key question of interest focuses on relating LBA parameters (components of an individual's memory retrieval and recognition processes – in particular drift rates and boundary) to how distinctly people encode memories, we assessed whether there were any correlations between the LBA parameters and behavioral scores (LDI and RS). We report Kendall's  $\tau$  rank correlation coefficient in the following analyses and adjust for multiple comparisons using the Bonferroni-Holm correction.

We found significant correlations between drift rates and LDI as shown in Figure 3. In particular, we found a negative correlation between the drift rate for the Repeat Accumulator and LDI in *Experiment 1* ( $\tau_{\text{Kendall}} = -0.276, p < 0.01$ ) and *Experiment 2* ( $\tau_{\text{Kendall}} = -0.20, p < 0.05$ ) trials. We further



Figure 2: Example plots where observed quantiles are overlaid with predicted quantiles for subjects old and young, correct and incorrect. Purple markers are observed RT quantiles for repeat trials, red for lures, and green for foils. Black lines are predictions from LBA. The horizontal vertical line represents the true proportion of repeat, lure, and foil trials  $(\frac{1}{3})$ .

found a positive correlation between the Lure Accumulator drift rate and LDI in *Experiment 1* ( $\tau_{\text{Kendall}} = 0.15, p < 0.01$ ). Finally, the correlations between drift rates for the Foil Accumulator and LDIs in *Experiment 1* or *Experiment 2* were not significant after adjusting for multiple comparisons.

We also observed a significant negative correlation between response caution (b - A) and LDI  $(\tau_{\text{Kendall}} = -0.14, p < 0.05)$  in *Experiment 1* only.

**Correlation Strengths** To compare correlation strengths, we used bootstrapping to resample the data and calculate Kendall's  $\tau$ s and the differences between each pair of  $\tau$ s (e.g.  $\tau_A - \tau_b$ ). We then examined whether the bootstrapped 95% confidence interval distributions of the differences between each pair of correlations included zero. If they did not include zero, we interpreted this as evidence in favor of a non-zero difference between the correlations compared.

Critically, we found that in Experiment 1, all three of the bootstrapped distributions of correlation differences between LDI and boundary, and LDI and the three accumulator drift rates did not include zero: boundary-Repeat (0.0973, 0.282), boundary-Lure (-0.492, -0.218), boundary-Foil (-0.412, -0.1479), Figure 4. We note that the CIs go in opposite directions for the Repeat vs Lure and Foil accumulators because of the negative correlation between LDI and Repeat accumulator drift rates. These results also held when we compared correlation strengths between response caution and the three accumulator drift rates: response caution-Repeat (0.054, 0.265), response caution-Lure (-0.456, -0.137), response caution-Foil (-0.397, -0.139). In Experiment 2, however, all of the CIs contained zero: boundary-Repeat (-0.139, 0.298), boundary-Lure (-.451, 0.052), boundary-Foil (-0.350, 0.105). Again, the same held for response caution: response caution-Repeat (-0.142, 0.272), response caution-Lure (-0.312, 0.101), response caution-Foil (-0.345, 0.125)

We also found that the correlations between the drift rate accumulators and LDIs were significantly different in *Experiment 1*. Specifically, the LDI-repeat accumulator thresholds were stronger than the LDI-lure accumulator drift (-0.680, -0.326) and the LDI-foil accumulator drift (-0.565, -0.309). We further found that the correlation between LDI-lure accumulator drift (0.026, 0.357). In *Experiment 2*, we only found that the LDI-repeat accumulator drift greater than the LDI-foil accumulator drift (-0.381, -0.043).



Figure 3: Correlations between Accumulator drift rates and the LDI across both experiments. We find statistically significant correlations between the drift rate of the Repeat accumulator and LDI in both experiments ( $\tau_{E1} = -0.276$ ,  $\tau_{E2} = -0.26$ ). We further find a significant correlation between the drift rate of the Lure accumulator and LDI in Experiment 1 ( $\tau_{E1} = 0.15$ )

#### **Stability of Measures**

Given the correlation between LDI and drift rates in both experiments, we wanted to see if the drift rate may in fact be a more stable behavioral measure than LDI. To evaluate the stability of the fit parameters and behavioral measures, we performed a split-halves analysis. Specifically, for each subject, we separately estimated each parameter and metric of interest on randomly selected halves of trials. We then computed the Mean Square Error for all parameters fit (both in the response time modeling and in choice behavior), Table 2. Specifically,



Figure 4: Boostrapped correlation differences between boundary and LDI, and drift rate and LDI for the three different accumulators in *Experiment 1*. All three 95% CIs do not include zero: boundary-Repeat (0.0973, 0.282), boundary-Lure (-0.492, -0.218), boundary-Foil (-0.412, -0.1479).

we calculated LBA measures, LDI, and Recognition Scores twice for for all odd numbered trials, and all even numbered trials separately.

Parameter	Exp.1	Exp. 2
Starting Point (A)	0.049(0.014)	0.045(0.023)
Boundary ( <i>b</i> )	0.028(0.011)	0.034(0.020)
Non Decision Time $(t_0)$	0.042(0.013)	0.112(0.036)
Drift: v <sub>Repeat</sub>	0.009(0.006)	0.026(0.017)
Drift: v <sub>Lure</sub>	0.010(0.007)	0.019(0.015)
Drift: v <sub>Foil</sub>	0.007(0.005)	0.021(0.016)
Drift: Standard Deviation	0.04(0.013)	0.067(0.028)
Lure Discrimination Index	0.017(0.008)	0.034(0.020)
Recognition Score	0.008(0.006)	0.018(0.014)

Table 2: Mean square errors (Standard Error) for all parameters estimated by the LBA model and (below the line) for standard behavioral measures derived from the MST.

Supporting the hypothesis that signal discrimination is a stable measure within-individual, we found that the MSE of the drift rates for all the accumulators were the lowest in both experiments. We note that the degree of stability is an order of magnitude greater than all the other parameters in *Experiment 1*, the larger dataset with more trials per subject. To quantify differences between MSE across LBA and behavioral parameters (i.e. stability in measurements), we use the non-parametric paired Wilcoxon Rank Sum test and again correct for multiple comparisons using the Bonferroni-Holm correction. We found that the drift rates were more stable than all other LBA parameters (p < 0.01) and both behavioral parameters (LDI, Recognition Score p < 0.01) in *Experiment 1*. In *Experiment 2*, we found that drift rates were significantly more stable than all the LBA parameters (p < 0.01) except

the non-decision time, which was trendingly significant after correcting for multiple comparisons (0.05 . However, like in*Experiment 1*, the drift rates were more stable than both behavioral parameters <math>(p < 0.01).

#### Discussion

We present one of the first model based analysis of response times in the Mnemonic Similarity Task (MST). We use a simple sequential sampling model, the Linear Ballistic Accumulator (LBA), where evidence is accumulated independently for all three possible responses.

Our approach decomposed responses for this task into separable components of response execution and signal detection, allowing us to assess the individual stability of these processes, across subjects. We hypothesized that either or both the response caution (either boundary, b, alone or boundary minus starting point, b-A) or drift rate,  $v_i$ , to lure or foil trials would be key variables of interest for behavioral discrimination performance. Specifically, if the LDI is indeed a measure of pattern separation, we would expect higher drift rates on Lure and/or Foil accumulators, suggesting a boosted signal. At the same time, to the extent LDI reflects individual variability in response caution, boundary, or starting point bias, then this would be reflected in these terms.

We found that, although both parameters were significantly correlated with LDI, the drift rates were both a stronger predictor of the standard behavioral measure and also a more stable within-subject measurement. The latter point is of considerable interest given the extensive evidence that MST is a useful individual difference marker, predicting neurological dysfunction and cognitive performance in a wide variety of clinical and laboratory measures (Stark et al., 2019).

The finding that LDI is strongly influenced by evidence strength supports the suggestion that MST measures the degree of pattern separation underlying these responses. Further, our findings may enhance the application of MST in several ways. First, the finding that drift rates are a more stable within-subject measure suggests that it could be used to more finely predict the same sorts of outcomes currently predicted by LDI. Future work should examine the correspondence of this drift rate to cognitive and neurological outcomes of interest. Second, the use of sequential sampling models can enable extracting trial-by-trial timeseries reflecting putative underlying computations that drive behavior, which should support analysis of more precisely defined functional neuroimaging measures (Long et al., 2016). Finally, the robust statistical frameworks often used to fit these sorts of models may allow further refinement of the approach, producing even more stable trait-level estimates by, e.g., incorporating informative priors and models of contaminant behavior.

In sum, we have provided initial evidence that joint modeling of choices and response times can improve inference of trait-level properties underlying a widely used clinical and laboratory assessment tool. Future work will examine the robustness of this new metric in the many settings in which the MST has been applied.

#### Acknowledgements

The authors are grateful to Sharon M. Noh for providing data for Experiment 1, and Nora C. Harhen for providing data for Experiment 2. This work was supported by NIMH P50MH096889, NIA R21AG072673, and a NARSAD Young Investigator Award from the Brain and Behavior Research Foundation (AMB).

#### References

- Brown, S. D., & Heathcote, A. (2008). The simplest complete model of choice response time: Linear ballistic accumulation. *Cognitive psychology*, 57(3), 153–178.
- Long, N. M., Lee, H., & Kuhl, B. A. (2016). Hippocampal mismatch signals are modulated by the strength of neural predictions and their similarity to outcomes. *Journal of Neuroscience*, 36(50), 12677–12687.
- Marr, D. (1971). Simple memory: a theory for archicortex. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 262(841), 23–81.
- Singmann, H., Brown, S., Gretton, M., Heathcote, A., Voss, A., Voss, J., & Terry, A. (2018). *Package 'rtdists'*.
- Stark, S. M., Kirwan, C. B., & Stark, C. E. (2019). Mnemonic similarity task: A tool for assessing hippocampal integrity. *Trends in cognitive sciences*, 23(11), 938–951.

### **Biologically-Plausible Memory for Continuous-Time Reinforcement Learning**

Madeleine Bartlett<sup>†</sup> (madeleine.bartlett@uwaterloo.ca)

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

Nicole Sandra-Yaffa Dumont<sup>†</sup> (ns2dumont@uwaterloo.ca)

Computational Neuroscience Research Group, University of Waterloo, Waterloo, ON, N2L 3G1, Canada

Michael P Furlong (michael.furlong@uwaterloo.ca)

Computational Neuroscience Research Group, University of Waterloo Waterloo, ON, N2L 3G1, Canada

#### Terrence C Stewart (tcstewar@uwaterloo.ca)

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

#### Abstract

Reinforcement learning, and particularly Temporal Difference learning, has been inspired by, and offers insights into, the mechanisms underlying animal learning. An ongoing challenge to providing biologically realistic models of learning is the need for algorithms that operate in continuous time and can be implemented with spiking neural networks. This paper presents a novel approach to Temporal Difference learning in continuous time –  $TD(\theta)$ . This approach relies on the use of Legendre Delay Networks for storing information about the past that will be used to update the value function. A comparison of the discrete-time TD(n) and continuous  $TD(\theta)$  rules on a simple spatial navigation RL task in a largely non-spiking network is presented, and the theoretical implications and avenues for future work are discussed.

**Keywords:** Reinforcement Learning; Temporal Difference learning; continuous time; Legendre Delay Network

#### Introduction

Reinforcement Learning (RL), as opposed to supervised learning, is a plausible description of animal learning. Animals must learn through continual interaction with their environment, and often demonstrate competence after very few interactions. While RL models of learning are useful, their implementations ignore fundamental aspects of biological implementations. In this paper we present a method of implementing continuous-time Temporal Difference (TD) learning rules with finite memory using a biologically plausible component, the Legendre Delay Network (LDN), a recurrent neural network that optimally represents time-varying signals over a finite history window (see Figure 1).

Psychological studies of animal learning have inspired many core RL algorithms, and similarities have been found between structures and signals in the mammalian brain and RL models. Dopaminergic neurons, thought to encode reward prediction error (Schultz et al., 1997; Cohen et al., 2012), similar to the TD error signal (Sutton, 1988; Sutton & Barto, 2018), project into the dorsal and ventral subdivisions of the striatum (Björklund & Dunnett, 2007). These regions, in turn, have been hypothesized to function like the actor and critic of Actor-Critic (AC) models (Joel et al., 2002).

These authors contributed equally.



Figure 1: A spiking neural network was used to implement the critic portion of an Actor-Critic network. The lower plot shows a snippet of the value function learned by the network. An LDN was used to remember this output and a delayed value signal was decoded from this LDN and plotted. The top spike raster plot displays the spiking activity of neurons from the population representing the LDN memory.

TD learning reflects dopaminergic neurons' behaviour during an association task wherein repeated exposure to a conditioned-unconditioned stimulus (CS-US) pairing results in excitation at the time of the learned CS (Schultz et al., 1997). The model further predicts a larger prediction error in response to unexpected rewards compared to expected rewards, and a smaller prediction error when a predicted reward is omitted than when it is received. Both of these predictions are also reflected in the behaviour of dopaminergic neurons (Schultz, 1998; Cohen et al., 2012; Nakahara et al., 2004).

Despite the similarities between the TD error signal and

neural correlates of RL, discrepancies between TD learning and biological RL remain. Namely, TD learning rules often operate in discrete time. The schedule of events – state transitions, actions taken, rewards received – must be described in terms of discrete time steps.

When TD learning rules are implemented for training artificial systems, they operate in a retrospective manner; the value of the state visited in a previous time step, t' < t is updated according to the rewards received between then, t', and now, t. For example, the value of the state  $s_{t-1}$  is updated according to the discounted value of the state the agent is currently in (i.e.  $s_t$ ) and the reward received at time t. TD(n) (Sutton & Barto, 2018, §7.1) improves the estimation of a state's value by updating the state value estimate using the states  $s_{t-n}, \ldots, s_t$  and the corresponding rewards  $r_{t-n}, \ldots, r_t$ .

The difficulty with these approaches that make them biologically implausible is that the TD formulation requires memory that is discretized across time steps. Spiking neurons, however, evolve in continuous time. Making spiking neurons implement discretized memory requires extra neural machinery.

The gap between TD and neural behaviour could be closed by using progressively smaller time steps, but this would result in larger memory requirements and longer training times to find the optimal policy. Consequently, to create RL models that more closely reflect biological systems, and that can cope with more complex problems, we need TD learning rules that exist in continuous time.

To address this problem we present a continuous time TD learning model using a recurrent neural network memory, the LDN, that is formulated in continuous time and is a biologically-plausible memory unit (Voelker & Eliasmith, 2018). An additional benefit of using the LDN is that our model naturally adapts to memories with arbitrary lengths. This is useful in mapping the TD(n) algorithm to a biologically plausible model that does not require additional resources as n grows.

We begin by reviewing prior approaches to continuous time RL, both non-spiking and spiking models of learning. We then introduce the principles of the Neural Engineering Framework, and the Legendre Delay Networks, which we use in this work. We then outline our modelling approach and describe  $TD(\theta)$ , our novel continuous time variant on the TD(n) learning algorithm. Next, we demonstrate  $TD(\theta)$  working on a continuous time RL task and in a spiking neural network. Finally, we discuss particular advantages of this continuous approach to modelling RL, as well as future directions for research.

#### Review

There has been past work on implementing RL algorithms in continuous time with spiking neural networks (Frémaux et al., 2013; Rasmussen et al., 2017). In such set-ups with online learning, future values are not available for TD learning. Instead, current estimates must be used for training past estimates. Obtaining the past activity of spiking neurons for such updates is a challenge. The Actor-Critic (AC) model in Frémaux et al. (2013) does not actually compute TD signals with spiking neurons to avoid this. The hierarchical reinforcement learning (HRL) model in Rasmussen et al. (2017) addresses the challenge by using two identical neural populations to represent current and delayed Q functions, with mechanisms for copying learned weights from one population to the other. This is similar to a TD(0) algorithm with a target network. However, the delay used is fixed in advance and this model does not generalize well to learning over longer time spans. Different tasks may require credit assignment over time windows of different lengths and, in many cases, better performance can be achieved by using reward information over many time steps for updates. Work on continuous time RL models - particularly ones that are biologically plausible – is limited. The theoretical framework of RL, Markov decision processes, can be formulated in continuous time and optimal policies can be obtained by solving the Hamilton-Jacobi-Bellman partial differential equation. This approach was used in Doya (2000) to develop algorithms for learning value functions and policies, which were found to learn a non-linear control task faster than traditional discrete time AC models. Continuous-Time Attention-Gated Memory Tagging (Zambrano et al., 2015) implements on-policy SARSA learning in continuous time using a neural working memory.

#### Background

#### **Reinforcement Learning**

Continuous-time RL is modelled as a continuous-time Markov decision process. There is a set of environment *states*, S and a set of agent *actions*, A. At any time t, the environment will be in some state,  $s(t) \in S$ . The agent will choose when to act and what action to take based on a stochastic *policy*,  $a(t) \sim \pi(s(t))$ . These actions will affect the state of the environment and the reward rate function, R(t) = R(s(t), a(t)). The task in RL is to learn a policy to maximize the expected discounted integral of future rewards:

$$\max_{\pi} \mathbb{E}_{\pi} \left[ \int_{t=0}^{\infty} \gamma^{t} R(t) dt \right], \tag{1}$$

where  $\gamma \in [0,1]$  is the discount factor. The above function, when at some particular state *s* at time *t*, is the value function.

$$V(s) = \mathbb{E}_{\pi} \left[ \int_{k=0}^{\infty} \gamma^{k} R(t+k) dk \Big| s(t) = s \right]$$
(2)

One can also define the 'Q' function, Q(s,a), in which the above expectation is also conditioned on the action taken at time *t*. A value (or Q) function can be learned by TD algorithms that take advantage of the recursive relationship between successive values.

$$V(s(t)) \approx \int_{k=0}^{\theta} \gamma^k R(t+k) dk + \gamma^{\theta} V(s(t+\theta)).$$
(3)

This expression can be used to update the value function.

$$V(s(t)) \leftarrow V(s(t)) + \lambda \left[ \int_{k=0}^{\theta} \gamma^k R(t+k) dk + \gamma^{\theta} V_{\pi}(s(t+\theta)) \right],$$
(4)

where  $\lambda$  is the learning rate, and the term in the square brackets is the TD error. This update, as written, is for tabular RL, in which the values of all states are stored in a table. To generalize to an infinite state space, one can model V(s) with a neural network trained using the TD error.

A popular architecture in RL is the Advantage Actor-Critic (A2C) model. In this setup, one learns both a value function (the *critic*) and a policy (the *actor*). The actor is used to select actions, while the critic is used to train the actor using the advantage function,

$$A(s,a) = Q(s,a) - V(s).$$
<sup>(5)</sup>

This advantage function can be approximated with the TD error signal.

#### **Neural Engineering Framework**

To create biologically realistic neural networks we require methods for representing vectors by the activity of spiking neurons, and to be able to perform computations on said vectors via projections between neural populations. The Neural Engineering Framework (NEF; Eliasmith & Anderson, 2003) provides such methods in the form of three principles: *representation*, *transformation* and *dynamics*.

The principle of *representation* explains how to encode a vector,  $\mathbf{x} \in \mathbb{R}^d$ , in the activity of a population of neurons,  $a(t) = G[\mathbf{E}\mathbf{x} + \mathbf{b}]$ , where  $\mathbf{E} = [\mathbf{e_1}, \dots, \mathbf{e_N}]^T$ ,  $\mathbf{e_i} \in \mathbb{R}^d$  are encoder weights for the  $i \in \{1, \dots, N\}$  neurons,  $\mathbf{b} \in \mathbb{R}^N$  are bias terms, and  $G[\cdot]$  is the neuron transfer function. Our experiments use the leaky integrate-and-fire neuron model for  $G[\cdot]$ , or it's rate approximation. *Representation* also explains how to decode the activity to recover the input vector,  $\mathbf{x}$ . The NEF's *transformation* principle provides the method for setting weights between two neural populations to compute a desired function. *Transformation* is achieved by solving for decoders – one for each neuron,  $\mathbf{d}_i$  – that compute a function of a population's input, instead of recovering the original input. Decoders can be solved for ahead of operations if the function is already known.

In this paper, our focus is on leveraging the principle of *dynamics*. Dynamical systems can be encoded in a population of spiking neurons using recurrent connections. It has been stated that synaptic weights (or more precisely, population decoders) can be optimized in advance if desired function samples are available. However, if the desired transformation is not known in advance, for example, the mapping between states and values in RL, online learning rules can be used to modify synaptic weights. The Prescribed Error Sensitivity (PES; MacNeil & Eliasmith, 2011) is a biologically plausible supervised learning rule. To learn a connection between a pre- and post-population of neurons, this rule modifies the pre-population's decoders in response to an error signal:

$$\Delta \mathbf{d}_i = \kappa \boldsymbol{\mathcal{E}}(t) a_i, \tag{6}$$

which is equivalent to modifying synaptic weights by

$$\Delta w_{ij} = -\kappa \alpha_j \mathbf{e}_j \cdot \boldsymbol{\mathcal{E}}(t) a_i \tag{7}$$

where  $\kappa$  is a learning rate,  $a_i$  are pre-population neural activities (filtered spikes),  $\alpha_j$  are post-population activities,  $\mathbf{e}_j$  are the post-population encoders, and  $\boldsymbol{\mathcal{E}}$  is an error signal we seek to minimize. This signal may be computed by other neural populations in a model. Biologically, we can think of those populations as dopaminergic neurons that can modify weights in this way via dopamine levels. Real data of spike timing dependent plasticity is matched by PES when used in combination with the unsupervised Bienenstock, Cooper, Munro (BCM) learning rule, which sparsifies weights (Bekolay et al., 2013).

#### Legendre Delay Network

Consider the problem of computing a delay of some signal u(t) (for example, computing a delayed reward for TD updates) using a recurrent neural network. In deep learning, recurrent networks are typically trained in a supervised fashion using backpropagation-through-time. However, this is not biologically plausible. In real behavioral tasks, examples of "correct" behavior are generally not available and, instead, learning must be done using only temporally sparse rewards. Additionally, it is unknown how derivatives of spiking activity would be calculated in the brain and propagated through multiple layers of neurons. Furthermore, the same connections and weights are used in its forward and backwards passes, but real synapses are unidirectional.

In this work we use properties of Legendre polynomial representations of time varying-functions, and the Legendre Delay Network (LDN: Voelker & Eliasmith, 2018) to encode history. Legendre polynomials are orthogonal basis functions that can be used to represent functions over fixed input windows. We use the shifted Legendre basis polynomials, defined by the functions  $P_0(t) = 1$ ,  $P_1(t) = 2t - 1$ , and the recursion  $(n+1)P_{n+1}(t) = (2n+1)P_n(t) + nP_{n-1}(t)$ . The polynomials are defined over the domain [0,1], and the coefficients of the Legendre representation of a function f(t) over a window  $[t,t+\theta]$  are  $a_n = \frac{2n+1}{2} \int_t^{t+\theta} f(\tau) P_n((\tau-t)/\theta) d\tau$ . A representation using the first q polynomials is said to have an order of q. Legendre polynomials are orthogonal, such that  $\int_0^1 P_i(t)P_j(t)dt = \frac{1}{2i+1}$  when i = j and zero otherwise. The LDN is a dynamic system that approximates the Legendre polynomial coefficients of an input signal over a sliding history window of length  $\theta \in \mathbb{R}^+$ . The coefficients are represented using the LDN's memory state,  $\mathbf{m} \in \mathbb{R}^{q}$ , for an order q Legendre representation. **m** is updated according to  $\dot{\mathbf{m}}(t) = \mathbf{A}\mathbf{m}(t) + \mathbf{B}u(t)$ , where u(t) is the input signal. To ef-



Figure 2: Screenshot of the  $8 \times 8$  Mini-Grid environment.

fect a Legendre basis, A and B are defined such that

$$A_{ij} = \frac{2i+1}{\theta} \begin{cases} -1 & i < j\\ (-1)^{i-j+1} & i \ge j \end{cases}, \quad B_i = \frac{(2i+1)(-1)^i}{\theta}.$$
(8)

The values of **A** and **B** are fixed once  $\theta$  and *q* are selected. For discrete-time applications we approximate **A** and **B** with  $\bar{\mathbf{A}} = e^{\mathbf{A}}$  and  $\bar{\mathbf{B}} = A^{-1}(e^{A} - \mathbf{I})\mathbf{B}$ , using a zero-order hold and dt = 1, as per Chilkuri & Eliasmith (2021).

#### Methods

In the case of RL, the PES learning rule can be used to modify synaptic weights in response to the TD error signal. Such errors are typically written as an update to the value function at time t using future information (rewards and/or values at time t+1, t+2, etc.). When learning online, the network does not have access to future information – it only has access to present values and the past via an LDN memory. This means that we will update the value function in the past (at say,  $t - \theta$ ) using information obtained since then. This requires the use of neurons' past activities in the PES update.

Assume we have a population of N neurons representing the state,  $s(t) \in \mathbb{R}^d$ . Let  $\mathbf{m}_{a_j}(t) \in \mathbb{R}^{q_a \times N}$  be the LDN memory of the  $j^{th}$  neuron's activities (a filtered spike train). Then the PES update is given by

$$\Delta \mathbf{d}_j = \kappa \boldsymbol{\mathcal{E}}(t) \mathbf{P}^{q_a}(\boldsymbol{\theta}) \mathbf{m}_{a_j}(t), \tag{9}$$

where  $\mathbf{P}^{q_a}(\theta) \in \mathbb{R}^{1 \times q_a}$  is the vector of the shifted Legendre polynomials (of degree one to  $q_a$ ), evaluated at  $\theta$ . The simplest RL learning rule that can be implemented in this way is the TD(0) rule – an update of the value at just a short time in the past  $(t - \Delta t)$  using only the current reward rate. Let  $\mathbf{m}_V(t) \in \mathbb{R}^{q_v}$  be an LDN memory of the value function. The TD(0) error and PES update is given by

$$\boldsymbol{\mathcal{E}}^{(0)}(t) = \boldsymbol{R}(t) + \boldsymbol{\gamma} \boldsymbol{V}(t) - \mathbf{P}^{q_{v}}(\Delta t) \mathbf{m}_{V}, \qquad (10)$$

$$\Delta \mathbf{d}_{i} = \kappa (R(t) + \gamma V(t) - \mathbf{P}^{q_{v}}(\Delta t)\mathbf{m}_{V})\mathbf{P}^{q_{a}}(\theta)\mathbf{m}_{a_{i}}(t).$$
(11)

Learning rules that use a longer history of rewards require an LDN memory of the reward rate over time,  $\mathbf{m}_R \in \mathbb{R}^{q_r}$ . A



Figure 3: A diagram of AC network using LDN memories. Blocks represent neural populations, grey blocks indicate populations representing LDNs, solid arrows represent connections, and dotted lines represent weight modification. LDNs are used to remember the reward received from the environment and the value function. The output of these LDNs projects onto the TD error population with connection weights given by (13). This TD error, along with decoded output from the LDN representing the activities of the state neurons, is used to modify the connection weights between the state and value populations via (14).

learning rule that uses the full  $\theta$  time window of the LDN memories is

$$\mathcal{E}^{(\theta)}(t) = \int_0^1 \gamma^{1-\tau} R(t-\theta\tau) d\tau + \gamma V(t) - V(t-\theta), \qquad (12)$$
$$= \left(\int_0^1 \gamma^{1-\tau} \mathbf{P}^{q_r}(\theta\tau) d\tau\right) \mathbf{m}_R(t) + \gamma v(t) - \mathbf{P}^{q_v}(\theta) \mathbf{m}_V(t)$$

$$\Delta \mathbf{d}_j = \kappa \boldsymbol{\mathcal{E}}^{(\theta)}(t) \mathbf{P}^{q_a}(\theta) \mathbf{m}_{a_j}(t).$$
(14)

This is the novel  $TD(\theta)$  learning rule. The discounted integral over the reward history can be directly computed from it's LDN representation and used to update the value function. An experiment was conducted to demonstrate how a simple, non-spiking version of this rule could be implemented, and how its performance compares to the standard TD(n) learning rule on a simple spatial navigation RL task. This experiment is a preliminary exploration of the developed learning rule, intended as a starting point from which to build a fully spiking, biologically plausible model of RL in continuous time. For this experiment, two AC networks were implemented, one using the standard TD(n) learning rule and the other using the a non-spiking version of TD( $\theta$ ). Each network was then tested on the Gym MiniGrid environment (Chevalier-Boisvert et al., 2018).



Figure 4: Schematic of the neural network. The TD update (dashed line) is computed by the network in Figure 3.

#### Learning Task

For this demonstration, we used the  $8 \times 8$  MiniGrid environment where the task is to learn how to navigate to a goal location (see Figure 2). This environment consists of  $6 \times 6$  (36) possible locations. At each time step, the agent is able to take 1 of 3 possible actions (move forward, turn left, turn right). At the beginning of each learning trial, the agent is initialised in the top left-hand corner and goal location is the bottom right-hand corner.

Per learning trial, the agent had a total of 200 time steps in which to find the goal location. The trial would be terminated either at the end of the 200 time steps or once the agent had reached the goal, and the environment was reset for the agent to try again. For each approach (TD(n) vs. TD( $\theta$ )) the network was run for a total of 500 learning trials, and we set n = 2.

Importantly, we found that when using  $TD(\theta)$ , good performance was obtained if the agent was made to wait for at least 2 time steps in each state (i.e. spending a total of 3 time steps in each state). We argue that this is because when the agent was not made to wait, the duration of reward presentation was too short, lasting only 1 *ms*. By making the agent wait, we extended the duration of the reward.

#### Actor-Critic Network

The AC Network was implemented in Python using the NEF (Eliasmith & Anderson, 2003) (see Figure 4 for the network schematic). The network's input was the agent's current state (a 3D vector containing the agent's x,y coordinate location and the direction it's facing), the most recent action selected and the most recent reward. The state information was transformed into a one-hot representation, which was then passed to the hidden layer consisting of 3,000 rate neurons. The TD update was performed in the rule node, and was used to train the network's decoder weights ( $W_{decoder}$ ). The network's outputs were the updated state value, and a vector containing the preferences for each action available to be taken in the next time step.



Figure 5: Plot showing total reward per episode, across all learning trials, for TD(n) Baseline and TD( $\theta$ ).

When using the standard TD(n) learning rule, rewards and state values needed to perform the TD(n) update were stored in arrays. However, with TD( $\theta$ ) where LDNs were used for storing the rewards and values, the reward was passed into an LDN node. The output from this node was the integral of the discounted Legendre polynomials across the LDN window,  $\left(\int_{0}^{1}\gamma^{1-\tau}\mathbf{P}^{q_{r}}(\theta\tau)d\tau\right)$ . A second LDN node (V(t)) was used to store the value of each state encountered. This value would be retrieved *n* time steps later when it was time for that state's value to be updated.

#### Results

To assess the performance of each network we calculated the total reward gained in each learning trial and plotted the rewards over the 500 learning trials for each approach. In the case of  $TD(\theta)$ , the total reward received at the end of each learning trial was divided by 3 to correct for the wait time. These results are shown in Figure 5. Both approaches show similar performance; both seem to find an effective, stable policy within 200 learning trials.

The learned value for each state (location and direction) in the MiniGrid task was also calculated and is shown in Figure 6. These plots reveal that both the TD(n) and TD( $\theta$ ) networks assigned high value to those states that led in a straight-line path to the goal. This further suggests that both networks were able to learn similar solutions for the task. The main take-away from this is that the TD( $\theta$ ) rule allows us to solve RL problems where the reward history is represented in continuous time. Given the potential for the LDN to be implemented in a spiking neural network, this approach shows promise for modelling RL in a more biologically plausible way.

#### Discussion

This paper presents a novel, continuous time approach to implementing TD learning. The proposed  $TD(\theta)$  is a version of TD(n) that incorporates LDNs for dynamically maintaining a memory of received rewards in continuous time. As a preliminary exploration of the novel  $TD(\theta)$  learning rule, an experiment was run comparing the performance of an AC network on a simple grid-world task when using the standard



Figure 6: The learned values for each state on the MiniGrid task for TD(n) Baseline and TD( $\theta$ ). From left to right, these plots show the learned values of each position in the MiniGrid world when the agent is facing right, down, left and up.

TD(n) learning rule in discrete time vs. TD( $\theta$ ) for continuous time. Figure 5 illustrates that TD( $\theta$ ) was able to learn a stable policy in roughly the same number of trials as TD(n).

An advantage of this novel approach over other methods is that it can be readily adapted for different lengths of memories without additional model complexity. With the standard TD(n) rule, for example, additional memory resources must be employed in order to use larger values of *n*. In contrast, when using LDNs the only change needed to accommodate a larger *n* is to increase  $\theta$  (the length of the LDN window). The use of LDNs may also prove beneficial when applied to the TD( $\lambda$ ) learning rule, which requires resources to maintain a memory of all previously visited states and their values, as well as the eligibility trace which describes how recently and frequently each state has been visited.

Further exploration is needed to establish whether the novel  $TD(\theta)$  learns similar policies or produces behaviours that deviate from existing TD learning rules. However, as a preliminary finding, this result is promising. It should be noted, however, that the  $TD(\theta)$  network did require that the agent wait in each state in order to learn the task. First steps for future work, therefore, will be to more fully explore the effects of having the agent wait, and to establish why it was needed. We will also explore possible alternative solutions to mitigate any effects due to reward presentation duration.

The MiniGrid task used to test the novel approach is relatively simple and formulated in discrete space and time. Future work will therefore also focus on applying the novel  $TD(\theta)$  rule to more complex, continuous problems. Additionally, given that LDNs can be implemented in a spiking network (Voelker & Eliasmith, 2018), by coupling this approach with biologically plausible methods for representing continuous state spaces such as Spatial Semantic Pointers (SSPs; Komer et al., 2019), it is theoretically possible to implement a critic network entirely in spiking neurons.

#### **Online Resources**

Experiment and analysis scripts can be found in the github repository (https://github.com/maddybartlett/ Bio\_Plausible\_Memory\_Continuous\_Time\_RL).

#### References

- Bekolay, T., Kolbeck, C., & Eliasmith, C. (2013). Simultaneous unsupervised and supervised learning of cognitive functions in biologically plausible spiking neural networks. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 35).
- Björklund, A., & Dunnett, S. B. (2007). Dopamine neuron systems in the brain: an update. *Trends in neurosciences*, 30(5), 194–202.
- Chevalier-Boisvert, M., Willems, L., & Pal, S. (2018). *Minimalistic Gridworld Environment for OpenAI Gym.* https://github.com/maximecb/gym-minigrid. GitHub.
- Chilkuri, N. R., & Eliasmith, C. (2021, 18–24 Jul). Parallelizing legendre memory unit training. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning* (Vol. 139, pp. 1898–1907). PMLR.
- Cohen, J. Y., Haesler, S., Vong, L., Lowell, B. B., & Uchida, N. (2012). Neuron-type-specific signals for reward and punishment in the ventral tegmental area. *nature*, 482(7383), 85–88.
- Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural computation*, *12*(1), 219–245.
- Eliasmith, C., & Anderson, C. H. (2003). Neural engineering: Computation, representation, and dynamics in neurobiological systems. MIT press.
- 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.
- Joel, D., Niv, Y., & Ruppin, E. (2002). Actor–critic models of the basal ganglia: New anatomical and computational perspectives. *Neural networks*, *15*(4-6), 535–547.
- Komer, B., Stewart, T. C., Voelker, A., & Eliasmith, C. (2019). A neural representation of continuous space using fractional binding. In *Cogsci* (pp. 2038–2043).
- MacNeil, D., & Eliasmith, C. (2011). Fine-tuning and the stability of recurrent neural networks. *PloS one*, 6(9), e22885.
- Nakahara, H., Itoh, H., Kawagoe, R., Takikawa, Y., & Hikosaka, O. (2004). Dopamine neurons can represent context-dependent prediction error. *Neuron*, 41(2), 269– 280.
- Rasmussen, D., Voelker, A., & Eliasmith, C. (2017). A neural model of hierarchical reinforcement learning. *PloS one*, *12*(7), e0180234.
- Schultz, W. (1998). Predictive reward signal of dopamine neurons. *Journal of neurophysiology*, 80(1), 1–27.

- Schultz, W., Dayan, P., & Montague, P. R. (1997). A neural substrate of prediction and reward. *Science*, 275(5306), 1593–1599.
- Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine learning*, *3*(1), 9–44.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- Voelker, A. R., & Eliasmith, C. (2018). Improving spiking dynamical networks: Accurate delays, higher-order synapses, and time cells. *Neural computation*, *30*(3), 569– 609.
- Zambrano, D., Roelfsema, P. R., & Bohte, S. M. (2015). Continuous-time on-policy neural reinforcement learning of working memory tasks. In 2015 international joint conference on neural networks (ijcnn) (pp. 1–8).

### Fast Online Reinforcement Learning with Biologically-Based State Representations

#### Madeleine Bartlett (madeleine.bartlett@uwaterloo.ca)

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

#### Terrence C Stewart (tcstewar@uwaterloo.ca)

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

#### Jeff Orchard (jorchard@uwaterloo.ca)

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

#### Abstract

In previous work, we provided a neurally-based Actor-Critic network with biologically inspired grid cells for representing spatial information, and examined whether it improved performance on a 2D grid-world task over other representation methods. We did a manual search of the parameter space and found that grid cells outperformed other representations. The present work expands on this work by performing a more extensive search of the parameter space in order to identify optimal parameter sets for each configuration using one of four representation methods (baseline look-up table, one-hot, random SSPs and grid cells). Following this optimization, the baseline, onehot and random SSPs methods did show improvement over the previous study, in some cases showing performance as good as grid cells. These findings, combined, suggest that whilst the baseline and one-hot methods do perform well once optimized, grid cells do not necessarily require optimization in order to produce optimal performance.

**Keywords:** Reinforcement Learning; grid cells; Spatial Semantic Pointers;

#### Introduction

Humans and non-human animals are able to learn how to interact with their environment in order to maximise rewards through a process of trial and error (Mackintosh, 2019). This ability has inspired the development of Reinforcement Learning (RL) methods for training artificial systems. The goal of RL methods is to learn a policy of how to move through an environment or perform a task in order to maximise reward (Sutton & Barto, 2018). In the case of neurally-based RL algorithms, this often involves implementing either a policy- or value-based algorithm. With value-based approaches, a network is provided with the current state  $s_t$  as input and then calculates the value of that state  $V(s_t)$  with the longer-term goal of maximising the value function V(s). Policy-based approaches often involve again providing the network with the current state and having the network produce a distribution indicating the likelihood of performing different actions (a) in that state ( $[p(s_t, a_1), p(s_t, a_2), \dots, p(s_t, a_n)]$ ). Regardless of the approach taken, this is generally a more difficult task than traditional neural-network learning because the network needs to both learn about the task, and learn the right way to represent the input data in order to produce the correct output.

In contrast, biological systems will, in most cases, already have a representation that can be re-purposed for a novel task.



Figure 1: Screenshot of the  $8 \times 8$  Mini-Grid environment.

For example, in the case of spatial navigation, much evidence points to grid cells as being involved in the encoding of spatial locations. Grid cells are neurons that encode a representation of space which takes the form of a repetitive hexagonal grid pattern (Hafting et al., 2005). This distinction is often pointed to as an explanation for why biological systems seem to learn RL tasks faster than artificial systems.

Taking inspiration from biological systems in the context of spatial navigation RL tasks has proved advantageous. A study by Gustafson & Daw (2011) involved training a network to solve a series of navigation tasks using a TD-based network where the state representation was in the form of a look-up table, place cells or grid cells. As a secondary finding, Gustafson & Daw (2011) observed that, in most tasks, the use of grid and place cell basis functions led to faster learning than when the state was represented using a tabular basis function. A study by Banino et al. (2018) involved generating grid cell representations of spatial information by training a recurrent neural network to perform path integration. This grid cell network was then use in conjunction with an Actor-Critic (AC) network and trained using deep RL to solve navigation tasks. This study found that performance when using this grid cell network was better than that of an agent that used place cell representations of the state.

The Neural Engineering Framework (NEF) offers additional, alternative biologically-plausible methods for representing space (Eliasmith & Anderson, 2003). Not only does the NEF provide tools for implementing models based on spiking neurons, but more recently spatial representations (Komer et al., 2019) and grid cells (Dumont & Eliasmith, 2020) can be seen as special cases of a general vector-based representation called Spatial Semantic Pointers (SSPs)

The current study is an extension of Bartlett et al. (in press) in which different methods for representing the state were compared in a spatial navigation RL task, including random SSPs and grid cells. In the previous work, a total of 4 representation methods (baseline, one hot, random SSPs and grid cells) were compared by training TD-based AC networks (using either the TD(0) or TD( $\lambda$ ) learning rules) to solve a simple spatial navigation RL task. To avoid questions of the biological plausibility of learning rules such as back-propagation, we only applied the learning rule to a single layer of neural connection weights. This means that the network must make use of the style of representation that is available to it, rather than learning a custom style of representation for the particular task. The previous exploration found that the use of biologically-inspired grid cells for representing the state resulted in the network learning to solve the task in fewer learning trials. The present work expands on this by performing a more thorough search of the parameter space for each configuration, in order to find optimal parameter sets. We then compare the optimized configurations to determine whether the use of grid cells does in fact lead to improved performance, or whether this finding was an artifact of the manually selected parameter values used in the initial study.

#### Methods

#### Learning Task

For these experiments, we compared the performance of each network configuration on the Gym MiniGrid navigation task (Chevalier-Boisvert et al., 2018). Specifically, we used the  $8 \times 8$  MiniGrid environment where the agent's task on each trial is to navigate to a goal location. This environment consists of  $6 \times 6$  (36) possible locations. At each timestep, the agent is able to take 1 of 3 possible actions (move forward, turn left, turn right). At the beginning of each learning trial, the agent was initialised in the top left-hand corner (Figure 1, red triangle) and was tasked with reaching the bottom right-hand corner (Figure 1, green square).

#### **Actor-Critic Network**

The AC Network was implemented in Python using the NEF (Eliasmith & Anderson, 2003) (see Figure 2 for the network schematic). Input to the network is the agent's current state, and the most recent action and reward. The state is a 3-dimensional vector containing the agent's location in the grid world (in the form of (x, y) coordinates) and the direction it's facing (0 = pointing right, 1 = down, 2 = left, 3 = up). This state information is transformed into the chosen representation (one hot, random SSPs, or grid cells) in the representation node. The representation is then passed to a hidden layer consisting of rate neurons utilizing a rectified linear function.



Figure 2: Schematic of Actor-Critic Network.

The neuron activities along with the action and reward are then used in a rule node where the TD update is performed. The TD update trains the network's weights to approximate the optimal policy for completing the task with maximum reward. The output from the network is the updated state value, and a vector containing the preferences for each action, which is used to decide which action to take in the next timestep.

#### Representations

**One Hot:** The one-hot method represents states by storing an array containing one value for each possible state. States are represented by setting all of the values in the array to 0 except for one which is set to 1. The position of this 1 value in the array corresponds with the state being represented. When implemented without the use of neurons, this method of representation is equivalent to a look-up table. As such, this method was used in two of the representation conditions: one hot and baseline. In the baseline condition, the one-hot method was implemented and the network did *not* contain any neurons. In the one-hot condition, however, the one-hot representation was passed to the hidden neuron layer before being used in the TD update. The baseline condition was the only condition that did not utilize the hidden neuron layer.

**Spatial Semantic Pointers:** Two different styles of neurally plausible vector-based representations were implemented. The first of these is randomly chosen SSPs (Komer et al., 2019). The SSP method extends the idea of vector symbolic architectures (VSAs) (Gayler, 2004) to continuous spaces. Say we want to represent an ordered list, e.g. [A, B, C]. With VSAs, we can do this by binding the list items (A, B and C) to *d*-dimensional vectors for each position in the list (e.g.  $POS_1, POS_2, POS_3$ ). Thus the list is represented as:

$$A \circledast POS_1 + B \circledast POS_2 + C \circledast POS_3,$$

where  $\circledast$  is the binding operator. Rather than generating unique random vectors for each position in the list, we can generate them in a more principled way. If we create a vector for the first position (*POS*), then we can generate a vector for the second position by binding the *POS* vector to itself. Thus for each integer index n of a structure, the positional vector can be generated by binding the first positional vector to itself n times (*POS*<sup>n</sup>):

$$A \circledast POS + B \circledast POS^2 + C \circledast POS^3.$$

Generalizing this method to representing continuous variables involves the use of fractional binding – rather than raising the position vector *POS* only to integer values (e.g.  $POS^2$ ), it is possible to raise it to some fractional power (e.g.  $POS^{1.5}$ ). The mathematical meaning of this operation is dependent on the particular choice of the  $\circledast$  operator in the VSA. One common choice is circular convolution. Since circular convolution can be implemented as multiplication in the Fourier domain, the corresponding fractional number of binding operations can be expressed as:

$$POS^n = \mathcal{F}^{-1} \{ \mathcal{F} \{ POS \}^n \}, \qquad n \in \mathbb{R}$$

Thus performing this fractional binding involves performing the Fourier transform  $\mathcal{F}$ , raising each Fourier coefficient to the fractional power *n*, and then doing the inverse Fourier transform  $\mathcal{F}^{-1}$ . The result is our SSP.

In our VSA system,  $\mathcal{F}\{POS\}$  is a unit-length complex number, so raising it to the exponent *n* simply multiplies its phase by *n*. In this way, an SSP encodes the value *n* in the phases of its Fourier coefficients. This *phase encoding* is similar in nature to how we represent time on an analog clock. The hour-, minute-, and second-hands of a clock change phase (rotate) as time progresses. Hence, we can tell what time it is by looking at the phase of the 3 hands on the clock. Importantly, the 3 hands oscillate at very different frequencies, allowing us to determine the time to the precision of 1 second, but over a 12-hour period.

Now that we can represent continuous variables, we can encode multi-dimensional state information into such vectors. For example, in the MiniGrid task, the state at any given time is made up of the agent's (x, y) coordinate location on the grid, and the direction in which it's facing (z). Encoding this as a single SSP, S, can be done using:

$$S = \mathcal{F}^{-1}\left(\mathcal{F}(X)^{x}\mathcal{F}(Y)^{y}\mathcal{F}(Z)^{z}\right)$$

where, for each value in the state, we choose a highdimensional unitary vector (X, Y, or Z). We then compute its Fourier transform,  $\mathcal{F}(X)$ , raise that to an exponent,  $\mathcal{F}(X)^x$ , and multiply it by the other transformed values,  $(\mathcal{F}(X)^x \times \mathcal{F}(Y)^y \times \mathcal{F}(Z)^z)$ . Finally, we take the inverse Fourier transform in order to get our final SSP for that state.

This method of encoding the state was used for the random SSP condition, with the additional note that the encoding weights ( $W_{encoders}$ , Figure 2) were randomly generated, resulting in neurons that were random pattern cells (see Figure 3A).

**Grid Cells:** In contrast with the random SSP method, by carefully selecting X, Y, Z, and  $W_{encoders}$  as per Dumont & Eliasmith (2020), it is possible to generate grid cells. While



Figure 3: Receptive fields of neurons with random encoders (A) and of grid cells (B) used to represent SSPs.

Table 1: Table showing which parameters were tested, the ranges of values tested, and which network configurations involved these parameters.

Parameter	Values Tested	Configurations	
Alpha	range 0 - 1	All	
Beta	range 0 - 1	All	
Gamma	range 0 - 1	All	
Lambda	range 0 - 1	All with $TD(\lambda)$	
Neurons	range 100 - 5000	All with neurons	
Sparsity	range 0 - 1	All with neurons	
Dimensions	64, 128, 256, 532	SSP rep	

the mathematical details of this derivation are outside the scope of this paper, the general principle is to choose vectors such that the waves produced by the Fourier transform cause triplets of wave functions to interfere with each other to produce grid patterns (see Figure 3B). Furthermore, grids of different sizes and orientations (as observed in the hippocampus) are all produced out of the same vector, using the same maths as in the previous section, purely by selecting our base vectors and encoding connection weights. It should also be noted that, while the construction of these vectors does involve complex numbers, the resulting neural network is a standard feed-forward single-hidden-layer network with realvalued weights.

#### **NNI Experiments**

To perform hyper-parameter tuning, we used the Neural Network Intelligence (NNI) toolkit (Microsoft, 2021). In total, 8 NNI experiments were performed. The parameters being searched differed between network configurations. A list of all parameters (along with the range of possible values) that were searched is presented in Table 1. The NNI experiments used an annealing algorithm for tuning, which starts by selecting random values for the parameters, but over time selects values that are closer to the best ones observed. The optimization goal was to identify the set of parameters that minimized the number of runs needed to reach a goal rollingmean reward of 0.95 over the last 100 learning trials. Each NNI experiment was run for 12 hours.



Figure 4: Scatter plots showing the optimization curves of the NNI experiments.



Figure 5: Scatter plots showing the number of trials to reach the goal rolling-mean reward for experiments using the best performing parameter combinations.

#### Results

#### **NNI Results**

The first step for analysis was to calculate the number of trials needed to reach the goal rolling mean of 0.95. For the purposes of analysis, if an experiment failed to reach the goal, its reported number of trials to reach goal was manually set to 2,000 (the max number of learning trials). This ensured that these experiments could be included in the analysis.

In Figure 4 we present the optimization curves for all 8 NNI experiments. These plots provide a general idea of how successful NNI was in finding good parameter sets for each configuration. For the configurations using the TD(0) learning rule, it appears that the NNI experiment was able to identify good parameter sets (values which resulted in reaching the goal in 200 learning trials or less) fairly quickly. Whilst a similar pattern is evident for three of the configurations us-

ing the TD( $\lambda$ ) learning rules, the configuration using random SSPs to represent the state continued to fail to reach the goal throughout the experiment. In the initial study (Bartlett et al., in press), the mean number of trials needed to reach the goal was generally greater for this configuration compared to the others. The current findings suggest that this higher mean may have been the result of a higher number of failed runs – the previous study used the same approach of including failed runs in the analysis by setting the number of trials needed to reach the goal to the maximum number of trials (10,000).

The next step was to look more closely at the 'best' performing parameter sets. We identified the top 2% of experiments that achieved the goal in the fewest learning trials for each configuration. Table 2 presents the minimum and maximum number of trials needed to reach the goal for the top 2% of experiments using each configuration. Whilst the number of trials needed to reach the goal are (mostly) smaller here

	TD(0) Baseline	TD(0) One Hot	TD(0) Random SSP	TD(0) Grid Cells
N Experiments in top 2%	27	22	31	17
N Trials (min, max)	149, 150	127, 136	108, 116	105, 108
	TD( $\lambda$ ) Baseline	$TD(\lambda)$ One Hot	$TD(\lambda)$ Random SSP	$TD(\lambda)$ Grid Cells
N Experiments in top 2%	30	21	6	19
N Trials (min, max)	134	99, 100	102, 109	99, 103

Table 2: Table showing the number of experiments in the top 2% and the minimum and maximum (min, max) number of trials needed for experiments in the top 2% to reach the goal rolling-mean reward.

than the averages found in (Bartlett et al., in press), it is worth noting that the use of grid cells and random SSPs still result in faster learning than the baseline condition (and the one-hot condition where TD(0) is used).

We then examined the stability of these 'best' parameter values by identifying all of the experiments for which all of the parameter values fell within the range of those identified as the top 2%. Figure 5 shows the number of trials it took for each of these experiments to achieve the goal. From these figures we can identify that where TD(0) was used, all four configurations demonstrated good stability of the identified parameter combinations. In contrast, when using  $TD(\lambda)$ , the configuration using random SSP representation demonstrated markedly worse stability than any of the other configurations. Apart from this, the results from these experiments seem to support the argument that, where the TD(0) rule is used, the use of grid cells for representing the state in a spatial navigation RL task results in better performance than other methods. On the other hand, where the  $TD(\lambda)$  rule is implemented, all three networks using neurons in the hidden layer outperformed the baseline method, achieving the goal in close to 100 trials (compared to 134 trials for baseline, Table 2). This demonstrates that tailored methods for representing the state do at least as well as other methods.

#### $TD(\lambda)$ SSP Results Exploration

Considering the instability of the TD( $\lambda$ ) with SSP representation configuration, we felt it necessary to further explore this configuration in an attempt to identify the cause of the instability. In Figure 6, we can see all of the combinations of hyper-parameter values tested in the NNI experiment. Whilst a wide range of values was explored for most of the parameters, it seems that there was somewhat less exploration of the number of neurons in the hidden layer, and the number of dimensions used in the SSP representation. That is, in Figure 6, many of the lines seem to converge to the same few points on the 'Neurons' and 'Dimensions' axes (indicating that most of the NNI experiments used these few values), whilst tending to be more spread out along the other axes. Specifically, the NNI exploration seems to have mainly tested 64 and 256 dimensions, and 4117 and 4480 neurons. One potential reason why the NNI experiments did not explore these variables as much is because there was little difference in performance when exploring the values available for these two parameters, and so the NNI stopped varying them. If this is the case, then we may find better performance when setting the number of neurons and dimensions to values outside the ranges used. We therefore decided to force exploration of larger values for these parameters by running experiments where only the dimensions or number of neurons were manipulated.

We examined the effect of adding dimensions by running the same random SSP network using either 512 or 1024 dimensions. For the rest of the parameters, we chose a set of values from the top 2%. Each value for the dimensions parameter was tested 20 times, with a different seed each time. The results were compared with those obtained when the random SSP representation used 256 dimensions. Figure 7 illustrates that whilst the mean number of trials needed to reach the goal did decrease with increased dimensionality, the variability did not change, suggesting that increasing dimensionality did not effect the variability in performance.

We then examine the effect of larger numbers of neurons in the hidden layer. Using the same procedure as above, we compared performance when using the original 4,117 neurons to networks whose hidden layer contained 5,000, 6,000, 7,000, 8,000, 9,000 and 10,000 neurons. Figure 8 illustrates that changing this variable did not improve the stability of the network's performance. Given this instability, it seems that good performance while using random SSPs to represent the state relies on the luck of the seed.

#### Discussion

This study explored the impact of using biologically inspired state representations on the performance of a TD-based AC network on a simple RL task. Two learning rules, TD(0) and TD( $\lambda$ ), were implemented, and performance on the Gym MiniGrid task was compared when the state was represented using a baseline tabular method without neurons, vs. one-hot, random SSP, or grid-cell SSP methods with neurons. The NNI toolkit was used to conduct a search of the parameter space for each of the 8 configurations. The results of these experiments were used to identify parameter sets that resulted in the network achieving a rolling average reward of 0.95 over the last 100 learning trials in the fewest number of trials.

We found that the best 2% of configurations solved the MiniGrid task in under 200 trials for all learning rules and



Figure 6: A parallel coordinates plot showing all of the hyperparameter value combinations tested in the NNI experiment using TD( $\lambda$ ) and SSP representation. Each line on this plot corresponds to one NNI run, where the values for each parameter are indicated by where that line crosses each vertical axis. The final axis (N Trials, far right) as well as the colour of the lines shows the number of trials needed for that run to reach the goal rolling mean reward.



Figure 7: Point plot showing the mean number of trials needed to reach the goal rolling mean reward, and 95% confidence intervals, for each experiment using different N dimensions in the SSP representation.

representation methods (Table 2). Where TD(0) was used, the minimum number of trials needed was 149 and 127 for the baseline and one-hot configurations, compared with 108 and 105 for the random SSP and grid-cell configurations (respectively). Similarly, with TD( $\lambda$ ) the baseline method required a minimum of 134 trials compared with 102 (random SSP) and 99 (grid cells).

In contrast, in Bartlett et al. (in press) we found that, following a manual search of the parameter space, grid cells markedly out-performed all three of the other representation methods regardless of learning rule. A manual search of the parameter space was able to identify a set of parameters such that, when using the TD( $\lambda$ ) learning rule, the grid cell network was able to achieve the goal rolling mean reward in an average of 105.4 trials, and 122.2 trials when using TD(0) (with the next fastest configurations achieving an average of 142.8 and 156.6 trials respectively). This is comparable to the 99–103 trials identified in the present study. However, fol-



Figure 8: Point plot showing the mean number of trials needed to reach the goal rolling mean reward, and 95% confidence intervals, for each experiment using different numbers of neurons in the hidden layer.

lowing the optimization carried out in the present study, we found that the advantage of grid cells over the other representation methods was much smaller than previously indicated. It is still worth noting, though, that whilst the baseline and one hot approaches do perform well once optimized, it seems that grid cells do not necessarily require optimization.

It should be noted that the Mini Grid task used in this study is fairly simple, so whilst the current study does not necessarily indicate a huge advantage of using grid cells over other methods, previous findings that grid cells do result in faster learning (Gustafson & Daw, 2011) suggests that when tested on more complex tasks, performance with grid cells may deviate more from non-biologically inspired methods.

#### **Online Resources**

Experiment and analysis scripts can be found in the github repository (https://github.com/maddybartlett/ Fast\_RL\_with\_Bio\_Based\_Reps).

#### References

- Banino, A., Barry, C., Uria, 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., Stewart, T., & Orchard, J. (in press). Biologically-based neural representations enable fast online shallow reinforcement learning. In *Cogsci.*
- Chevalier-Boisvert, M., Willems, L., & Pal, S. (2018). *Minimalistic gridworld environment for openai gym.* https:// github.com/maximecb/gym-minigrid. GitHub.
- Dumont, N., & Eliasmith, C. (2020). Accurate representation for spatial cognition using grid cells. In *Cogsci*.
- Eliasmith, C., & Anderson, C. H. (2003). *Neural engineering: Computation, representation, and dynamics in neurobiological systems.* MIT press.
- Gayler, R. W. (2004). Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience. *arXiv preprint cs/0412059*.
- Gustafson, N. J., & Daw, N. D. (2011). Grid cells, place cells, and geodesic generalization for spatial reinforcement learning. *PLoS Computational Biology*, 7(10), e1002235.
- Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., & Moser, E. I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052), 801–806.
- Komer, B., Stewart, T. C., Voelker, A., & Eliasmith, C. (2019). A neural representation of continuous space using fractional binding. In *Cogsci* (pp. 2038–2043).
- Mackintosh, N. J. (2019). Classical and operant conditioning. In *Companion encyclopedia of psychology* (pp. 379–396). Routledge.
- Microsoft. (2021, 1). Neural Network Intelligence. Retrieved from https://github.com/microsoft/nni
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

#### Towards a Method for Evaluating Convergence Across Modeling Frameworks

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

Glenn Gunzelmann (glenn.gunzelmann@us.af.mil) Air Force Research Laboratory

Wright Patterson AFB, OH USA

#### Abstract

Model convergence is an alternative approach for evaluating computational models of cognition. Convergence occurs when multiple models provide similar explanations for a phenomenon. In contrast to competitive comparisons which focus on model differences, identifying areas of convergence can provide evidence for overarching theoretical ideas. We proposed criteria for convergence which require models to be high in predictive and cognitive similarity. We then used a cross fitting method to explore the extent to which models from distinct computational frameworks-quantum cognition and the cognitive architecture ACT-R-converge on explanations of the interference effect. Our analysis revealed the models to be moderately high in predictive similarity but mixed for cognitive similarity. Though convergence was limited, the analysis suggests that interference effects emerge from interactions between uncertainty and the degree to which an individual relies on typical cases to make decisions. This result demonstrates the utility of convergence analysis as a method for integrating insights from multiple models.

**Keywords:** ACT-R; Quantum cognition; Interference effects; Model convergence

#### Introduction

Model comparison often proceeds as a zero-sum game in which two or more models offering different explanations make opposing predictions. The winner of such competitions is assumed to offer a more convincing representation of the underlying cognitive processes. Although competitive comparisons can be useful to varying degrees, one potential limitation is that one may overlook areas of convergence by focusing exclusively on differences between models. Two models may point to similar conclusions for a particular empirical phenomenon even though they may differ in other regards. One important benefit of convergence is that confidence in an explanation will increase when two models are in agreement. As an example of convergence, two distinct computational frameworks, one based on the Adaptive Control of Thought-Rational (ACT-R) and the other based on the drift diffusion model-provided similar explanations for the deleterious effect of sleep loss on performance. Namely, they both explain a reduction in the signal-to-noise ratio and a reduction in response inhibition (Walsh et al., 2017).

Convergence offers an alternative approach for evaluating what models reveal about human cognition (Gunzelmann, Christopher R. Fisher (christopher.fisher.27.ctr@us.af.mil)

Cubic Defense Beavercreek, OH 45324 USA

Leslie M. Blaha (leslie.blaha@us.af.mil) Air Force Research Laboratory Wright Patterson AFB, OH USA

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

2019). The present study extends the existing work by elaborating upon the definition of convergence and its implications for theoretical correspondence. We then conduct an exploratory evaluation of the extent to which two distinct models of interference effects—an existing quantum cognition model and a model developed in ACT-R—converge on conclusions consistent with a single theoretical perspective.

#### **Model Convergence**

As shown in Figure 1, models can be compared along two orthogonal dimensions: predictive similarity and cognitive similarity. Predictive similarity is the degree to which the predictions of two models follow the same pattern. At minimum, we require the predictions to follow the same qualitative pattern, i.e., both models predict an effect in the same direction. Cognitive similarity is defined as the degree to which two models posit similar mental representations (i.e., the content and organization of information about the external environment) and/or cognitive processes (i.e., how information is transformed, manipulated, and combined) that are relevant for a particular empirical phenomenon. Although this space is continuous, it can be helpful to refer to prototypical examples or describe the space more coarsely as quadrants. Convergence occurs when two or more models are highly similar along both dimensions.



Figure 1: Four points in the space spanned by predictive similarity and cognitive similarity. Point A represents competitive comparisons and Point D represents convergence.

The left half of Figure 1 reflects so-called "zones of contention" where competing models propose different mental representations and/or cognitive processes to explain an empirical phenomenon (McClelland, 2009). At Point A in the bottom-left quadrant, critical tests can distinguish between competing models on the basis of their opposing predictions. By contrast, for Point B in which predictions are similar, different mental representations and cognitive processes cannot be distinguished on the basis of their predictions. Indeed, such ambiguity often leads to the development of the critical tests conducted in bottom-left quadrant, a cycle that can repeat itself many times (Gunzelmann, 2019). The right half represents cases in which models are cognitively similar and thus propose similar mental representations and/or cognitive processes. Point C in the bottom-right quadrant represents an unusual situation in which two models high in cognitive similarity yield differing predictions. In this case, the models provide contradictory evidence for a common explanation. Point D in the top-right quadrant represents the case where models converge on a common explanation: both models rely on similar mental representations and/or cognitive processes and make similar predictions. When convergence occurs, we find more evidence for an explanation than we would otherwise. We believe viewing model comparisons through the lens of convergence adds clarity to theoretical implications and may provide additional evidence for an overarching theory. By contrast, the competitive approach seeks to refute one of the models. Although both approaches have different goals, taken together, they offer complementary methods for evaluating theoretical support (Gunzelmann, 2019).

#### **Current Application**

We explore whether quantum cognition and ACT-R provide converging explanations of the interference effect. Interference effects emerge when uncertainty about an event changes the marginal probability of a subsequent decision, resulting in a violation of the law of total probability (Wang & Busemeyer, 2016). The models we investigated derive from highly disparate computational frameworks with strong empirical support. The belief-action entanglement model is based on the mathematical formalism of quantum probability which has been used to explain several empirical phenomena where models based on classical probability generally fail (Wang & Busemeyer, 2016). By contrast, ACT-R is a cognitive architecture in which cognition emerges from interactions between specialized information processing modules for declarative and procedural memory, perception, and action among others (Anderson et al., 2004). Given that both frameworks have withstood many rounds of empirical testing, one might expect points of convergence to emerge.

#### **Categorization-Decision Paradigm**

One popular paradigm for studying interference effects emerging from the interactions of categorization and decision making is the categorization-decision paradigm (Wang & Busemeyer, 2016). On each trial, a face from a "good" category or a "bad" category is presented, and participants must decide whether to attack or withdraw. Each face consisted of either a g-type or b-type feature, which were typically associated with the good category or bad category, respectively. Further, participants were typically rewarded for attacking a bad category and withdrawing from a good category. However, these associations were probabilistic, and atypical associations occurred in some trials.

Uncertainty about the category was manipulated across three conditions to elicit an interference effect. In the decisiononly (d) condition, no category information was provided prior to the decision to act, and categorization was presumed to occur implicitly (Wang & Busemeyer, 2016). In the categorizethen-decide (cd) condition, participants were asked to selfcategorize the feature then decide upon an action. In the third explicit-categorization (xd) condition, the true category was provided prior to the action decision.

#### **Belief-Action Entanglement Model**

The belief-action entanglement (BAE) model is a quantum cognition model of interference effects in categorization and decision making. A full mathematical description of the model can be found in Wang & Busemeyer (2016). In the BAE model, states evolve within a finite Hilbert space H (N-dimensional universal vector space) across a field of complex numbers. The potential of a state is given by the unit-length vector  $\Psi$ . A defining feature of quantum systems, to include cognitive systems, is that a measurement changes the state. Consequently, transitions occur when  $\Psi$  is measured, e.g., a feature is categorized or an action is selected.

The BAE represents category-action events as basis states where GW symbolizes the combined event of categorizing a feature as good then deciding to withdraw. The initial state  $\psi_f$  is uncertain and superposed over the four possible basis states,  $\psi_f = [GW, GA, BW, BA]^{\top}$ . Basis states are assigned amplitudes such that the square magnitude gives its probability:  $|\psi_{GW}|^2 = \Pr(GW)$ . The parameter *j* governs the probability a b-type or g-type feature will be judged as belonging to either category, e.g., for a b-type feature,  $\psi_f =$  $\psi_b = \frac{1}{2} \left[ \sqrt{1-j}, \sqrt{1-j}, \sqrt{j}, \sqrt{j} \right]^{\top}$ .

Prior to action evaluation, the state remains in the superposed  $\Psi_f$  in the d condition. In cd and xd, the state transitions to either being in the good or bad category. After transitioning to the bad category, as an example, the state is updated to  $\Psi_f \rightarrow \Psi_b = \frac{1}{2} [0,0,1,1]^{\top}$ , where the latter values represent BW and BA and the state is only superposed over the actions.

During action evaluation, the state transitions according to the reward rate and utility parameters which influence the probability of a action given a feature and category. For example,  $\mu_{b,b}$  is the utility for attacking a b-type feature categorized as bad. The transition to the final action state is computed using a separate unitary matrix for each feature type. When the category is ambiguous as in cd and d, the transition includes the entanglement parameter  $\gamma$  which amplifies amplitudes for typical category-action events, e.g., GW and BA, and attenuates amplitudes for atypical events, e.g. GA and BW. Alternately,  $\gamma$  has no effect in xd because the true category is known. Consequently, the BAE model predicts that interference effects emerge from differences in the utilities for each feature type and the influence of  $\gamma$  on uncertain states.

#### **ACT-R Model**

We developed a memory-based ACT-R model of the interference effect. and focus our description on the declarative memory system.

#### **Declarative Memory**

In ACT-R, the basic unit of declarative knowledge is a set of slot-value pairs called a chunk:  $\mathbf{c}_m = \{(s_i, v_i)\}_{i \in I_m}$ , where  $s_i$  and  $v_i$  are the slot and value of pair *i*, and  $I_m$  is the index set for slot-value pairs of chunk *m*. We will use the set  $Q_m = \{s_i\}_{i \in I_m}$  to denote a set of slots (e.g., domain) in  $\mathbf{c}_m$ . The mapping from slots to values is defined as  $c_m(s) = v$ , where *v* is null if the chunk does not include *s*.

The set of slots for each chunk is defined as  $Q = \{\text{feature, category, action}\}$ , where the feature can be b-type or g-type, the category can be good or bad and the action can be attack or withdraw. Declarative memory M consists of  $2^3 = 8$  chunks formed by permuting the possible values for feature, category and action. For example,  $\mathbf{c}_{\text{gba}} = \{(\text{feature, g-type}), (\text{category, bad}), (\text{action, attack})\}$  is a chunk for attacking a g-type face in the bad category. We will use a three letter abbreviation, such as gba, to denote the feature, category, and action values of a chunk.

#### **Memory Activation**

Each chunk is associated with an activation value representing its ability to be retrieved. As activation increases, the probability of retrieval increases. We omit the base-level learning mechanism because learning was not observed in Wang & Busemeyer (2016). Activation is defined as  $a_m = \beta_m + \rho_m + \varepsilon_m$ where  $\beta$  is the base-level constant,  $\rho$  is the partial matching term, and  $\varepsilon \sim \text{logistic}(0, s)$  is logistically distributed noise with scalar parameter *s*. The partial matching mechanism allows chunks that do not match the retrieval request **r** to be retrieved as a decreasing function of mismatch. The retrieval request is treated as a chunk with slot-value pairs. We use a binary mismatch penalty function:  $\rho_m = -\delta \sum_{q \in Q_r} I(c_m(q), r(q))$ , where  $\delta$  is the mismatch penalty parameter,  $Q_r$  is the set of slots in the request, and *I* is an indicator function which returns 1 when both inputs are not equal and returns 0 otherwise.

#### **Retrieval Process**

Upon stimulus presentation, a retrieval request **r** based on all available information is submitted to declarative memory. For example, in the d condition, only the feature is available, but in the xd condition both the feature and the category are available. The chunk with the highest activation value above the retrieval threshold  $\tau$  is retrieved and determines the eventual response. To simplify the model, we set the retrieval threshold to -10

under the assumption that chunks are sufficiently active to be retrieved.

#### **Model Predictions**

In the predictions for each condition below, we use A to denote a random variable for the action, F to denote a random variable to denote the feature, and C as a random variable to denote the category.

**d** condition Participants decided to attack or withdraw from a face with feature f. The retrieval request is  $\mathbf{r} = \{(\text{feature}, f)\}$ . We will define  $R_d$  as the set of chunks that map to a decision to attack  $R_d = \{c_m \in M : c_m(\text{feature}) = r(\text{feature}), c_m(\text{action}) = \text{attack}\}$ . In other words,  $R_d$  is the set of chunks that match feature f and have a value "attack" for the action slot. The approximate probability of attacking is computed using the soft max function (Weaver, 2008):

$$\Pr(A = a \mid F = f) = \frac{\sum_{k \mid \mathbf{c}_k \in R_d} e^{\mu_k / \sigma}}{\sum_{j \mid \mathbf{c}_j \in M} e^{\mu_j / \sigma}}$$
(1)

where  $\sigma = s\sqrt{2}$  and the expected activation is  $\mathbb{E}[a_m] = \mu_m$ .

**xd condition** Participants were told the true category v for a face with feature f then decided to attack or withdraw, leading to the retrieval request  $\mathbf{r} = \{(\text{feature}, f), (\text{category}, v)\}$ . The set of chunks that map to the decision to attack is defined as:  $R_{\text{xd}} = \{c_m \in M : c_m(\text{feature}) = r(\text{feature}), c_m(\text{category}) = r(\text{category}), c_m(\text{action}) = \text{attack}\}$ . The probability of attacking a face with feature f in category v is given by:

$$\Pr(A = a \mid F = f, C = v) = \frac{\sum_{k \mid \mathbf{c}_k \in R_{\mathrm{xd}}} e^{\mu_k / \sigma}}{\sum_{j \mid \mathbf{c}_j \in M} e^{\mu_j / \sigma}}$$
(2)

**cd condition** Participants categorized a face with feature f as good or bad followed by a separate response to attack or withdraw. The retrieval request for the categorization is  $\mathbf{r}_c = \{(\text{feature}, f)\}$ . The set of chunks that map to a category response v is defined as  $R_{\text{cd},c} = \{c_m \in M : c_m(\text{feature}) = r_c(\text{feature}), c_m(\text{category}) = v\}$ . The probability of categorizing face with feature f as v is given by:

$$\Pr(C = v \mid F = f) = \frac{\sum_{k \mid \mathbf{c}_k \in R_{\text{cd},c}} e^{\mu_k / \sigma}}{\sum_{j \mid \mathbf{c}_j \in M} e^{\mu_j / \sigma}}$$
(3)

The judged category *v* is incorporated into the retrieval request for the subsequent decision:  $\mathbf{r}_d = \{(\text{feature}, f), (\text{category}, v)\}$ . The set of chunks that map to the decision to attack is the same as in the cd condition:  $R_{\text{xd}} = R_{\text{cd,d}}$ , which implies that the probability of attacking a face with feature *f* categorized as *v* is equal to equation 2 from the xd condition.

#### **Cross Fitting**

To measure predictive and cognitive similarity, we used a cross fitting method inspired by Donkin et al. (2011). In our study, predictive similarity is measured by comparing the qualitative predictions of the two models, whereas cognitive
similarity is measured by assessing the mapping of parameters from one model to another. Our cross fitting method entails two steps. First, we generated predictions from the BAE model by varying one parameter at a time while holding the others constant at their best fitting values reported in Wang & Busemeyer (2016). Second, we fit the ACT-R model to the predictions of the BAE by minimizing Kullback-Leibler divergence (KLD; Kullback & Leibler, 1951). KLD is the amount of information lost by using one distribution in place of another, i.e., how much information is lost when using the best fit ACT-R model to represent the BAE mode. One advantage of comparing two probability distributions using KLD instead of fitting a model to a finite sample of simulated data is that it eliminates the role of noise in the mapping.

We selected three parameters on the basis of their qualitatively distinct roles in the model: the entanglement parameter,  $\gamma$ , the category judgement parameter, j, and a utility parameter,  $\mu_{b,b}$ . Each parameter was varied across 20 equally spaced values:  $\gamma \in [0,2]$ ,  $j \in [.01,.99]$ , and  $\mu_{b,b} \in [-1,1]$ . We set s = .2and base level constants  $\beta_{bbw} = 0.0$  and  $\beta_{ggw} = 0.2$  to ensure identifiability of the model parameters. We used differential evolution to minimize KLD.

#### **Convergence Predictions**

Psychologically, interference effects can result from increased on reliance typical associations in the absence of certain information (Fiske & Taylor, 1991). For example,  $\beta_{bba}$  and  $\mu_{b,b}$ represent the influence of typical associations between a face with a b-type feature in the bad category and the decision to attack. The strength of influence varies with certainty about the category. If convergence is present, the BAE and ACT-R accounts of these processes should be relatable.

First, we expect typical associations to strengthen the probability to attack, Pr(A), for b-type features in both models, irrespective of category certainty. In the BAE, this should be most evident as  $\mu_{b,b}$  increases. In ACT-R, we expect to observe a comparable increase  $\beta_{bba}$  with a commensurate decrease in  $\beta$ s for atypical associations according to equations 1 and 2. Second, we expected category uncertainty in the d condition to moderate the Pr(A). In the BAE, changes in *j* should vary the influence of typical associations. Because the retrieval request only contains the feature, a comparable process in ACT-R should systematically influence  $\beta$ s for typical categories and actions according to equation 1.

Third, we expect  $\gamma$  and  $\delta$  parameters to modulate the influence of utility and  $\beta$  parameters, respectively. In particular, we expect  $\gamma$  to amplify the effect of typical associations for the Pr(*A*), but only with category uncertainty in cd and d. In ACT-R, the analogous effect should occur at higher values of  $\delta$  which increase the probability of selecting an exact match.Consequently, we expect the influence of  $\beta_{bba}$  to be amplified while attenuating the influence of atypical  $\beta$ s.

#### Results

We assess predictive and cognitive similarity between the BAE and ACT-R for each of the three BAE parameters to determine

Figure 2: Best fitting values for the base level constant ( $\beta$ ) parameters and the mismatch penalty parameter ( $\delta$ ) for ACT-R as a function of  $\gamma$  from the BAE model.



Figure 3: Response probabilities for the BAE (red) and ACT-R (black) models as a function of  $\gamma$  paneled by response category. Subplot titles give condition, face type, category.



whether the models converge on similar explanations of the interference effect.

The entanglement parameter  $\gamma$ . Predictive similarity: for both models, the response probabilities follow qualitatively similar patterns in cd and d, a distinction more pronounced in cd than d (see Figure 3). However, Pr(A) patterns are qualitatively dissimilar in xd. Specifically, the BAE model is invariant to  $\gamma$ , as intended, whereas as ACT-R simply reproduces pattern of probabilities in cd. This is not surprising as equation 2 computes the Pr(A) in both xd and cd. The results indicate predicative similarity is moderately high for ambiguous category knowledge but low for unambiguous categorization.

Cognitive similarity: for simplicity, we focus on mappings where  $\gamma$  is less than 1 (see Figure 2). Though the pattern is not strictly linear, decreases in  $\gamma$  and increases in  $\delta$  favor typical associations, as predicted. In ACT-R specifically, the Figure 4: Best fitting values for the base level constant ( $\beta$ ) parameters and the mismatch penalty parameter ( $\delta$ ) for ACT-R as a function of *j* from the BAE model.



Figure 5: Response probabilities for the BAE (red) and ACT-R (black) models as a function of j paneled by response category. Subplot titles give condition, face type, category.



process entails expected interactions with  $\beta_{bba}$  as well as the atypical  $\beta_{gbw}$ . As a result, we conclude the models exhibit high cognitive similarity for modulating bias.

The category judgement parameter *j*. Predictive similarity: The BAE and ACT-R produced identical distributions for Pr(A) in d (see Figure 5). By contrast, Pr(A) in xd and cd remained invariant in both models, indicating they were constrained to the d condition, as expected. All told, these patterns indicate high predictive similarity.

Cognitive similarity: The relatively linear decreases in  $\delta$  and atypical  $\beta$  values with increases in the *j* parameter reveal an unexpected mapping between the two models (see Figure 4). In the BAE model, the Pr(*A*) derives from an uncertain superposition state over possible outcomes and is systematically modulated by *j*. In contrast, the ACT-R model is less systematic and it is not clear the mental states represented by

parameter interactions. Specifically, the Pr(A) increases at high values of  $\delta$ , which approximates increasing bias in the decision, and also at low values of  $\delta$ , ostensibly representing indecision between alternatives. Because ACT-R's varied account cannot easily be reconciled with the BAE account, we conclude the models are low in cognitive similarity when category is uncertain and not made explicit.

The utility parameter  $\mu_{b,b}$ . Predictive similarity: Visual inspection of Figure 7 indicates that predictive similarity is high when  $\mu_{b,b}$  is varied. The predictions exhibit some discrepancy for b-type faces in the bad category in the xd and cd conditions. Nonetheless, the predictions are qualitatively similar throughout.

Cognitive similarity: The varied behavior of ACT-R parameters across the range of  $\mu_{b,b}$  was surprising (see Figure 6). In the BAE,  $\mu_{b,b}$  exerts a relatively linear effect on the Pr(*A*), as expected. In ACT-R, the Pr(*A*) varies with parameter interactions when  $\mu_{b,b}$  is above versus below 0. Specifically, when  $\mu_{b,b} > 0$ ,  $\beta_{bba}$  amplifies the Pr(*A*) when an exact match is more probable (e.g., at higher  $\delta$  values), in line with our expectations. Alternatively when  $\mu_{b,b} < 0$ , the atypical  $\beta_{gbw}$ increasingly attenuates the Pr(*A*) when a mismatch becomes more likely (e.g., at lower  $\delta$  values) which was neither expected nor a predictable function of  $\delta$ . Because only a portion of ACT-R interactions are analogous  $\mu_{b,b}$ 's function, cognitive similarity between the models was deemed moderate, at best, for the influence of typical associations.

Figure 6: Best fitting values for the base level constant ( $\beta$ ) parameters and the mismatch penalty parameter ( $\delta$ ) for ACT-R as a function of  $\mu_{b,b}$  from the BAE model.



#### Discussion

The goal of the present study was two-fold. First, we elaborated upon the definition of model convergence. Second, we explored the extent the BAE, a model based in quantum cognition, and a model based in ACT-R provide converging explanations of the interference effect. Our criteria for convergence required models be both high in predictive and cognitive similarity. For interference effects, we expected similarities to emerge from interactions between category certainty and the influence of typical associations on decisions.

Both models exhibited moderately high predictive similarity. Predictions were more similar when the category was uncertain (cd and d) but diverged when the category was certain (xd). In ACT-R, the divergence can be attributed to the partial



Figure 7: Response probabilities for the BAE (red) and ACT-R (black) models as a function of  $\mu_{b,b}$  paneled by response category. Subplot titles give condition, face type, category.

matching mechanism which is constrained to implementing penalties for mismatches in slot-value pairs. The architecture does not permit penalizing mismatches at the condition level. Hence, when the category was included in the retrieval request, the model was unable to differentiate between an uncertain category in cd and a true one in xd. The BAE can account for unambiguous category knowledge because the entanglement parameter  $\gamma$  was not applied to state transitions in xd (Wang & Busemeyer, 2016).

Cognitive similarity between the two models was mixed. The BAE's  $\gamma$  parameter and ACT-R's mismatch penalty  $\delta$  modulated the influence of typical and atypical associations in comparable ways. Overall, we found the expected relationship between  $\mu_{b,b}$  and  $\beta_{bba}$ . However, for  $\mu_{b,b}$  and *j*, ACT-R parameter mappings were by determined by the ratio of  $\beta$ s (see equations 1, 2, 3) and varying values of  $\delta$  which at times appeared unsystematic and difficult to predict. The variability is surprising given that both the BAE and ACT-R models produce interference effects and can account for violations of total probability. One explanation for the unexpected mappings may be due to the idiosyncrasy of a particular implementation rather than the function of partial matching. If so, then cognitive similarity may be higher than assessed.

Indeed, while useful, our cross fitting analysis may have obscured areas of cognitive similarity. First, our mappings were asymmetrical such that ACT-R parameters were cross fitted as a function of the BAE parameters but not the other way around. However, ACT-R's fluctuating parameter interactions pose a challenge for symmetrical mappings, and it is unclear whether mapping to a single parameter would be sufficient to evaluate convergence. Second, our mappings centered on best fitting values, ergo limiting the scope of our analysis; the full space of potential convergence was not explored. Evidence for similarity would be greater if the relationships hold across a larger sub-space of parameters. Even so, our approach of evaluating parameters near the best fitting is a reasonable starting point.

With respect to supporting a single theoretical perspective, our analysis was informative, even as convergence was limited. Had we conducted a competitive comparison, the theoretical contribution of the losing model might have been eclipsed. As it stands, not only have we accumulated evidence for the psychological processes underlying interference effects, but our analysis identified areas where future research can further elucidate how and when the human mind is influenced by the strength of beliefs in uncertain situations.

### 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 by Air Force Office of Scientific Research grant 21RHCOR080. Approved for public release. Case number: AFRL-2022-2018.

### 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.
- Donkin, C., Brown, S., Heathcote, A., & Wagenmakers, E.-J. (2011). Diffusion vs linear ballistic accumulation: Different models but same conclusions about psychological processes? *Psychonomic Bulletin & Review*, 18(1), 61–69.
- Fiske, S. T., & Taylor, S. E. (1991). *Social cognition*. Mcgraw-Hill Book Company.
- Gunzelmann, G. (2019). Promoting cumulation in models of the human mind. *Computational Brain & Behavior*, 2(3), 157–159.
- Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1), 79–86.
- McClelland, J. L. (2009). The place of modeling in cognitive science. *Topics in Cognitive Science*, *1*(1), 11–38.
- Walsh, M. M., Gunzelmann, G., & Van Dongen, H. (2017). Computational cognitive modeling of the temporal dynamics of fatigue from sleep loss. *Psychonomic Bulletin & Review*, 24(6), 1785–1807.
- Wang, Z., & Busemeyer, J. R. (2016). Interference effects of categorization on decision making. *Cognition*, 150, 133– 149.
- Weaver, R. (2008). Parameters, predictions, and evidence in computational modeling: A statistical view informed by ACT-R. *Cognitive Science*, 32(8), 1349–1375.

# Leveraging Cognitive Models for the Wisdom of Crowds in Sequential Decision Tasks

Erin H. Bugbee (ebugbee@cmu.edu) Chase McDonald (chasemcd@cmu.edu) Cleotilde Gonzalez (coty@cmu.edu) Department of Social and Decision Sciences, Carnegie Mellon University Pittsburgh, PA 15213 USA

Abstract

Many decisions we face in life are sequential, where alternatives appear over time. We often must decide whether to take the opportunity and stop searching or to continue evaluating potentially better future alternatives. Humans are notoriously poor at stopping optimally in sequential decision-making tasks. These sequential decisions are difficult because they involve the consideration of how past, present, and future decisions affect the outcome. Recent research suggests that the wisdom of the crowd (WoC) - that is, aggregated decisions of many people that outperform most individuals - can be applied to sequential decision tasks and potentially help improve stopping decisions. Current models rely on a process of fitting human data, making it difficult to understand how those individuals would behave in new problems. Furthermore, these models do not account for the learning process that humans experience while making these decisions. In this work, we demonstrate how simulated agents using a cognitive model derived from Instance-Based Learning Theory (IBLT) can produce WoC that is similar to WoC from human participants in two sequential decision tasks. We demonstrate that the WoC performance from simulated groups of agents is better than the performance of most agents and that the Instance-Based Learning (IBL) crowd behavior is similar to the human crowd behavior. Thus, cognitive models that account for learning and experience can be used to inductively predict the behavior of human crowds in sequential decision tasks.

**Keywords:** wisdom of crowds; sequential decision making; cognitive modeling; instance-based learning

#### Introduction

Sequential decision making is ubiquitous in everyday life. As we navigate the world and make decisions, we often do not face all possible alternatives at once. Instead, alternatives emerge over time, and to maximize benefits, a choice must involve the selection of an alternative at the right time, before the opportunity disappears. For example, to select a rental apartment in a dynamic market, one must decide when to stop visiting new possibilities and make an offer before the current option becomes unavailable.

The literature on sequential decisions has underscored that people are often suboptimal in making stopping decisions in sequential tasks, given the tradeoffs of risk and uncertainty (Lejuez et al., 2002; Lee, 2006; Guan, Stokes, Vandekerckhove, & Lee, 2020; Guan, 2019; Bugbee & Gonzalez, 2022b). Recently, the possibility of using the Wisdom of Crowds (WoC) has been suggested as a way to address these difficulties in sequential decision problems (Thomas, Coon, Westfall, & Lee, 2021). The WoC (Surowiecki, 2005) suggests that the aggregation of individual estimates or decisions can outperform most of the individuals in the crowd, and a significant amount of work has demonstrated that the aggregation of collective wisdom can be beneficial in a large number of tasks. However, the benefits of WoC for sequential decision tasks have only recently been suggested (Thomas et al., 2021). The idea is to aggregate the answers from a group of individuals in each choice of a sequence to produce a crowd answer (e.g., whether to stop exploring or not), and such aggregate would produce an answer closer to the optimal stopping point compared to individual decisions.

In their work, Thomas et al. (2021) aggregate individual predictions to retrieve WoC predictions. These WoC predictions, along with the individual responses, are then compare to the predictions of cognitive models at the individual and crowd level. They used statistical models of individuals to provide model-based predictions, showing that the aggregation of these predictions can result in accurate behavior in not only problems that individuals completed, but also new problems that participants did not previously experience. The models that Thomas et al. (2021) present are descriptive statistical models, where the parameters of the models are fit to the data of individuals, and these parameter values are then used to generalize to new problems within the same class of sequential decision-making problems. These are not process models that represent the individual learning through a sequence; and thus, they would fail to account for behavior in situations in which people learn from past choices.

In this research, we build on the work of Thomas et al. (2021) to test the benefits of WoC using cognitive models. We rely on two known sequential decision tasks and their previously collected data sets (Guan, 2019; Guan et al., 2020). Further, we utilize two existing cognitive models of sequential decisions in these tasks (Bugbee & Gonzalez, 2022a, 2022b). In contrast to the work of Thomas et al. (2021), these cognitive models are generative process models that learn through experience to produce predictions of human stopping decisions in the absence of human data. These models act based on a theory of decisions from experience, Instance-Based Learning (IBL) Theory (Gonzalez, Lerch, & Lebiere, 2003), and are able to replicate human sequential decisions closely. The question in this research is whether the WoC predictions in groups of agents generated with IBL models result in similar values as the groups of human participants in the same sequential decision making tasks. The replication of the human crowd behavior in addition to the individual human behavior has significant benefits for applying the WoC to sequences of decisions in new situations for which human data might not exist.

### **Sequential Decision Tasks and Data Sets**

For this work, we used experimental data previously collected by Guan (2019) and Guan et al. (2020) in two sequential decision tasks: the Balloon Analog Risk Task (BART) and the Optimal Stopping Task.

### Balloon Analog Risk Task (BART)

BART is a sequential decision making task in which a decision maker inflates a balloon. The level of inflation corresponds to the reward that the decision maker can receive. At each time point, the decision maker decides whether to pump the balloon and increase its value or bank the current monetary amount. However, with each pump of the balloon, there is a probability that the balloon bursts, causing the decision maker to receive a reward of 0 for that problem. This leads to the need to balance exploring through pumping with exploitation through banking, with the goal of maximizing total reward. Each balloon has a predefined burst time generated from the constant probability of bursting, although participants are not told these probabilities.

In the experiment from Guan et al. (2020), 56 participants completed the BART in a within-subjects design. Participants were presented balloons with a fixed probability of bursting with each pump (either P(Burst) = 0.1 or P(Burst) = 0.2)<sup>1</sup>. Every participant completed 50 problems with each probability, and the order of the problems and conditions was randomized between participants. Each problem started with a balloon with a hypothetical value of \$1. For each decision, the participant had the option to pump the balloon ("Pump") and increase its monetary value by \$1, or stop ("Bank") and collect the current monetary value. However, each pump action risks bursting the balloon, which results in collecting \$0 for that problem. The participant continued making Bank or Pump decisions until either the balloon burst or the participant chose the Bank action and collected the money. The stated goal was to maximize the total reward on all problems. Participants were compensated for their time but were not rewarded based on their performance.

### **Optimal Stopping Task**

In the Optimal Stopping Task, the same 56 participants from Guan et al. (2020) were presented with sequences of cats. They were instructed to choose the cat in the sequence with the highest weight. Participants were presented with cats sequentially and were required to "Select" or "Pass' each cat. Once passed, the cat could not be returned to. The participants were instructed that the last cat in the sequence must be chosen if none is chosen prior. If they chose the cat with the



Figure 1: A screenshot of the BART experiment obtained from Guan (2019).

maximum weight, they were correct, and otherwise they were incorrect. Participants received feedback on the accuracy of their choice, but not about unseen cats.

The task had a within-subjects design. Each participant experienced four conditions, in which both the distribution of weights and the length of the sequence were varied. Weights ranged between 0 and 100 pounds, according to either a uniform (i.e., "neutral") or beta(4,2) (i.e., "plentiful", as weights are skewed toward higher weights) distribution scaled to 0 to 100. The sequences consisted of 4 or 8 cats. Participants were told the length of the sequence but not the distribution of cat weights.

All participants completed a group of 40 problems within each condition with a randomized problem order among the participants. The order of conditions was also randomized among participants.



Figure 2: A screenshot of the Optimal Stopping Task experiment obtained from Guan (2019).

### **Instance-Based Learning Theory**

We use cognitive models based on Instance-Based Learning Theory (IBLT) recently implemented in Bugbee and Gonzalez (2022b) and Bugbee and Gonzalez (2022a). IBLT outlines a theory of decisions from experience, derived from mechanisms proposed in the ACT-R cognitive architecture (Anderson & Lebiere, 2014). The theory was developed to explain human learning in dynamic environments (Gonzalez et al., 2003). It provides an algorithm for learning from experience and making decisions, which can be used to implement a computational model of these processes that simulates human behavior.

There are three primary components of the decision making algorithm: recognition and retrieval of past instances, as

<sup>&</sup>lt;sup>1</sup>This deviates from the typical BART design (e.g. Lejuez et al. (2002)), in which the probability of the balloon bursting increases as the number of pumps increases.

a function of their similarity to a current decision; calculation of the expected utility of decision alternatives, and a choice rule that allows for generalization from past experience. Past instances are stored in memory and are effectively memory units consisting of situations  $s \in S$ , decisions  $a \in A$ , and the realized utility x of taking action a after observing situation s. An option is defined as k = (S,A): making a decision A in the situation S.

At time *t*, there are  $n_{k,t}$  different generated instances  $(k, x_{i,k,t})$  for  $i = 1, ..., n_{k,t}$ , corresponding to selecting *k* and achieving the outcome  $x_{i,k,t}$ . Each instance *i* in memory has an activation value, which represents how readily available this information is in memory, and is determined by similarity to past situations, recency, frequency, and noise (Anderson & Lebiere, 2014). The activation is described by Equation 1, for option *j*, when presented with option *k* (that is, the current situation is described by *k*):

$$\Lambda_{i,k,j,t} = \ln\left(\sum_{t'\in T_i} (t-t')^{-d}\right) + \alpha S(k,j) + \sigma \ln\frac{1-\xi}{\xi} \qquad (1)$$

where

$$S(k,j) = \sum_{j} Sim_j(f_j^k, f_j^j)$$
<sup>(2)</sup>

and  $\alpha$ , *d* and  $\sigma$  are the mismatch penalty, decay, and noise parameters, respectively. Furthermore,  $T_i \subset \{0, ..., t-1\}$  is the set of previous timestamps in which the instance *i* was observed and *Sim<sub>j</sub>* is a similarity function that calculates the similarity of the *j*th attribute of an option *k*,  $f_j^k$ . The rightmost term represents Gaussian noise to capture individual variation in activation, and  $\xi$  is a random number drawn from a uniform distribution U(0,1) at each time step and for each instance and option.

The probability of retrieving an instance *i* from memory is a function of its activation  $\Lambda_{i,k,j,t}$  relative to the activation of all instances:

$$p_{i,k,j,t} = \frac{\exp(\frac{\Lambda_{i,k,j,t}}{\tau})}{\sum_{j=1}^{n_{j,t}}\exp(\frac{\Lambda_{j,k,j,t}}{\tau})}$$
(3)

where  $\tau$  is the temperature parameter. As  $\tau \to 0$ , the selection of actions is deterministic, and as  $\tau \to \infty$ , all actions become equally likely.

The expected utility of option k is given by the *blending* mechanism calculated as in Gonzalez and Dutt (2011):

$$V_{k,t} = \sum_{i=1}^{n_{j,t}} p_{i,k,j,t} x_{i,j,t}.$$
(4)

The blending operation (Equation 4) is the sum of all past experiences weighted by their probability of retrieval, for which the option with the maximum blended value is selected greedily. In particular, at the *l*-th step of an episode, the agent selects the option  $(s_l, a_l)$  with

$$a_l = \arg\max_{a \in A} V_{(s_l, a), t} \tag{5}$$

When the agent receives delayed results, the agent updates expected utilities using a credit assignment mechanism (Nguyen, McDonald, & Gonzalez, 2021). Throughout the present work, we use default parameter values for decay d = 0.5 and noise  $\sigma = 0.25$ . The mismatch penalty  $\alpha$  is set for each task individually.

#### **IBL Model of BART**

We use a previously developed IBL model for the BART (Bugbee & Gonzalez, 2022b). The instance structure in this model is as follows: the situation has the feature of the number of pumps of the balloon prior to the present decision, the decision is to pump the balloon or bank, and the utility depends on the outcome of that decision. If the balloon bursts from that decision, then the utility is 0, since the model should learn that pumping at that number of pumps led to bursting the balloon and receiving no money for that problem. If the balloon does not burst from that decision, then the utility is the value of the balloon or the number of pumps thus far plus one for the initial value, since the model should learn that pumping at that number of pumps did not burst the balloon. The model uses partial matching, in particular linear similarity, to compare the current instance to past ones, and a mismatch penalty of  $\alpha = 5$ .

#### **IBL Model of Optimal Stopping Task**

We similarly use an IBL model for the Optimal Stopping Task proposed by Bugbee and Gonzalez (2022a). The instance structure of this model is as follows: the situation has the feature of the value of the current alternative and the number of alternatives remaining in the sequence, the decision is to select the alternative or pass, and the utility is 1 if the selected alternative is the maximum and 0 otherwise. The model uses a credit assignment mechanism such that the utility is propagated back to the previous decisions in the sequence once a select action is made and the outcome is observed. The model uses partial matching, in particular linear similarity, to compare the current instance to past instances, and a mismatch penalty of  $\alpha = 10$ .

#### **Model Simulation Methods**

We use cognitive models based on IBLT (Gonzalez et al., 2003) and implemented using PyIBL, a Python implementation of IBLT (Morrison & Gonzalez, 2021). As mentioned, these models were developed and reported in Bugbee and Gonzalez (2022b) for the BART and Bugbee and Gonzalez (2022a) for the Optimal Stopping Task.

For each human participant in the data set, we simulate an IBL model agent experiencing the same stimuli, that is, the exact problems and conditions in the same order as the human. Therefore, we can map each IBL model agent to a corresponding human participant in the original study. Importantly, the models are not fit to the human data, so the correspondence between models and humans is only a result of the similarity of their experiences. As a result, we have 56 simulated IBL model agents making choices in each task,



Figure 3: (a) Distribution of mean rewards for the human participants and (b) distribution of mean rewards for IBL model agents in the BART task, compared to the Thomas et al. (2021) Model Crowd and the optimal decision process.



Figure 4: (a) Individual behavior (left panel) and crowd behavior (right panel) for the human participants and (b) individual and crowd behavior for IBL agents for a problem in the BART task in which the balloon bursts at pump 15. For the individual behavior, lines indicate the number of pumps for each individual, and red dots indicate that the participant burst the balloon. For the crowd behavior, the dashed line corresponds to half of the participants to visualize the majority decision. The human crowd banks at pump 4 and the IBL model crowd banks at pump 3.

where each simulated agent maps directly to a particular human participant.

### Wisdom of Crowds Aggregation

In alignment with Thomas et al. (2021) we use the behaviorbased majority decision to determine the WoC decision. That is, for each decision, the behavior of the crowd is that of the majority of participants.

In the BART, the behavior-based WoC crowd behavior is governed by the majority decision to pump or bank on each trial. Each individual, given that they have not already banked, decides whether to pump the balloon or bank the money. Once a participant decides to bank, presumably they have decided to bank on all following decisions, so we impute those after banking as bank decisions as well. This is a deviation from Thomas et al. (2021), where they remove participants after they make a bank decision. The crowd follows the majority until the majority either banks or the balloon bursts.

In the Optimal Stopping Task, the behavior-based WoC behavior depends on the majority decision at each alternative. For a particular alternative, each individual decides to select that alternative or pass and see the next one. The crowd will select the alternative if that is the selection of the majority; otherwise, it will pass and continue until either the majority selects a particular cat or the end of the sequence is reached and the last cat must be chosen. This is directly in alignment with Thomas et al. (2021).

#### Results

For the results, we will show the individual behavior for the human participants and IBL agents alongside their respective crowd behaviors corresponding to the majority decisions. For the BART, the crowd decision is determined for each pump or bank decision. For the Optimal Stopping Task, the crowd decision is determined at each select or pass decision.

#### WoC in the BART Task

Figures 3a and 3b show the distribution of mean rewards, the average reward, and the crowd behavior for the human participants and IBL agents respectively. The figures also display the optimal reward and the reward of the model crowd from Thomas et al. (2021) for comparison.

The distribution of mean rewards is slightly lower for the IBL model than for humans. This is explained by the need for the IBL model to learn from experience how to gain points without "reading instructions" while human participants read



Figure 5: (a) Distribution of mean rewards for human participants and (b) distribution of mean rewards for IBL model agents in the Optimal Stopping Task, compared to the Thomas et al. (2021) Model Crowd and the optimal decision process.

instructions and therefore start with more understanding of the task. The crowd behavior is comparable for the human and IBL model in both conditions. We see that the crowd performs better than the average across participants in both conditions for the human and IBL models. These results indicate that the crowd performs this task better than the average individual, and the IBL crowd behavior closely replicates the human crowd.

The "Thomas et al. Model Crowd" represented by the blue square in the figures comes from Thomas et al. (2021), and it is based on the Two-Parameter BART model (van Ravenzwaaij, Dutilh, & Wagenmakers, 2011). This model assumes that participants have a target number of pumps for each problem that they do not adapt over problems, which depends on their risk propensity and belief about the burst probability of the balloon (for more details, see Thomas et al. (2021)). Ultimately, the IBL crowd behavior shows improved performance over that of the Two-Parameter BART model in the P(Burst) = 0.2 condition, and has slightly worse performance in the P(Burst) = 0.1 condition.

The optimal performance represented by the red circle was determined by Monte Carlo simulation in Thomas et al. (2021), as the optimal number of pumps is challenging to derive. This shows 10 pumps to be optimal for P(Burst) = 0.1 yielding around \$4.00 on average, and 4 pumps to be optimal for P(Burst) = 0.2, yielding around \$1.60. Thomas et al. (2021) explained that the optimal performance appears low because the problems used in Guan et al. (2020) are fairly unrepresentative of the true environment. As many problems had late burst trials, it is possible to perform better than optimal, which we see for some human individuals and the human crowd, as well as for some IBL agents and the IBL crowd.

Figure 4 shows an example of the behavior of humans (a) and IBL models (b) in which the balloon bursts at pump 15. We observe comparable pumping behavior for humans and IBL agents. In this problem we see more IBL agents pumping more (up to pump 15) when the balloon bursts. But we also

observe that for the crowd behavior the human crowd banks at pump 4 and the IBL model crowd banks at pump 3.

### **Optimal Stopping Task**

Figures 5a and 5b show the accuracy distribution for humans and IBL model agents in the four conditions of the optimal stopping task. We observe similar distributions of accuracy between participants and IBL model agents.

We also see that crowd behavior in the IBL model is comparable to that of the human participants — in fact, the IBL model crowd performance is better in the Length 8 conditions relative to the performance of the humans. The crowd behavior is better than the average participant in all conditions for both the human and the IBL model agents. This indicates that the WoC is better than the average participant, and that the IBL WoC closely replicates the WoC of human participants.

The "Thomas et al. Model Crowd" from Thomas et al. (2021), represented by the blue square, is based on three fixed-then-linear strategies used to set thresholds for making stopping decisions. That is, participants may have fixed thresholds over positions and choose the first alternative that exceeds that threshold; they may have a starting threshold which they decrease linearly throughout the sequence; or they may have a fixed threshold for some fixed trials in the sequence, which they then decrease linearly. It is assumed that a participant uses the same strategy for all problems. The relationship between the human WoC and the Thomas et al. Model Crowd is similar to that of the IBL model WoC and the Model Crowd, again suggesting that the IBL model can replicate the human crowd behavior.

The optimal performance represented by the red circle was determined according to the findings of Gilbert and Mosteller (1966), as reported in Thomas et al. (2021). The optimal strategy is to choose the first value that is the current maximum in the sequence and is above the optimal threshold calculated based on the position in the sequence. Thomas et al. (2021) clarify that the optimal performance is surpassed since there

are a finite number of experimental problems. We see that both individuals and the various crowds sometimes have comparable or even greater accuracy than the optimal strategy.



Figure 6: Probability of stopping by position, for individuals and the crowd for the human participants and IBL agents.

Figure 6 shows the stopping probabilities in the Optimal Stopping Task at each position in the sequence for the four conditions. Each solid line corresponds to a particular individual's stopping probabilities over all problems. The dashed lines correspond to the stopping probabilities of the crowd. The similarity of the IBL agents to the human participants, as well as the IBL crowd and the human crowd, indicates that the IBL model is able to capture the stopping probability of human participants under each condition, and that the IBL model is able to replicate the human crowd closely.

### Discussion

The WoC, involving an aggregation of individual decisions, has been shown to be a powerful and effective method of producing results that are better than many people in that group (Surowiecki, 2005). However, it is unclear whether this wisdom can be beneficial in sequential decision-making tasks. Recent research suggests that simple aggregation rules (e.g., a majority decision) can result in more optimal stopping decisions in these tasks compared to the stopping decisions of most individuals in the group (Thomas et al., 2021).

This research builds on the work of Thomas et al. (2021) by demonstrating that it is possible to use cognitive models to simulate a crowd of agents and that the WoC resulting from the simulated crowd is similar to the WoC resulting from human participants. We demonstrate this idea in two sequential decision tasks. Importantly, in contrast to the descriptive statistical models in Thomas et al. (2021), we employ the learning models based on a cognitive theory of decisions from experience, IBLT (Bugbee & Gonzalez, 2022a, 2022b).

The simulation results demonstrate how these models provide predictions of human behavior and that the WoC derived from the aggregation of the simulated agents results in improved performance relative to the individual agents. Importantly, the WoC predictions of the model are similar to the WoC calculated from human data.

The cognitive models we utilize for WoC are learning models, and this addresses a primary limitation described by Thomas et al. (2021), in that their statistical models could not dynamically adapt as human decision makers. Although Thomas et al. (2021) show that models that fit human data can generalize to problems in the same class of tasks, our work demonstrates that a cognitive model that accounts for learning without relying on specific human data can be used across distinct tasks of varying structure, while providing comparable individual-level predictions and WoC decisions.

Learning is likely to occur in human participants to some extent, and there is value in being able to capture behavioral changes as their experience grows. Our results demonstrate that the cognitive models we propose can learn to perform at the same level as human participants and that the WoC derived from crowds of IBL agents are similar to the WoC derived from human crowds. In future work, these models could be applied to settings in which human adaptation is a prominent feature of the task.

### Acknowledgments

An Open Science Framework project is available at https://osf.io/275gp/ with the data, code, and analysis files. This research was supported by AFRL Award FA8650-20-F-6212, sub-award number 1990692 to Cleotilde Gonzalez. The authors thank John Anderson, Dan Bothell, Christian Lebiere and Leslie Blaha for their helpful feedback regarding this work. We also thank Michael Lee for assistance with the data.

#### References

- Anderson, J. R., & Lebiere, C. J. (2014). *The atomic components of thought*. Psychology Press.
- Bugbee, E. H., & Gonzalez, C. (2022a). Deciding when to stop: Cognitive models of sequential decisions in optimal stopping tasks. *In preparation*.
- Bugbee, E. H., & Gonzalez, C. (2022b). Making predictions without data: How an instance-based learning model predicts sequential decisions in the balloon analog risk task. In *Proceedings of the annual meeting of the cognitive science* society.
- Gilbert, J. P., & Mosteller, F. (1966). Recognizing the maximum of a sequence. *Journal of the American Statistical Association*, 61(313), 35–73. Retrieved from http://www.jstor.org/stable/2283044
- Gonzalez, C., & Dutt, V. (2011). Instance-based learning: Integrating decisions from experience in sampling and repeated choice paradigms. *Psychological Review*, 118(4), 523–51.

- Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instancebased learning in dynamic decision making. *Cognitive Science*, 27. doi: 10.1016/S0364-0213(03)00031-4
- Guan, M. (2019). A cognitive modeling analysis of risk in sequential choice tasks. Retrieved from https://escholarship.org/uc/item/802684nb
- Guan, M., Stokes, R., Vandekerckhove, J., & Lee, M. D. (2020). A cognitive modeling analysis of risk in sequential choice tasks. *Judgment and Decision Making*, *15*, 823-850.
- Lee, M. D. (2006). A hierarchical bayesian model of human decision-making on an optimal stopping problem. *Cognitive science*, *30*(3), 1–26.
- Lejuez, C. W., Read, J. P., Kahler, C. W., Richards, J. B., Ramsey, S. E., Stuart, G. L., ... Brown, R. A. (2002). Evaluation of a behavioral measure of risk taking: the balloon analogue risk task (bart). *Journal of Experimental Psychology: Applied*, 8(2), 75.
- Morrison, D., & Gonzalez, C. (2021). *Pyibl version 4.0.* Retrieved 2021-03-30, from http://pyibl.ddmlab.com/
- Nguyen, T. N., McDonald, C., & Gonzalez, C. (2021). Credit assignment: Challenges and opportunities in developing human-like ai agents (Tech. Rep.). Carnegie Mellon University.
- Surowiecki, J. (2005). The wisdom of crowds. Anchor.
- Thomas, B., Coon, J., Westfall, H., & Lee, M. (2021, 07). Model-based wisdom of the crowd for sequential decision-making tasks. *Cognitive Science*, *45*, e13011. doi: 10.1111/cogs.13011
- van Ravenzwaaij, D., Dutilh, G., & Wagenmakers, E.-J. (2011). Cognitive model decomposition of the bart: Assessment and application. *Journal* of Mathematical Psychology, 55(1), 94-105. (Special Issue on Hierarchical Bayesian Models) doi: https://doi.org/10.1016/j.jmp.2010.08.010

# **Clarifying System 1 & 2 through the Common Model of Cognition**

Brendan Conway-Smith (brendan.conwaysmith@carleton.ca), Robert L. West (robert.west@carleton.ca) Department of Cognitive Science, Carleton University

Ottawa, ON K1S5B6 Canada

#### Abstract

There have been increasing challenges to dual-system descriptions of System-1 and System-2, critiquing them as imprecise and fostering misconceptions. We address these issues here by way of Dennett's appeal to use computational thinking as an analytical tool, specifically we employ the Common Model of Cognition. Results show that the characteristics thought to be distinctive of System-1 and System-2 instead form a spectrum of cognitive properties. By grounding System-1 and System-2 in the Common Model we aim to clarify their underlying mechanisms, persisting misconceptions, and implications for metacognition.

**Keywords:** dual-system; dual-process; system-1; system-2; common model; metacognition; computational architecture

#### Introduction

This paper re-visits Dennett's (1981) notion that philosophical discussion can benefit from the use of computational modelling. We do this by showing how recent criticisms of the dual-systems view of the mind (System-1 and System-2), can be clarified using the Common Model of Cognition to ground the discussion (Laird, Lebiere & Rosenbloom, 2017).

The terms System-1 and System-2 refer to a dualsystem model that ascribes distinct characteristics to what are thought to be opposing aspects of cognition (Wason & Evans, 1974; Stanovich, 1999; Strack & Deutsch, 2004; Kahneman, 2003, 2011). System-1 is considered to be evolutionarily old and characterized as fast, associative, emotional, automatic, and not requiring working memory. System-2 is more evolutionarily recent and thought to be slow, declarative, rational, effortful, and relying on working memory. Kahneman (2003) referred to System-1 as "intuitive" and System-2 as "rational", thus linking them to higher level folk psychology concepts. The neural correlates of System-1 and System-2 have also been studied (e.g., Tsujii & Watanabe, 2009). System-1 and System-2 are often used in fields such as psychology, philosophy, neuroscience, and artificial intelligence as a means for ontologizing the functional properties of human cognition.

Recently, however, this dual-system model has been criticized for lacking precision and conceptual clarity (Keren & Schul, 2009), leading to significant misconceptions (Pennycook et al., 2018; Houwer, 2019) and obscuring the dynamic complexities of psychological processes (Moors, 2016). One of the originators of dual-system theory stated that an important issue for future research is the problem that "current theories are framed in general terms and are yet to be developed in terms of their specific computational architecture" (Evans, 2003).

Following Dennett (1981) we argue that a computational description is essential for clarifying high level, psychological characterizations such as System-1 and System-2. At the time, Dennett received significant pushback on his view. However, we argue that it was too early in the development of computational models to fully appreciate the pragmatic value of his position.

In the spirit of this endeavour, Proust (2013) has argued that a more precise computational definition is needed to understand the role of System-1 and System-2 in metacognition. Proust defined these systems in terms of informational typologies (System-1 non-conceptual; System-2 conceptual). Similarly, Thomson et al. (2015) argued that the expert use of heuristics (System-1) could be defined in terms of instance based learning in ACT-R. In fact, there are numerous ways that cognitive models and cognitive architectures can and have been mapped onto the System-1 and 2 distinction. For example, dual-process approaches to learning have been instantiated within the CLARION architecture, modelling the interaction between implicit and explicit processes (Sun, Terry & Slusarz, 2005). System-1 and 2 have also been instantiated directly into the LIDA architecture (Faghihi et al., 2014).

While it is useful to work on modelling different aspects of System-1 and 2, the larger question is, in what sense is System-1 and 2 a valid construct? What are the necessary and sufficient conditions that precisely define System-1 and 2? And what are the cognitive and neural alignments to System-1 and System-2 (Evans, 2003)?

### **The Common Model**

The Common Model of Cognition, originally the 'Standard Model' (Laird et al., 2017) is a consensus architecture that integrates decades of research on how human cognition functions computationally. The Common Model represents a convergence across

cognitive architectures regarding the modules and components necessary for biological and artificial intelligence. These modules are correlated with their associated brain regions and verified through neuroscience (Steine-Hanson et al., 2018). Neural evidence strongly supports the Common Model as a leading candidate for modeling the functional organization of the human brain (Stocco et al., 2021).

The computational processes of the Common Model are categorized into five components — working memory, perception, action, declarative memory, and procedural memory. Procedural memory is described as a production system which contains units called production rules (or 'productions'). The production system interacts with different modules through working memory represented as buffers. While these components are implemented differently among Common Model-type architectures, they describe a common functionality across implementations.

### System-1

Researchers generally describe System-1 by using a constellation of characteristics. Specifically, System-1 is described as fast, associative, emotional, automatic, and not requiring working memory (Kahneman, 2011; Evans, 2003; Strack & Deutsch, 2004). System-1 is considered to be evolutionary old and present within animals. It is composed of biologically programmed instinctive behaviours and operations that contain innate modules of the kind put forth by Fodor (1983). System-1 is not comprised of a single system but is an assembly of sub-systems that are largely autonomous (Stanovich & West, 2000). Automatic operations are usually described as involving minimal or no effort, and without a sense of voluntary control (Kahneman, 2011). Researchers generally agree that System-1 is made of parallel and autonomous subsystems that output only their final product into consciousness (often as affect), which then influences human decision-making (Evans, 2003). This is one reason the system has been called "intuitive" (Kahneman, 2003).

System-1 relies on automatic processes and shortcut strategies called heuristics — problem solving operations or rule of thumb strategies (Simon, 1955). The nature of System-1 is often portrayed as non symbolic, and has been associated with reinforcement learning (Barto et al., 1981) and neural networks (McLeod, 1998). Affect is integral to System-1 processes (Mitchell, 2011). Affect based heuristics result from an individual evaluating a stimulus based on their likes and dislikes. In more complex decisionmaking, it occurs when a choice is either weighed as a net positive (with more benefits than costs), or as net negative (less benefits than costs) (Slovic et al., 2004).

System-1 can produce what are called "cognitive illusions" that can be harmful if left unchecked. For example, the 'illusion of validity' is a cognitive bias in which individuals overestimate their ability to accurately predict a data set, particularly when it shows a consistent pattern (Kahneman & Tversky, 1973). Biases and errors of System-1 operate automatically and cannot be turned off at will. However, they can be offset by using System-2 to monitor System-1 and correct it.

## System-1 in the Common Model

System-1 can be associated with the production system which is the computational instantiation of procedural memory in the Common Model (Singley & Anderson, 1989). Procedural knowledge is represented as production rules ("productions") which are modeled after computer program instructions in the form of condition-action pairings. They specify a condition that, when met, will perform a prescribed action. A production can also be thought of as an *if-then* rule (Anderson, 1993). If it matches a condition, then it fires an action. Production rules transform information to resolve problems or complete a task, and are responsible for state-changes within the system. Production rules fire automatically off of conditions in working memory buffers. Their automaticity is due to the fact that they are triggered without secondary evaluation. Neurologically, production rules correlate with the 50ms decision timing in the basal ganglia (Stocco, Lebiere, & Anderson, 2010). The production system can enact reinforcement learning in the form of utility learning, where faster or more useful productions are rewarded and are more likely to be used later (Anderson, 1993). In a similar way, problem solving heuristics can be implemented as production rules (Payne et al., 1988).

The Common Model production system has many of the properties associated with System-1 such as being fast, automatic, implicit, able to implement heuristics, and reinforcement learning. However, the Common Model declarative memory system also has some of the properties associated with System-1. Specifically, associative learning and the ability to implement heuristics that leverage associative learning (Thomson et al., 2015). Here, it is important to understand that the Common Model declarative memory cannot operate without the appropriate productions firing, and without the use of buffers (working memory). Therefore, from a Common Model perspective, System-1 minimally involves productions firing based on buffer conditions, but can also involve productions directing declarative memory retrieval, which also relies on buffers. Based on this, System-1 cannot be defined as being uniquely aligned with either declarative or procedural memory. System-1 activity must involve production rules and buffers, and can also involve declarative knowledge.

## System-2

Researchers generally view System-2 as a collection of cognitive properties, characterized as slow,

propositional, rational, effortful, and requiring working memory (Kahneman, 2011; Strack & Deutsch, 2004; Frankish 2010). System-2 involves explicit propositional knowledge that is used to guide decisionmaking (Epstein & Pacini, 1999). Propositional knowledge is associated with relational knowledge (Halford, Wilson, & Phillips, 2010) which represents entities (e.g.: John and Mary), the relation between them (e.g.: loves) and the role of those entities in that relation (e.g.: John loves Mary). Higher level rationality in System-2 is also said to be epistemically committed to logical standards (Tsujii & Watanabe, 2009). System-2 processes are associated with the subjective experiences of agency, choice, and effortful concentration (Frankish, 2010). The term "effortful" encompasses the intentional, conscious, and more strenuous use of knowledge in complex thinking. Higher level rationality is considered responsible for human-like reasoning, allowing for hypothetical thinking, long-range planning, and is correlated with overall measures of general intelligence (Evans, 2003).

Researchers have studied various ways in which System-2's effortful processes can intervene in System-1 automatic operations (Kahneman, 2003). Ordinarily, an individual does not need to invoke System-2 unless they notice that System-1 automaticity is insufficient or risky. System-2 can intervene when the anticipated System-1 output would infringe on explicit rules or potentially cause harm. For example, a scientist early in their experiment may notice that they are experiencing a feeling of certainty. System-2 can instruct them to resist jumping to conclusions and to gather more data. In this sense, System-2 can monitor System-1 and override it by applying conceptual rules.

## System-2 in the Common Model

Laird (2020) draws on Newell (1990), Legg and Hutter (2007) and others to equate rationality with intelligence, where "an agent uses its available knowledge to select the best action(s) to achieve its goal(s)." Newell's Rationality Principle involves the assumption that problem-solving occurs in a problem space, where knowledge is used to navigate toward a desired end. As Newell puts it, "an agent will use the knowledge it has of its environment to achieve its goals" (1982, p. 17). The prioritizing of knowledge in decision-making corresponds with the principles of classical computation involving symbol transformation and manipulation.

The Common Model architecture fundamentally distinguishes between declarative memory and procedural memory. This maps roughly onto the distinction between explicit and implicit knowledge — where declarative knowledge can be made explicitly accessible in working memory, procedural knowledge operates outside of working memory and is inaccessible. However, declarative knowledge can also function in an implicit way. The presence of something within working memory does not necessarily mean it

will be consciously accessed (Wallach & Lebiere, 2003).

Higher level reasoning involves the retrieval of 'chunks', representing propositional information, into buffers (working memory) to assist in calculations and problem-solving operations. This appears to correlate with what System-2 researchers describe as "effortful", as this requires more computational resources (i.e., more productions) to manage the flow of information through limited space in working memory (buffers). As Kahneman points out, System-1 can involve knowledge of simple processes such as 2+2=4. However, more complex operations such as 17x16 require calculations that are effortful, a characteristic that is considered distinctive of System-2 (Kahneman, 2011).

Effort, within the Common Model, involves greater computational resources being allocated toward a task. Moreover, the retrieval and processing of declarative knowledge requires more steps and more processing time when compared to the firing of productions alone. This longer retrieval and processing time can also account for the characteristic of "slow" associated with System-2.

## **Emotion in System-1 and 2**

Emotion and affect plays a vital role in the distinction between System-1 and System-2 processes (Chaiken & Trope, 1999; Kahneman, 2011). Decisions in System-1 are largely motivated by an individual's implicit association of a stimulus with an emotion or affect (feelings that something is bad or good). Behavior motivated by emotion or affect is faster, more automatic, and less cognitively expensive. One evolutionary advantage of these processes is that they allow for split-second reactions that can be crucial for avoiding predators, catching food, and interacting with complex and uncertain environments.

Emotions can bias or overwhelm purely rational decision processes, but they can also be overridden by System-2 formal rules. While emotions and affect have historically been cast as the antithesis of reason, their importance in decision-making is being increasingly investigated by researchers who give affect a primary role in motivating decisions (e.g., Zajonc, 1980; Barrett & Salovey, 2002). Some maintain that rationality itself is not possible without emotion, as any instrumentally rational system must necessarily pursues desires (Evans, 2012).

## **Emotion in the Common Model**

Feelings and emotions have strong effects on human performance and decision-making. However, there is considerable disagreement over what feelings and emotions are and how they can be incorporated into cognitive models. However, while philosophical explanations of affect have been debated, functional accounts of emotions and feelings within cognitive models have been built. Emotions have been modeled as amygdala states (West & Young, 2017), and somatic markers as emotional tags attached to units of information (Domasio, 1994). In Sigma models, lowlevel appraisals have been modeled as architectural selfreflections on factors such as expectedness, familiarity, and desirability (Rosenbloom, et al., 2015). Core affect theory has been modeled in ACT-R to demonstrate how an agent may prioritize information using emotional valuation (Juvina, Larue & Hough, 2018). Also, feelings have also been modelled by treating them as non propositional representations in buffers or "metadata" (West & Conway-Smith, 2019).

Overall, the question of how to model emotion in the Common Model remains unresolved. However, as indicated in the research above, emotion has multiple routes for interacting with cognition in the Common Model.

## Effort in System-1 and 2

The concept of "effort" makes up a significant and confusing dimension of System-1 and System-2. While it is mainly associated with System-2 rationality, a precise definition of "effort" remains elusive and is largely implicit in discussions of System-1 and 2. Because System-2 is considered to have a low processing capacity, its operations are associated with greater effort and a de-prioritizing of irrelevant stimuli (Stanovich, 1999).

Effort can be associated with complex calculations in System-2 to the extent that it taxes working memory. Alternatively, effort can be associated with System-2's capacity to overrule or suppress automatic processes in System-1 (Kahneman, 2011). For example, various System-1 biases (such as the "belief bias") can be subdued by instructing people to make a significant effort to reason deductively (Evans, 1983). The application of formal rules to "control" cognitive processes is also called metacognition — the monitoring and control of cognition (Flavell, 1979; Fletcher & Carruthers, 2012). Researchers have interpreted metacognition through a System-1 and System-2 framework (Arango-Muñoz, 2011; Shea et al., 2014). System-1 metacognition is thought to be implicit, automatic, affect-driven, and not requiring working memory. System-2 metacognition is considered explicit, rule-based, and relying on working memory.

While the concept of "effort" is considered to be the monopoly of System-2, a computational approach suggests that effort is a continuum — with low effort cognitive phenomena being associated with System-1, and high effort cognitive phenomena being associated with System-2.

## Effort in the Common Model

The Common Model helps to elucidate how "effort" can be present in System-1 type operations in the

absence of other System-2 characteristics. While neither dual-system theories nor the Common Model contain a clear definition of "effort", computational characteristics associated with effort can be necessary to System-1. For instance, "effort" is often associated with the intense use of working memory. However, the Common Model requires working memory (along with its processing limitations) for both System-1 and System-2 type operations. There is no reason why System-1 should necessarily use less working memory than System-2 in the Common Model. Instead, it would depend on the task duration and intensity.

System-1 and System-2 metacognition can also be clarified by importing Proust's (2013) more precise account. Proust attempted to elucidate these two systems by claiming that they should be distinguished by their distinctive informational formats (System-1 non-conceptual; System-2 conceptual). In this sense, System-1 metacognition can exert effortful control while simultaneously being implicit and non conceptual. For example, consider a graduate student attending a conference while struggling not to fall asleep. An example of System-1 metacognition would involve the context implicitly prompting them to feel nervous, noticing their own fatigue, and then attempting to stay awake. This effort is context-driven, implicit, non conceptual, and effortful. Alternatively, System-2 metacognition can exert effort by way of explicit concepts, as in the case of a tired conference-attendee repeating the verbal instruction "try to focus". Either of these scenarios could be modelled using the Common Model, and to reiterate, there is little reason why System-1 should require less effort.

Another way to think about effort is in terms of the expense of neural energy. In this sense, effort can be viewed as the result of greater caloric expenditure in neurons. The neural and computational dynamics responsible for the effortful control of internal states have shown to be sensitive to performance incentives (Egger et al., 2019). Research also indicates that the allocation of effort as cognitive control is dependent on whether a goal's reward outweighs its costs (Shenhav, et al., 2017). Both of these relate to reinforcement learning, which is associated with System-1.

Examining this question through the Common Model suggests that "effort" is not traditionally well defined, nor is it the sole privy of System-2. Rather, effort can be involved in processes characteristic of both System-1 and System-2.

## Conclusion

The Common Model sheds light on the specific mechanisms that give rise to the general traits associated with System-1 and System-2. Interpreting System-1 and System-2 within the Common Model results in our concluding that the "alignment assumption" (that the two systems are opposites) is a false dichotomy. There are, of course, cases where all

properties of System-1 and System-2 are cleanly bifurcated on either side. However, between these two extremities lies a spectrum where the characteristics are mixed. Few, if any, of these properties are 'necessary and sufficient' to be sharply distinctive of either.

Evidence for this is as follows:

1. *System-2 is grounded in System-1*. While System-1 depends on procedural memory, so too does System-2. System-2 cannot operate separately due to the architectural constraints of the Common Model. Even if a System-2 process were primarily driven by declarative knowledge, it would still require System-1 procedural knowledge to be retrieved and acted upon.

2. System-1 and System-2 characteristics are often mixed as they routinely act together. System-2 goal-directed rationality often requires affect in the from of a desired end. Also, System-2 rationality is subject to System-1 affective biases.

3. Both System-1 and System-2 require working memory. While conventional views claim that System-1 does not require working memory, the constraints of the Common Model necessitate it. Production rules (procedural knowledge) are activated by the content of buffers (working memory) and hence are required by both systems.

4. Effort can be directed toward both System-2 rationality and System-1 metacognitive control. The effortful allocation of cognitive resources in System-1 can be based on an implicit cost-benefit analysis.

Regardless of whether one adopts the Common Model architecture, researchers should be cautious of assuming the System-1 and System-2 dichotomy within their work. The framework is far from settled and deep issues continue to be unresolved. Questions remain as to whether System-1 and System-2 constitute an ontology or a convenient epistemology.

Since before Descartes, substance dualism has continually been reimagined as mind and soul, reason and emotions, and opposing modes of thought. These have been expressions of the human species' attempt to make sense of our own minds, its processes, and how this understanding maps onto our personal experience. Clearly, System-1 and System-2 captures something deeply intuitive about the phenomenology of cognition. However, as we have discussed Kahneman's System-1 biases it may be worth asking — is System-2 a System-1 illusion? That is, do we assume the existence of System-2 simply because we so often act as if it exists?

By situating System-1 and System-2 within the Common Model of Cognition, we have attempted to bring light to this subject by clarifying its underlying mechanisms, misconceptions, and the base components needed for future research.

## References

- Anderson, J. R. (1993). Knowledge representation. Rules of the mind.
- Anderson, J. R., & Lebiere, C. (2003). The Newell test for a theory of cognition. Behavioral and brain Sciences, 26, 587.
- Arango-Muñoz, S. (2011). Two levels of metacognition. Philosophia, 39(1), 71-82.
- Barrett, L. F., & Salovey, P. (Eds.). (2002). The wisdom in feeling: Psychological processes in emotional intelligence. Guilford Press.
- Barto, A. G., Sutton, R. S., & Brouwer, P. S. (1981). Associative search network: A reinforcement learning associative memory. Biological cybernetics.
- Chaiken, S., & Trope, Y. (Eds.). (1999). Dual-process theories in social psychology. Guilford Press.
- Damasio, A. R. (1994). Descartes' error and the future of human life. Scientific American, 271(4), 144-144.
- Dennett, D. C. (2017). Brainstorms: Philosophical essays on mind and psychology. MIT press.
- Epstein, S., & Pacini, R. (1999). Some basic issues regarding dual-process theories from the perspective of cognitive–experiential self-theory.
- Evans, J.St.B.T. (1983) On the conflict between logic and belief in syllogistic reasoning. Mem. Cogn. 11.
- Evans, J. S. B. (2003). In two minds: dual-process accounts of reasoning. Trends in cognitive sciences.
- Evans, J. S. B. (2012). Spot the difference: distinguishing between two kinds of processing. Mind & Society, 11(1), 121-131.
- Faghihi, U., Estey, C., McCall, R., & Franklin, S. (2015). A cognitive model fleshes out Kahneman's fast and slow systems. Biologically Inspired Cognitive Architectures, 11, 38-52.
- Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. American psychologist, 34(10), 906.
- Fletcher, L., & Carruthers, P. (2012). Metacognition and reasoning. Philosophical Transactions of the Royal Society B: Biological Sciences, 367(1594).
- Fodor, J. (1983). The Modularity of Mind, Crowell, Scranton.
- Frankish, K. (2010). Dual-Process and Dual-System Theories of Reasoning: Dual-Process and Dual-System Theories of Reasoning. Philosophy Compass.
- Juvina, I., Larue, O., & Hough, A. (2018). Modeling valuation and core affect in a cognitive architecture: The impact of valence and arousal on memory and decision-making. Cognitive Systems Research, 48.
- Halford, G. S., Wilson, W. H., & Phillips, S. (2010). Relational knowledge: The foundation of higher cognition. Trends in cognitive sciences.
- Houwer, J. (2019). Moving beyond System 1 and System 2: Conditioning, implicit evaluation, and habitual responding might be mediated by relational knowledge. Experimental Psychology, 66(4), 257.
- Kahneman, Daniel; Tversky, Amos (1973). "On the Psychology of Prediction". Psychological Review.

- Kahneman, D. (2003). A perspective on judgment and choice: Mapping bounded rationality. American Psychologist, 58(9), 697–720.
- Kahneman, D. (2011). Thinking, fast and slow. Macmillan. Anderson, J. R., & Lebiere, C. (2003). The Newell test for a theory of cognition. Behavioral and brain Sciences, 26, 587.
- Keren, G., & Schul, Y. (2009). Two is not always better than one: A critical evaluation of two-system theories. Perspectives on psychological science, 4(6), 533-550.
- Laird, J. (2020). Intelligence, knowledge & human-like intelligence. Journal of Artificial General Intelligence.
- Laird, J., Lebiere, C., & Rosenbloom, P. (2017). A standard model of the mind. Ai Magazine.
- Legg, S., & Hutter, M. (2007). Universal intelligence: A definition of machine intelligence. Minds and machines, 17(4), 391-444.
- McLeod, P., Plunkett, K., & Rolls, E. T. (1998). Introduction to connectionist modelling of cognitive processes. Oxford University Press.
- Mitchell, D. G. (2011). The nexus between decision making and emotion regulation: a review of convergent neurocognitive substrates. Behavioural brain research, 217(1), 215-231.
- Moors, A. (2016). Automaticity: Componential, causal, and mechanistic explanations. Annual review of psychology, 67, 263-287.
- Panksepp, J., & Biven, L. (2012). The archaeology of mind: neuroevolutionary origins of human emotions. WW Norton & Company.
- Proust, J. (2013). The philosophy of metacognition: Mental agency and self-awareness. Oxford.
- Rosenbloom, P. S., Gratch, J., & Ustun, V. (2015, July). Towards emotion in sigma: from appraisal to attention. In International conference on artificial general intelligence (pp. 142-151). Springer, Cham.
- Payne, J. W., Bettman, J. R., & Johnson, E. J. (1992). Behavioral decision research: A constructive processing perspective. Annual review of psychology.
- Pennycook, G., De Neys, W., Evans, J. S. B., Stanovich, K. E., & Thompson, V. A. (2018). The mythical dualprocess typology. Trends in Cognitive Sciences.
- Shea, N., Boldt, A., Bang, D., Yeung, N., Heyes, C., & Frith, C. D. (2014). Supra-personal cognitive control and metacognition. Trends in cognitive sciences.
- Shenhav, A. et al. (2017) Toward a rational and mechanistic account of mental effort. Annu. Rev. Neurosci. 40, 99–124
- Simon, H. A. (1955). A behavioural model of rational choice. The Quarterly Journal of Economics, 69.
- Slovic, P., Finucane, M. L., Peters, E., & MacGregor, D. G. (2004). Risk as analysis and risk as feelings: Some thoughts about affect, reason, risk, and rationality. Risk Analysis: An International Journal.
- Stanovich, K. (1999). Who is rational? Studies of individual differences in reasoning. Psychology Press.

- Stanovich, K. E., & West, R. F. (2000). Individual differences in reasoning: Implications for the rationality debate?. Behavioral and brain sciences.
- Steine-Hanson, Z., Koh, N., & Stocco, A. (2018). Refining the Common Model of Cognition through large neuroscience data. Procedia computer science.
- Strack, F., & Deutsch, R. (2004). Reflective and impulsive determinants of social behavior. Personality and social psychology review, 8(3), 220-247.
- 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.
- Stocco, A., Sibert, C., Steine-Hanson, Z., Koh, N., Laird, J. E., Lebiere, C. J., & 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, 235, 118035.
- Sun, R., Slusarz, P., & Terry, C. (2005). The interaction of the explicit and the implicit in skill learning: a dual-process approach. Psychological review.
- Tay, S. W., Ryan, P., & Ryan, C. A. (2016). Systems 1 and 2 thinking processes and cognitive reflection testing in medical students. Canadian medical education journal, 7(2), e97.
- Thomson, R., Pyke, A., Hiatt, L. M., & Trafton, J. G. (2015, July). An Account of Associative Learning in Memory Recall. In CogSci.
- Tsujii, T.; Watanabe, S. (2009). "Neural correlates of dual-task effect on belief-bias syllogistic reasoning: a near-infrared spectroscopy study". Brain Research.
- Wallach, D., & Lebiere, C. (2003). Implicit and explicit learning in a unified architecture of cognition. Advances in Consciousness Research, 48, 215-252.
- Wason, P. C., & Evans, J. S. B. (1974). Dual processes in reasoning?. Cognition, 3(2), 141-154.
- 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.
- West, R. L., & Young, J. T. (2017, October). Proposal to add emotion to the standard model. In 2017 AAAI Fall Symposium Series.
- Zajonc, R. B. (1980). Feeling and thinking: Preferences need no inferences. American psychologist.

# **Combining Machine Learning and Cognitive Models for Adaptive Phishing Training**

### Edward A. Cranford (cranford@cmu.edu)

Department of Psychology, Carnegie Mellon University 5000 Forbes Avenue, Pittsburgh, PA 15221 USA

### Shahin Jabbari (shahin@drexel.edu)

Department of Computer Science, Drexel University 3675 Market Street, Office 1151, Philadelphia, PA 19104 USA

### Han-Ching Ou (hou@g.harvard.edu)

Department of Computer Science, Harvard University 29 Oxford Street, Cambridge, MA 02138 USA

### Milind Tambe (milind\_tambe@harvard.edu)

Center for Research in Computation and Society, Harvard University 150 Western Avenue, Room 2.107, Allston, MA 02134 USA

#### Cleotilde Gonzalez (coty@cmu.edu)

Department of Social and Decision Sciences, Carnegie Mellon University 5000 Forbes Avenue, Pittsburgh, PA 15221 USA

### Christian Lebiere (cl@cmu.edu)

Department of Psychology, Carnegie Mellon University 5000 Forbes Avenue, Pittsburgh, PA 15221 USA

#### Abstract

Organizations typically use simulation campaigns to train employees to detect phishing emails but are non-personalized and fail to account for human experiential learning and adaptivity. We propose a method to improve the effectiveness of training by combining cognitive modeling with machine learning methods. We frame the problem as one of scheduling and use the restless multi-armed bandit (RMAB) framework to select which users to target for intervention at each trial, while using a cognitive model of phishing susceptibility to inform the parameters of the RMAB. We compare the effectiveness of the RMAB solution to two purely cognitive approaches in a series of simulation studies using the cognitive model as simulated participants. Both approaches show improvement compared to random selection and we highlight the pros and cons of each approach. We discuss the implications of these findings and future research that aims to combine the benefits of both methods for a more effective solution.

**Keywords:** cognitive models; model-tracing; restless multiarmed bandit; Instance-Based Learning; ACT-R; phishing

#### Introduction

Phishing remains one of the biggest threats to cybersecurity in an organization (APWG Phishing Report, 2021). Typical training of employees involves limited cybersecurity awareness tutorials and simulation campaigns (Yeoh et al., 2021). During simulation campaigns, phishing emails are sent to employees, usually selected at random, and if a user clicks on a link embedded in the email, then they are given immediate feedback and training about how to detect phishing emails. While the method is effective compared to no intervention, it may be ineffective if it targets more phishaware users than naïve users who are more susceptible to phishing. We believe that simulation campaigns could be improved through personalization by strategically selecting who to target. However, to determine who to target for training, one needs a representation of the cognitive states of each individual in the organization (i.e., their propensity to fall victim to a phishing attack).

Recent advances in simulation campaigns attempt to personalize training to determine which users to select based on risk propensity (e.g., Cyber Guru, 2019), but these approaches do not account for human experiential learning and adaptivity through repeated interactions with the environment. Recent research in end-user susceptibility to phishing emails (Cranford et al., 2021) implies that phishing classification decisions can be framed as decisions from experience in accordance with Instance-Based Learning Theory (IBLT; Gonzalez, Lerch, & Lebiere, 2003). In line with IBLT, phishing decisions are made by retrieving classifications from memory and generalizing across past experiences, or instances, that are similar to the current email. Decisions are thus influenced by memory effects such as recency, frequency, and similarity of past emails to the features of the current email, and contribute to learning and adaptivity (e.g., Hakim et al., 2020; Singh et al. 2019; 2020).

The present research is a first step toward developing a training methodology that uses cognitive principles to determine what users to select to receive training at each time step. The problem can be framed as a scheduling problem that aims to optimize the targeting of users in order to maximize the overall probability of adopting safe email behavior, without bombarding users with interventions. Our solution combines cognitive models of end-user susceptibility to phishing emails and machine learning methods to identify users most in need of training. We use the Restless Multi-Armed Bandit (RMAB) framework that models each user (arm) as a Markov Decision Process (MDP), using the cognitive model to define the MDP. RMABs have been used successfully in healthcare settings to strategically assign intervention to patients most in need (Biswas et al., 2021), and anti-phishing training presents an analogous situation.

We also present a purely cognitive approach that incorporates model-tracing techniques to trace user behavior and identify which users to select at each time step. The RMAB solution is compared to the cognitive solution in a set of simulation studies using cognitive models as simulated participants. The results show that both approaches are equally more effective than random selection but differ in selection preferences. We highlight the pros and cons of each approach and discuss plans for future research that aims to combine the strengths of the MAB and cognitive approaches.

### Modeling a Phishing Training Task

The task was designed to replicate a real-world phishing training scenario that could still be implemented in a human laboratory experiment. Users are run simultaneously in batches and are presented either a phishing email or a ham email on each trial as determined by the selection algorithm. Ham emails are non-spam, non-phish, "good" emails, intended for the specific recipient with a legitimate purpose. After each trial, users are provided feedback only after incorrectly classifying a phishing email, which represents immediate phishing awareness training from an organization, while users do not typically receive feedback otherwise.

While human subjects' experiments are greatly limited by the number of users that can be run simultaneously in a laboratory setting (e.g., 10 is a practical number), simulations are less restrictive. Therefore, in all reported analyses, we simulated 1000 users (near maximum possible for parallel simulations with 16GB RAM) for 100 trials of training (near maximum trials possible in a 1-hour laboratory experiment).

#### **Defining Users**

Among the vast individual differences and factors that influence phishing susceptibility, including demographics such as age, sex, and education (e.g., Sheng et al., 2010), and personality and social factors such as the Big 5 or the Dark Triad (Curtis et al., 2018; Yang et al., 2022), one of the most important factors is amount of email usage and knowledge and experience with phishing emails and network security (Lin et al., 2019; Sheng et al., 2010; Yang et al., 2022). In fact, these factors of overall *email usage* and *phishing and network security experience* align well with our own theory that defines user susceptibility to phishing as arising from decisions from experience as outlined by IBLT. Therefore, we designed a set of users that we could simulate in our IBL model based on individual differences in initialized instances. Each user in the model is initialized with a random number of emails (10-100 in increments of 10, uniformly distributed; Initialized Length), which represents individual differences in the amount of *email usage*, of which a random proportion are phishing emails (0.7-1.0, normally distributed within limits and rounded to the nearest 0.05, M = 0.85, sd = 0.05; Ham Proportion), which represents individual differences in the amount of *phishing and network security experience*. We used the same set of users in all simulations reported below.

#### **Cognitive Model Description**

Cranford et al. (2021) developed a generalizable IBL model of phishing susceptibility as arising from decisions from experience. The model accurately predicted classification decisions in two different tasks with different databases of phishing and ham emails: the Phishing Training Task (PTT; Singh et al., 2019) and the Phishing Email Susceptibility Test (PEST; Hakim et al., 2020). This model was used in the simulations reported below to generate predictions of human decision making against each selection algorithm and served as a basis for designing the Cognitive Selection algorithms.

The cognitive model was developed in ACT-R (Anderson & Lebiere, 1998) and makes classification decisions in accordance with the IBL process. On each trial, the model generates a classification decision by retrieving similar past instances based on the context features of the email. The features of the emails include the sender, subject, body, link text, and url. Decisions are thus based on the semantic similarity between email features. The semantic similarity values between features of two emails are computed using the University of Maryland Baltimore County's semantic textual-similarity tool (Han et al., 2013), which uses a combination of latent semantic analysis (LSA) and WordNet. Retrieval of past instances is based on ACT-R's blending mechanism (Lebiere, 1999; Gonzalez et al., 2003) which returns a consensus value (in this case, a classification of ham or phish) across all memories, rather than from a specific memory:

$$V = \underset{V_o}{\operatorname{argmin}} \sum_{i} P_i \times \left(1 - Sim(V_o, V_i)\right)^2 \tag{1}$$

The value V is the one that minimizes the dissimilarity between the possible decisions and the actual decision in chunk *i*, weighted by the probability of retrieval  $P_i$  of the matching chunk *i* in memory.

$$P_i = \frac{e^{A_{i/t}}}{\sum_i e^{A_{j/t}}}$$
(2)

 $P_i$  reflects the ratio of an instance's activation  $A_i$  and temperature *t*, which defaults to  $\sqrt{2} * s$ , where *s* equals the variance parameter of noise. The activation  $A_i$  of an instance *i*, is determined by:

$$A_i = \ln \sum_{j=1}^n t_j^{-d} + MP * \sum_k Sim(v_k, c_k) + \varepsilon_i$$
(3)

where the first term reflects the power law of practice and forgetting, where  $t_j$  is the time since the *j*th occurrence of chunk *i* and *d* is the decay rate (set to 0.5). The second term reflects the sum of similarities of each contextual feature *k* for the current item *c* and the corresponding element in memory chunk *v*, weighted by the mismatch penalty *MP* (set to 2.0). The final term represents noise, a random value from a normal distribution with mean of zero and variance *s* of 0.25, and introduces stochasticity in retrieval.

After making a classification, the instance is saved to memory and influences future decisions. However, if the email was a phishing email and it was incorrectly classified, the user is given feedback, and the decision is changed from ham to phishing to reflect the ground truth classification.

### **Multi-Armed Bandits Selection Algorithm**

The MAB problem is a well-studied online machine learning setting. In the classic problem, also known as stochastic MAB (Cesa-Bianchi & Lugosi, 2006), in each round, the learner (here the security team of the company) selects an arm (here an employee of the company) for an intervention (here sending a phishing email) and receives feedback (here the proficiency of the participant against the phishing attack) which is typically referred to as the reward. This process continues for a fixed number of rounds (referred to as the time horizon) and the goal is to maximize the total reward observed by the learner.

The classic setting assumes the arms are static such that the distribution of rewards for each arm remains stationary regardless of past arm selections. This is not the case in our setting, as users react to training and potentially become less vulnerable to future phishing attacks. Various extensions to MAB have been proposed in the literature to model these reward distribution changes. The most general framework to model such scenario is what is known as the RMAB (Whittle, 1998) in which each arm is modeled as an MDP.

Since each arm represents an employee in our problem, the MDP can be used to model the progress of an employee throughout training. In general, an MDP is a quadruple consisting of (1) states (here the different degree of proficiency of the employee in detecting phishing attacks), (2) actions (here whether the training has been provided for the employee or not), (3) rewards or the value associated with being in each of the states (here whether or not the phishing attack can fool the employee in the employee's current state of proficiency) and the (4) transition probabilities which is a distribution over the possible next states given the current state and the chosen action (here how proficiency can change given the current level of proficiency and whether a training has been performed or not).

In our problem, we propose the following stylized MDP to model an employee. We assume there are two states, referred to as "good" and "bad" states. We further assume that there only two actions: a training intervention (action 1) and no intervention (action 2). The rewards for being in a good or bad state are assumed to be 1 and 0, respectively. The employee-dependent transition probabilities can be succinctly represented by 4 parameters:  $p_{gb}^1$ ,  $p_{gb}^2$ ,  $p_{bg}^1$ , and  $p_{bg}^2$ , where  $p_{xy}^i$  denote the probability of transfer from state *x* to state *y* when action *i* is taken.<sup>1</sup>

We used the cognitive model, described above, to generate the transition probabilities for each user cluster that were needed for the MDP. We simulated 1000 cognitive agents performing the task paired against a random selection algorithm. We defined a good state as a correct classification, and a bad state as an incorrect classification. Based on the model's sequence of decisions, probabilities were computed as the proportion of transitions from a good or bad state at time *t* to a good state at *t*+1 as opposed to a bad state at *t*+1, depending on the action (i.e., type of email sent) at time *t*.

While cognitive architectures and Markov Decision Processes (MDP) are quite different modeling approaches, they also share substantial similarities. Both embody the Markovian assumption of future behavior being probabilistically determined by the current state of the system and inputs from the environment. However, the current state for cognitive architectures consists of knowledge and skills held in memories, together with their activation, enabling both a more graded and combinatorial representation. Also, state transitions in cognitive architectures are largely determined by constrained mechanisms resulting from a theory of cognition, rather than needing to be trained from data. Therefore, unlike MDPs, cognitive architectures can make a priori predictions in the absence of data (Lebiere et al, 2003). Cognitive architectures can then be used to provide a high-fidelity model of human behavior on a limited set of available data, then run many times over new generalization conditions to provide large data sets for training MDPs (Sycara et al, 2015).

We highlight that in our formulation, while the states, the actions and the rewards are known, the transition probabilities for each of the employees are unknown and should be learned during the learning process. In general, RMAB problems are computationally hard and optimal solutions are only known for specific cases. We build on Whittle Index Q-Learning (WIQL), a recent algorithm proposed by Biswas et al. (2021), to design an algorithm which we call SuperArm-WIQL to solve our formulation of the RMAB problem. Intuitively, had we known everything about the MDPs in the RMAB problem, we could have used heuristic algorithms such as Whittle Index (Whittle, 1998) to decide which employee to target for intervention on any given round.<sup>2</sup> Without knowing the MDPs, one can use any off-the-shelf algorithm to simultaneously learn the parameters of the MDPs first before applying the Whittle Index heuristic. Biswas et al. (2021) use Q-Learning for this process and hence the name WIQL.

<sup>&</sup>lt;sup>1</sup> Since there are two states and two actions, it seems like to fully represent the transition probabilities we require 8 parameters. However, observe that  $p_{xy}^i + p_{xx}^i = 1$  for all states *xy* and action *i* as the transition will finally move to either of the two available states. Therefore, we can reduce the total parameter to only 4.

<sup>&</sup>lt;sup>2</sup> We ignore the issue of indexability and conditions in which the Whittle Index heuristic is optimal.

The downside of such an approach is that learning the parameters of the MDP for each employee separately will result in a time and computational cost which is proportional to the number of employees. In practice, each round of sending phishing emails is costly and furthermore, the amount of available phishing emails is limited. Hence, naively applying WIQL will be too time-consuming, slow, and impractical. To deal with this problem, we first cluster the employees (or arms) into different groups (or super arms) and combine the learning experiences of all the users together. We call this algorithm SuperArm-WIQL. In the extreme, where there is only one arm per group, SuperArm-WIOL reduces to WIOL but with a small number of groups (compared to the total number of employees) and sufficiently similar arms in each group, SuperArm-WIQL will converge to a good policy much quicker.

We performed a K-means cluster analysis on the set of users described above to minimize the within-cluster sum of squares based on the Initialized Length and Ham Proportion attributes. A scree plot revealed four clusters were optimal  $(SS_{bet}/SS_{tot} = 71.67\%)$ . Figure 1 shows the visualization of the four clusters, which we labeled according to their location in the landscape of Initialized Length and Ham Proportion: 1 = "high-high", 2 = "low-low", 3 = "low-high", and 4 = "high-low".



Figure 1: Cluster plot of simulated users





#### **Simulation Results**

The results of the RMAB simulation using the SuperArm-WIQL are presented in Figure 2, compared to Random and NoAction (no users selected for intervention) selection algorithms. Rewards are calculated as the sum of users in a good state (i.e., correctly classifying a given email) at each trial, and the plot shows the moving average reward with a window size of 50. To start the simulations, users are randomly assigned to states with 50% probability, and quickly transition toward good states. The NoAction and Random algorithms show that performance quickly plateaus as users align with the average transition probabilities given the possible actions. The results of the NoAction algorithm are a bit misleading because it only measures user proficiency in classifying ham emails (which is already high) and does not account for proficiency with phishing emails. Most notably, the results show that by selecting users strategically, the RMAB (blue) outperforms the Random algorithm (green) in terms of the number of users in good states, and continues to improve across trials, eventually outperforming the NoAction algorithm (red).

### **Cognitive Selection Algorithms**

We designed two versions of the cognitive selection algorithm. The cognitive selection algorithms use cognitive principles to select which users to send phishing emails to on each trial, given a budget of 20% on each trial. Both methods use a technique called model tracing to track a user's history of decision making (e.g., Anderson et al., 1995). For each trial, the algorithms store information about what email was presented to each user and what their decision was. This history is then used in the blending equation described above to compute probabilities of classifying an email as ham  $(V_{ham})$  or phishing  $(V_{phish})$ , without adding any noise  $\varepsilon_i$ .

The first method, Cog-Low, simply computes the overall probability of classifying an email as ham or phishing at time *t*, without using the partial matching term. Therefore, the probabilities only reflect the influence of recency and frequency of all past instances. The participants with the lowest probability of classifying an email as phishing are selected for intervention (i.e., are sent a phishing email), with the hypothesis that their future probability of classifying phishing emails correctly will improve. The algorithm thus seeks to always improve the worst users on each trial.

The second method, Cog-EV, uses a more complex calculation that weighs the anticipated future benefits of sending a phishing email, in terms of correctly classifying phishing emails, against the anticipated future costs, in terms of incorrectly classifying ham emails, to determine which users will most benefit from a phishing training intervention.

As another improvement over Cog-Low, Cog-EV includes the partial matching term to determine the probabilities of correctly classifying an email of category k (ham or phish). Similarities are computed by averaging across the similarity of instance i to all other instances of the same category k. After computing the initial probabilities, another phishing instance is added to the user's history to compute the future probabilities of correctly classifying a ham or phishing email given a phishing intervention. The expected value for sending a phishing email ( $EV_{intervention}$ ) is reflected by the equation:

$$EV_{intervention} = \frac{(V_{phish|intervention}^{t-1} - V_{phish}^{t}) - (V_{ham}^{t-1} - V_{ham|intervention}^{t-1})$$
(4)

where  $V_{phish}$  and  $V_{ham}$  are the probabilities of correctly classifying a phishing or ham email, respectively, and are derived via blending.

### **Cognitive Simulations**

We used instances of the cognitive model as simulated users to predict the effectiveness of the selection algorithms against humans. All simulations were seeded with the same initial random state and started with the same set of initialized users to ensure consistent replication. We used ACT-R's built-in mechanism for running multiple models in parallel. The selection algorithm determined which user to send phishing emails to on each trial. To minimize repeated presentation of emails per user, we used the 186 phishing emails from the PTT but combined the ham emails from both the PTT and the PEST, for a total of 177 ham emails. We compared the RMAB, Cog-Low, and Cog-EV algorithms to two baseline algorithms, NoAction and Random (random selection from a uniform distribution), resulting in 5 total conditions.

#### Results

The moving average accuracy across trials, with a window size of 50, is presented in Figure 3. The NoAction condition represents the high baseline accuracy in classifying ham emails correctly given no phishing training intervention. Between all other conditions, the RMAB and Cog-EV conditions perform best in terms of overall accuracy, but there is an interaction between phishing and ham accuracy such that phishing accuracy increases at the expense of ham accuracy. This reflects the tradeoff in signal detection due to frequency and recency effects.

Phishing accuracy improves the least in the RMAB condition, while the Random, Cog-Low, and Cog-EV conditions display similar improvements. However, the RMAB and Cog-EV conditions display the least decline in ham accuracy, while there is a greater decrease in Random.

and more so in Cog-Low. These results are however difficult to interpret because they do not reflect differences in user selection preferences. It is possible that some algorithms are sending users the type of email that they are most likely to get correct, thus artificially inflating the overall accuracy. Therefore, we examined which users are being sent phishing emails as well as unbiased signal detection measures.Figure 4 shows a scatterplot of the mean accuracy for phishing and ham emails for each user, colored according to the proportion of phishing emails received, which is normalized within each selection condition (z-score). The results reveal distinct selection profiles. Accounting for the distribution of phishing emails across clusters, depicted in Figure 5 (z-scored phishing proportions), the Random condition displays no selection preferences and user accuracy trends with their phishing proportion. The RMAB selects users with high email experience and most phishing emails (high-low), which incidentally are already good at classifying phishing emails, while users that are poor at classifying phishing but good with ham emails receive more ham emails (top left tail of scatterplot). The Cog-Low mostly selects users with high experience and fewest phishing (high-high) which hypothetically need the most intervention, while sending the fewest phishing emails to the group that needs least intervention (low-low). The Cog-EV mostly send phishing emails to the users with low email usage (low-low and lowhigh), which are ones in which a training intervention will be most impactful, while sending the fewest phishing emails to the high-low group. However, there are a number of users that receive many phishing emails and thus their ham accuracy suffers (bottom right tail of scatterplot). If false alarms are not costly for a user or organization (i.e., by not responding important emails or causing excessive verification work for the security team) then this may be an acceptable solution.

Finally, to get a sense of the overall improvement of users from the start of the training task ("Initial" state) to the end of the training task ("Final" state). We examined change ( $\Delta$ ) in *d*-prime scores from the first 20 trials of the task to the last 20 trials of the task. We used a loglinear adjustment to account for missing cells when computing the hit rates and false-alarm rates (Stanislaw & Todorov, 1999). The results in Figure 6 show that Random selection improves sensitivity for



Figure 3: Moving average accuracy across trials for each selection condition. Total (left) and by Email Type (right). ws = 50.



Figure 4: Scatterplot of individual ham and phishing accuracy colored by the normalized proportion (*z*-score) of phishing emails received within each selection condition.



Figure 5: Normalized mean proportion (*z*-score) of phishing emails sent to each cluster within each selection condition.



Figure 6: Delta *d*-prime scores from Initial state to Final state for each cluster within each selection condition.

users with lowest ham experience (high-low and high-high). The RMAB only improves the high-high even though they received the fewest phishing emails, but performance declines significantly for the low-low group. The Cog-Low improves performance more as the number of phishing emails presented increases. And lastly, Cog-EV is the only condition that improves sensitivity across all clusters.

### Conclusion

Our simulations demonstrate the benefits of personalized anti-phishing training for organizations. The cognitive model proved useful in estimating transition probabilities for the MDP, and the RMAB was effective at improving performance. However, selection preference analyses revealed potential shortcomings of each of the methods. For one, the reward function for the RMAB should be redesigned so that it learns to send phishing emails to those most in need of intervention instead of those doing well. Current research is exploring methods such as defining states in terms of only phishing accuracy, but this would only lead to improvements in phishing classification. Another method could be to define rewards in terms of the users that misclassify emails (i.e., rewarded for intervening on those users that needed it).

Overall, the Cog-EV algorithm proved most successful at increasing phishing detection while minimizing false alarms. Future research will aim at validating these simulation results in human laboratory experiments. One limitation of the current simulations is that users were only given phishing emails as training interventions. However, it may be more realistic for users to receive phishing emails with some small probability in non-intervention events. We will consider this design change and its implications for selection algorithms.

The cognitive solutions have lower computational overhead and thus an advantage of selecting users at the individual level, while the RMAB is limited to generalizing at the group level. It is likely that the RMAB would perform better as the number of clusters approaches the number of users. Therefore, future research is aimed at finding the optimal tradeoff between the number of clusters and computational costs. Future research is also aimed at implementing a method that takes advantage of the benefits of both RMAB and cognitive models. For example, the cognitive model could be used to provide updated transition probabilities or additional learning rate parameters that can be used by the RMAB. Such an approach could both alleviate computational costs for the RMAB while providing more accurate predictions of individual behavior than Q-learning.

Finally, in other future research we plan to investigate not only whom to target but also which specific email to send and how to tailor emails to an individual. Such an approach could leverage information about what email features an individual is most susceptible to (e.g., Singh et al., 2020) or the type of attack for which they are most likely to fall prey (e.g., email characteristics or social engineering strategy used, or topic relevance; De Kimpe et al., 2018; Lin et al., 2019; Parsons et al., 2019). In this sense, IBL cognitive models are perfectly suited for every aspect of personalized anti-phishing training.

### Acknowledgments

We thank Susobhan Gosh for their contribution of the code which we adapted to run our RMAB solution.

This research was sponsored by the Army Research Office and accomplished under MURI Grant Number W911NF-17-1-0370.

#### References

- Anderson, J. R., Corbett, A. T., Koedinger, K., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal* of Learning Sciences, 4, 167–207.
- Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Erlbaum.
- Anti-Phishing Working Group (2021). *Phishing Activity Trends Report:* 4<sup>th</sup> *Quarter* 2021. Retrieved from https://apwg.org/trendsreports/. APWG.
- Biswas, A., Aggarwal, G., Varakantham, P., & Tambe M. (2021). Learn to intervene: An adaptive learning policy for restless bandits in application to preventive healthcare. *Proceedings of the 30th IJCAI*, (pp. 4039–4046).
- Cesa-Bianchi, N., & Lugosi, G. (2006). Prediction, learning, and games. *Cambridge University Press*.
- Cranford, E. A., Singh, K., Aggarwal, P., Lebiere, C., & Gonzalez, C. (2021). Modeling phishing susceptibility as decisions from experience. *Proceedings of the 19th Annual Meeting of the ICCM* (pp. 44–49). Virtual.
- Curtis, S. R., Rajivan, P., Jones, D. N., & Gonzalez, C. (2018). Phishing attempts among the dark triad: Patterns of attack and vulnerability *Computers in Human Behavior*, 87, 174-182.
- Cyber Guru Phishing (2019). *Experiential learning to reduce phishing risk*. Retrieved from https://www.cyberguru.eu/cyber-guru-phishing/
- De Kimpe, L., Walrave, M., Hardyns, W., Pauwels, L., & Ponnet, K. (2018). You've got mail! Explaining individual differences in becoming a phishing target. *Telematics and Informatics*, 35(5), 1277–1287.
- Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance based learning in dynamic decision making. *Cognitive Science*, 27(4), 591-635.
- Han, L., Kashyap, A. L., Finin, T., Mayfield, J., & Weese, J. (2013). UMBC\_EBIQUITY-CORE: Semantic Textual Similarity Systems. In *Proceedings of the 2nd JCLCS* (pp. 44-52). Atlanta, GA.
- Hakim, Z.M., Ebner, N.C., Oliveira, D.S. *et al.* (2020). The Phishing Email Suspicion Test (PEST) a lab-based task for evaluating the cognitive mechanisms of phishing detection. *Behavioral Research Methods*.
- Lebiere, C. (1999). A blending process for aggregate retrievals. *Proceedings of the 6th ACT-R Workshop*. George Mason University, Fairfax, Va.
- Lebiere, C., Gray, R., Salvucci, D. & West R. (2003) Choice and learning under uncertainty: A case study in baseball batting. *Proceedings of the 25th Annual Meeting of the CSS*. (pp. 704-709). Mahwah, NJ: Erlbaum.

- Lin, T., Capecci, D. E., Ellis, D. M., Rocha, H. A., Dommaraju, S., Oliveira, D. S., & Ebner, N. C. (2019). Susceptibility to spear-phishing emails: Effects of internet user demographics and email content. ACM Transactions on Computer-Human Interaction, 26(5), 1–28.
- Parsons, K., Butavicius, M., Delfabbro, P., & Lillie, M. (2019). Predicting susceptibility to social influence in phishing emails. *International Journal of Human-Computer Studies*, 128, 17–26.
- Sheng, S., Holbrook, M., Kumaraguru, P., Cranor, L. F., & Downs, J. (2010). Who falls for phish? a demographic analysis of phishing susceptibility and effectiveness of interventions. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (pp. 373–382).
- Singh, K., Aggarwal, P., Rajivan, P., & Gonzalez C. (2019). Training to detect phishing emails: Effect of the frequency of experienced phishing emails. *Proceeding of the 63<sup>rd</sup> International Annual Meeting of the HFES*. Seattle, WA.
- Singh, K., Aggarwal, P., Rajivan, P., & Gonzalez, C. (2020). What makes phishing emails hard for humans to detect? In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 64(1). Chicago, IL.
- Stanislaw, H., & Todorov, N. (1999). Calculation of signal detection theory measures. *Behavior Research Methods*, *Instruments & Computers*, 31(1), 137–149.
- Sycara, K., Lebiere, C., Pei, Y., Morrison, D., Tang, Y., & Lewis, M. (2015). Abstraction of analytical models from cognitive models of human control of robotic swarms. *Proceedings of the 13<sup>th</sup> ICCM*. Groningen, NL.
- Whittle, P. (1998). Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25, 287–298.
- Yang, R., Zheng, K., Wu, B., Li, D., Wang, Z., & Wang, X. (2022). Predicting user susceptibility to phishing based on multidimensional features. *Computational Intelligence and Neuroscience*, 2022(7058972), 1–11.
- Yeoh, W., Huang, H., Lee, W-S, Jafari, F. A., & Mansson, R. (2021). Simulated phishing attack and embedded training campaign. *Journal of Computer Information Systems*, 1–20.

Proceedings of the 20th International Conference on Cognitive Modelling (ICCM 2022)

## Gamma Power as an Index of Sustained Attention in Simulated Vigilance Tasks

Taylor Curley (taylor.curley@cubic.com) Cubic Defense

Beavercreek, OH 45324 USA

Lorraine Borghetti (lorraine.borghetti.1.ctr@us.af.mil)

Air Force Research Laboratory WPAFB, OH 45433 USA

Megan B. Morris (megan.morris.3@us.af.mil) Air Force Research Laboratory

WPAFB, OH 45433 USA

#### Abstract

Performance on the psychomotor vigilance test (PVT; Dinges and Powell, 1985)—a common index of sustained attention—is affected by the opposing forces of fatigue and sustained effort, where reaction times and error rates typically increase across trials and are sometimes offset by additional efforts deployed toward the end of the task (i.e., an "end-spurt"; c.f. Bergum and Klein, 1961). In ACT-R (Adaptive Control of Thought-Rational; Anderson et al., 2004), these influences on task performance have been modeled as latent variables that are inferred from performance (e.g., Jongman, 1998; Veksler and Gunzelmann, 2018) without connections to directly observable variables. We propose the use of frontal gamma ( $\gamma$ ) spectral power as a direct measure of vigilant effort and demonstrate its efficacy in modeling performance on the PVT in both the aggregate and in individuals.

Keywords: ACT-R; EEG; fatigue; vigilance; microlapse

#### Introduction

A well-documented phenomenon in human performance research is the decline in performance during extended vigilance tasks due to cognitive and physical fatigue (c.f., Ackerman, 2011). The relative simplicity of common sustained attention tasks, such as the psychomotor vigilance test (PVT; Dinges and Powell, 1985), however, overshadows the complex and arcane connections between task outcomes and the neural mechanisms that give rise to these outcomes (Ishii et al., 2014; Kim et al., 2017). Despite this, changes in electroencephalographic (EEG) activity have been shown to provide a potentially reliable marker of mental fatigue (Tran et al., 2020).

One way to examine links between cognitive and neural mechanisms of sustained attention is by integrating data from behavioral and neural sources into a single model (Turner et al., 2017). In the ACT-R (Adaptive Control of Thought-Rational; Anderson et al., 2004) cognitive architecture, for example, researchers have begun to use event-related potentials (ERPs; Cassenti et al., 2011) and neural "blips" (Borst and Anderson, 2015) to link selection and duration of individual behaviors (productions) to EEG data. Despite extensive work on modeling the effects of time-on-task (Veksler and Gunzelmann, 2018) and sleep deprivation (Gunzelmann et al., 2009, 2015) on the PVT, ACT-R practitioners have yet



Figure 1: Observed relative gamma spectral power density across 2 minute time bins during the PVT. From Borghetti et al.  $(2021)^1$ .

to directly investigate the use of EEG in modeling fatiguerelated decrements during vigilance tasks.

We propose the use of estimated power in frontal gamma ( $\gamma$ ) wave forms in models of vigilant attention. Specifically, we argue that  $\gamma$  power measured during the PVT is a reliable index of sustained attention that reflects fatigue (e.g., performance decreases across time) and compensation (e.g., end-spurts) and can be directly applied to ACT-R parameters. To this end, we first review relevant investigations of EEG and ACT-R as they relate to vigilance and then introduce a method for incorporating  $\gamma$  power into ACT-R models of the PVT.

#### **EEG and Fatigue**

Recently, Borghetti et al. (2021) reported a study examining electrophysiological measurements from 34 young adult participants ( $M_{age} = 22.6$ ) over the course of a 10-min PVT in which participants were asked to respond immediately when a stimulus appears on the screen. Vigilance decrements during the PVT were exemplified by positive shifts in the distributions of reaction times, indicating increasingly slower responses, as well as increases in premature responses, i.e., false alarms (Doran et al., 2001). The results of the behavioral task also show a slight improvement in task performance in later trials, indicating an increase in effort, i.e. an "end-spurt" (e.g., Bergum and Klein, 1961).

The authors examined spectral power density, or an esti-

mate of the power in a neural signal given a particular frequency, over the course of the 10-min task, focusing on theta ( $\theta$ , 3-8 Hz), alpha ( $\alpha$ , 9-14 Hz), beta ( $\beta$ , 15-30 Hz), and gamma ( $\gamma$ , 30-100 Hz) wave forms<sup>1</sup>. The top half of Figure 1 illustrates the main findings of the study: Significant trends indicating decreases in  $\gamma$  spectral power across time-on-task in both the frontal (Fz) and parietal (Pz) regions of the brain, with a significant end-spurt toward the end of the task (Morris et al., 2020). Borghetti et al. (2021) concluded that frontal  $\gamma$ indexes the dynamic between fatigue and sustained attention in the PVT. This is consistent with similar research indicating increases in  $\gamma$  activity across vigilance tasks (Kim et al., 2017) and positive associations between task performance and amplitudes of  $\gamma$  oscillations (Herrmann et al., 2010).

#### Fatigue and Compensatory Effort in ACT-R

The ACT-R cognitive architecture provides a rich environment for investigating effort and fatigue in goal-driven tasks, where influences on effort during the task are modeled as parameters affecting the selection and execution of procedural knowledge, i.e., "productions". During the course of the task, the model selects productions with the greatest estimated utilities (U), or a parameter indicating the strength and appropriateness of a given behavior at a given time. In prior versions of ACT-R, utilities were determined by the probability that a given goal will lead to success (P), the value of the current goal (G), and the cost of using that particular production to reach a goal (C). In the current version, production selection is a function of an initial utility value parameter (v), noise on this value ( $\sigma^2$ ), and a threshold parameter ( $\tau$ ), wherein the model selects the production with the highest above-threshold utility value to fire. Production utility values can either remain static or can update to reflect changes in the model's environment, such as production learning/reinforcement (e.g., Lovett and Anderson, 1996).

Previous studies have conceptualized vigilant effort as a direct influence on production utilities. Jongman (1998), for example, used parameterized "motivation" in a previous ACT-R architecture to directly influence G, where greater G values represent greater effort allocated toward achieving a goal and lead to better task outcomes, but lower G values result in firing inappropriate productions. Belavkin (2001) also used G to influence utility values, but conceptualized the parameter as reflecting a more general "arousal" state, where decreases in G result in fewer above-threshold productions, resulting in "giving-up" behavior. In contrast, Gunzelmann et al. (2009) simulated fatigue by imparting its effects on both utility values (through the G parameter) and  $\tau$  as a function of "arousal" (A), which is derived from biomathematical estimates of arousal (c.f. Van Dongen, 2004). The decrease in

utility and  $\tau$  values represent the deleterious effects of fatigue and efforts enacted to compensate for fatigue, respectively. Gunzelmann et al. (2009) also incorporated "microlapses", or simulated lapses in attention. Microlapses occur when the utility module is unable to select a production, such as when all utility values are lower than  $\tau$ . The occurrence of a microlapse results in a penalty to utility values and thus increases the probability of future microlapses. While the number of microlapses that occur during a simulated task is not controlled by the modeler, the penalty to utility values can be freely-estimated.

More recently, Veksler and Gunzelmann (2018) generalized decrements in arousal as stemming from the effects of time spent engaging in the task ("time-on-task") and simulated microlapses. Specifically, the authors estimate the utility of a production U by imposing a penalty on the initial production utility value (v) as a function of both the number of microlapses ( $N_{ml}$ ) and the time spent on the experiment (t):

$$U(t) = \upsilon \left[\lambda^{N_{ml}} (1+t)^{\rho}\right],\tag{1}$$

where v is the initial utility value parameter, *t* is time spent on the task (scaled to minutes),  $\lambda$  scales the effect of microlapses on utility values, and  $\rho$  scales the effect of time-ontask. As fatigue increases and production values decrease, the probability of sampling an inappropriate production increases, leading to increases in false alarms.

In contrast, the production utility selection threshold is only affected by time-on-task:

$$UT(t) = \tau (1+t)^{\kappa}, \tag{2}$$

where  $\tau$  is the initial utility theshold parameter and  $\kappa$  scales the effect of time-on-task on the threshold. Lower thresholds under conditions of fatigue allow the model to select productions whose  $\upsilon$  values have decreased. This compensation is imperfect, however, as lowering the production selection threshold also allows the model to fire productions that are not appropriate for the context. In models of the PVT, this leads to increases in false starts and misses.

### **Candidates for Integration**

We now review mechanisms for 1) translating  $\gamma$  spectral power to units appropriate for use in ACT-R simulations and 2) applying transformed  $\gamma$  estimates to the ACT-R cognitive architecture.

Scaling Spectral Power Estimates. Similar to previous research (e.g., Belavkin, 2001; Gunzelmann et al., 2009; Jongman, 1998; Veksler and Gunzelmann, 2018), we conceptualize sustained attention as a parameter  $\zeta$  that is typically bounded between zero and one. In the proposed model, however,  $\zeta$  can occasionally exceed its upper bound, meaning that parameterized effort cannot go below zero (meaning "absolute" fatigue), but can surpass unity (meaning "extra" effort). Thus,  $\zeta$  can capture decrements due to fatigue as well as compensatory efforts that offset fatigue, such as the end-spurt effect (Morris et al., 2020).

<sup>&</sup>lt;sup>1</sup>These results are based on a correction to the gamma spectral power analyses. In the original version of the paper, gamma estimates decreased sharply between 0 and 2 minutes and declined slightly across minutes 2 and 10. The corrected analyses indicate that gamma power increases between time bins 4 (6 - 8 m) and 5 (8 - 10 m), as shown in Figure 1.

### Proceedings of the 20th International Conference on Cognitive Modelling (ICCM 2022)

Param.	Description	Bounds	Value	ζ Model?
υ	Initial production utility value	[ 0.0, Inf ]	Free	Yes
τ	Initial production utility theshold	[ 0.0, Inf ]	Free	Yes
ρ	Production utility time-on-task penalty	[-1.0, 0.0]	Free	No
κ	Utility threshold time-on-task penalty	[-1.0, 0.0]	Free	No
λ	Microlapse penalty	$[\ 0.0, 1.0\ ]$	Free	Yes
φ	Conflict resolution time	N/A	0.05	Yes

Table 1: Descriptions of fatigue-related parameters in the ACT-R model of the PVT. The "Value" column indicates if a value is freely-estimated, and if not, what the value is fixed to. The " $\zeta$  Model?" column indicates if the parameter is included in the model that uses  $\gamma$  power as a performance moderator. All 6 parameters are included in the full ("Fatigue") model (c.f. Veksler and Gunzelmann, 2018).

One way to normalize fatigue moderator values is by adjusting the values to the smallest value and the range of the values. This normalization method has been used to scale biomathematical estimates of arousal in previous investigations of the PVT (Gunzelmann et al., 2009), where estimates start with high values and monotonically decrease as a function of time. An interesting aspect of this method that is reflected in the fatigue moderators proposed by Gunzelmann and colleagues (Gunzelmann et al., 2009; Veksler and Gunzelmann, 2018) is that the normalized values start at 1 (the highest possible value) and decrease with time-on-task, implying that performance cannot meet or exceed that from t = 1. Therefore, we opted to normalize  $\gamma_i$  to the first observation in order to simulate end-spurt effects.

Given a set of observed spectral power estimates (total or relative)  $\Gamma_i = {\gamma_{i,1}, ..., \gamma_{i,t}}$ , for participant *i* at time *t*, as well as the range of these values,  $\gamma_{r_i} = \text{range}{\gamma_{i,1}, ..., \gamma_{i,t}}$ , we can calculate effort as:

$$\zeta_{i,t} = 1 + \left(\frac{\gamma_{i,t} - \gamma_{i,1}}{\gamma_{r_i}}\right). \tag{3}$$

Here,  $\zeta_{i,1} = 1$  and all subsequent values are interpreted as diminished effort due to time-on-task ( $\zeta_{i,t} \leq \zeta_{i,1}$ ) or additional (i.e., compensatory) effort compared to baseline ( $\zeta_{i,t} \geq \zeta_{i,1}$ ), allowing the model to account for end-spurt effects.

Applying Fatigue Decrements. The theoretic interpretation of  $\gamma$  with respect to vigilance is intentionally vague (i.e., an index of sustained attention) and does not allow for a straightforward implementation of the  $\zeta$  parameter in the ACT-R architecture. In these simulations, we integrate parameterized effort in a linear function with the initial production utility parameter  $\upsilon$  (similar to Eq 1) with brief lapses in attention. Therefore, the modulated utility value at a given time, U(t), can be calculated as a function of  $\upsilon$ ,  $\zeta$ , and the number of simulated microlapses ( $N_{ml}$ ):

$$U(t) = \upsilon \cdot [\lambda^{N_{ml}} \cdot \zeta_{i,t}]. \tag{4}$$

### The Current Study

The estimated penalties to utility values and thresholds in ACT-R are imperfect. First, they are "smoothed" approxima-

tions of behavior and are unlikely to directly capture stochastic, asynchronous intraindividual variability across time, leading to error inflation when fitting fatigue parameters to individual participants. Second, these mechanisms are indirect inferences resulting from observations of behavioral data and have yet to be empirically linked to outside indicators.

The current project addresses these issues by examining the extent to which neural indices of vigilance correspond to the deleterious effects of fatigue in the PVT. Specifically, spectral power density in  $\gamma$  waveforms is expected to accurately capture fatigue and effort in ACT-R models of task performance. We expect to find that models using the observed power density estimates (Equations 3 and 4) in place of fatigue functions (Equations 1 and 2) will fit the observed data as well as, if not better than, models with these functions in both the aggregate and at the level of the individual.

### Methods

Thirty-four adult volunteers ( $M_{age} = 22.60$ ;  $SD_{age} = 4.08$ ) recruited through the University of Dayton Research Institute (UDRI) participated in a single 2-h study session consisting of three experiment tasks with simultaneous EEG recording. The study was approved by institutional review boards at both UDRI and the Air Force Research Laboratory (AFRL), and all individuals were compensated for their participation in the study.

We provide a quick overview of the behavioral and electrophysiology methods below; further details can be found in Borghetti et al. (2021).

### **Behavioral**

Participants were asked to participate in a 10-m PVT task as a part of the 2-h study session. During the PVT, participants were asked to monitor a computer screen with a black background and to press "j" on a standard computer keyboard as quickly as possible to a target stimulus, i.e., white numbers in the middle of the screen displaying the time (in ms) since target onset. The time in between the previous response and the onset of a new stimulus, the interstimulus interval (ISI), was randomly selected from an interval between 2 and 10 s. ISIs were exact integers and selected from a uniform distribution.



Figure 2: Performance data by time bins for average RTs for valid trials (left) and average proportion of lapses (right). Error bars represent the standard error of the mean.

### EEG

Briefly, participants were fitted with an EEG cap with 64 electrodes, with 2 flat, unlinked electrodes applied to the mastoids. These data were processed using custom MAT-LAB scripts along with the EEGLAB toolbox (Delorme and Makeig, 2004). After applying a 1 Hz high-pass filter and removing artifacts, these data were epoched into segments of  $\pm 1500$  ms with respect to stimulus onset and divided into five, 2-m time bins. For the gamma spectral analysis, we assayed power in the 70-100 Hz frequency band for frontal (Fz) and parietal (Pz) cortical regions.

### Computational

The computational model was programmed using a Julia language (Bezanson et al., 2017) implementation of the ACT-R cognitive architecture (Anderson et al., 2004). In ACT-R, the PVT has been modeled as a time-inhomogenous semi-Markov process consisting of three phases (Gunzelmann et al., 2009; Veksler and Gunzelmann, 2018): Wait, Attend, and Respond. The Wait production occurs prior to stimulus onset in anticipation of the next trial, while the Attend and Respond productions occur after a critical stimulus has been visually processed and after the decision has been made to engage in a response, respectively. These productions typically occur in the Wait-Attend-Respond sequence, but the order can be disrupted if an inappropriate production is selected on the basis of low utility values (U). This can lead to *false starts*, where the Respond production is selected in the absence of a valid stimulus (i.e., RTs < 150 ms), and *lapses*, where the model fails to select the Attend or Respond productions in the presence of a valid stimulus (i.e., RTs > 500 ms). Additionally, response latency is penalized whenever there are no productions that exceed the production utility threshold (UT)by adding 50 ms for each occurrence (microlapse; c.f. Gunzelmann et al., 2009).

Importantly, the ACT-R model of the PVT simulates fatigue by applying a penalty to a) only initial utility values (Belavkin, 2001; Jongman, 1998) or b) both initial utility values and utility thresholds (Gunzelmann et al., 2009, 2015; Veksler and Gunzelmann, 2018). Here, we only penalize utility values derived from Equations 3 and 4 based on re-scaled gamma power estimates. Table 1 provides descriptions of the parameters, the ranges of possible values, and the models that



Figure 3: Best-fitting estimates for v and  $\tau$  (left),  $\lambda$  (top right), and associated AIC values (bottom right).

they are used in.

### Results

#### **Behavioral**

We performed statistical analyses on responses categorized into 3 types: False starts (RTs < 150 ms), lapses (RTs > 500 ms), and valid responses (150 ms  $\leq$  RTS  $\leq$  500 ms). For computational ease, we binned the data into five, 2-m bins and applied an inverse transformation to the RTs, i.e., 1/(RT \* 1000) (Ratcliff, 1993).

A repeated measures ANOVA on the aggregated inverted RT values with a Greenhouse-Geisser correction on the degrees of freedom (W = 0.53, p = 0.02) indicates that the effect of time bin is significant, F(2.89,98.41) = 11.54, p < 0.05, where average RTs increase between the first and fourth time bins (i.e., minutes 0 - 8), but decrease slightly in the fifth time bin (i.e., minutes 8 - 10; c.f. Figure 2). A similar oneway logistic GLM on lapses indicates that the log-odds of this type of response change across time bins, F(4,3652) = 3.48, p < 0.05, where lapse rates decrease between bins 1 and 2, increase between bins 2 and 4, and then decrease again between bins 4 and 5 (c.f. Figure 2). A one-way logistic GLM indicates that the probability of a false start on any given trial is not different across time bins, F(4,3651) < 0.1.

#### **Spectral Power**

For frontal  $\gamma$  (Figure 1), a Friedman test on total power estimates across time bins is significant,  $\chi^2(4) = 11.3$ , p = 0.02. Follow-up paired comparisons indicate that estimates increase significantly between bins 2 and 3, p < 0.05, decrease significantly between bins 3 and 4, p < 0.05, and increase with marginal significance between bins 4 and 5, p = 0.07, although only the significance of the first comparison survives after Bonferroni corrections to the degrees of freedom.

### Computational

We estimated the parameters for two different models—one using the fatigue moderators described by Veksler and Gunzelmann (2018) and another using gamma power estimates—using the data from individual participants and ag-

### Proceedings of the 20th International Conference on Cognitive Modelling (ICCM 2022)

		Parameters				Fit Indices			
Estimate	Model	υ	τ	ρ	κ	λ	-2LL	AIC	BIC
Aggregate	Fatigue	4.01	2.90	-0.28	-0.20	0.98	5829.99	5839.99	5877.58
	Gamma	3.15	2.07	-	-	0.74	3920.76	3926.76	3949.25
	Fatigue	5.78	0.32	-0.41	-0.17	0.81	6591.08	6601.08	6604.58
Individual		(0.51)	(0.08)	(0.05)	(0.03)	(0.03)	(227.69)	(227.69)	(227.69)
maividuai	Gamma	3.36	2.74	-	-	0.73	4090.21	4096.21	4098.01
		(0.08)	(0.07)	-	-	(0.01)	(424.64)	(424.64)	(424.64)

Table 2: Best-fitting parameters for aggregated data (top) and summary statistics of the best-fitting parameters for individuals (bottom). For individuals, we report the means and standard errors of the mean (in parentheses) of these estimates. "Fatigue" refers to models using the decrement parameters described by Veksler and Gunzelmann (2018) while "Gamma" refers to the proposed model.



Figure 4: Reaction time distributions for valid responses across time bins for observed RTs (blue) and simulated RTs generated using the Gamma model (yellow).

gregated across all participants. Model fit was calculated using the summed log-likelihoods of the simulated RT data to log-normal distributions based on the observed RTs. We used a simplex search algorithm via Optim.jl (Mogensen and Riseth, 2018) to find the parameter values that maximized the likelihood of the two PVT models given the observed data. We repeated the optimization procedure 15 times for each set of data, using new starting values on each iteration to avoid local minima. Table 2 details the best-fitting parameters by data source ("Aggregate" vs. "Individual") and by the type of model ("Fatigue" vs. "Gamma").

Overall, the model using gamma spectral power density as a direct influence on utility values provides a better fit to the observed data than the model using established computational fatigue moderators. For the aggregated data, the difference in fit statistics suggest that there is decisive evidence (Kass and Raftery, 1995) in favor of the Gamma model,  $\log B_{10} =$ 1928.33. Similarly, the difference in average fit values across all participants for the two models also suggests that there is decisive evidence in favor of the Gamma model,  $\log B_{10} =$ 2506.57. Across individuals, the Gamma model is favored over the Fatigue model for all but 5 of the 34 participants in the study.

### Discussion

In this paper, we introduced an ACT-R model of vigilant attention that directly integrates frontal  $\gamma$  spectral power density estimates into the parameters of the model that influence task performance. We compared the ability of the new model to fit observed RT data to that of a similar model of PVT performance and found that the proposed model provides a better fit to both aggregated and individual data than previous models of fatigue. These results suggest that frontal  $\gamma$  power estimates can be used as a measure of sustained attention and effort in models of vigilance.

The proposed model represents an initial step in developing models of fatigue and vigilance that incorporate directlyobservable neural data. In this model, changes in observed neural data simply constrain the parameters of the behavioral model, i.e., a "direct-input approach" (Turner et al., 2017), implying a unidirectional influence. Future models, however, will need to simultaneously account for both neural and behavioral data and account for the bidirectional relationship between the two. Similarly, the use of frontal  $\gamma$  power in our model represents only one potential application of EEG data in cognitive models; our future research will use similar models to explore how other neural indices, such as beta ( $\beta$ ) and alpha ( $\alpha$ ) frequency bands, can be used as observable estimates of fatigue and arousal in computational models of vigilance.

### 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-2022-1771. The authors thank Bella Veksler and Chris Fisher for their help in model development and comments on the paper.

### References

- Ackerman, P. L. (2011). *Cognitive fatigue: Multidisciplinary perspectives on current research and future applications.* American Psychological Association.
- Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060.
- Belavkin, R. V. (2001). Modelling the inverted-U effect in ACT-R. In *Proceedings of the 2001 Fourth International Conference on Cognitive Modeling*, pages 275–76.
- Bergum, B. O. and Klein, I. C. (1961). A survey and analysis of vigilance research. Technical report, No HUMRRO-RR-8. George Washington University, Alexandria, VA. Human Resources Research Office.
- Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98.
- Borghetti, L., Morris, M., Rhodes, L. J., Haubert, A., and Veksler, B. (2021). Gamma oscillations index sustained attention in a brief vigilance task. International Annual Meeting of the Human Factors and Ergonomics Society.
- Borst, J. P. and Anderson, J. R. (2015). The discovery of processing stages: Analyzing EEG data with hidden semi-Markov models. *NeuroImage*, 108:60–73.
- Cassenti, D. N., Kerick, S. E., and McDowell, K. (2011). Observing and modeling cognitive events through eventrelated potentials and ACT-R. *Cognitive Systems Research*, 12(1):56–65.
- Delorme, A. and Makeig, S. (2004). EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134(1):9–21.
- Dinges, D. F. and Powell, J. W. (1985). Microcomputer analyses of performance on a portable, simple visual RT task during sustained operations. *Behavior Research Methods, Instruments, & Computers*, 17(6):652–655.
- Doran, S. M., Van Dongen, H. P., and Dinges, D. F. (2001). Sustained attention performance during sleep deprivation: evidence of state instability. *Archives of Italian Biology: Neuroscience*, 139(3):253–267.
- Gunzelmann, G., Gross, J. B., Gluck, K. A., and Dinges, D. F. (2009). Sleep deprivation and sustained attention performance: Integrating mathematical and cognitive modeling. *Cognitive Science*, 33(5):880–910.
- Gunzelmann, G., Veksler, B. Z., Walsh, M. M., and Gluck, K. A. (2015). Understanding and predicting the cognitive effects of sleep loss through simulation. *Translational Issues in Psychological Science*, 1(1):106–115.
- Herrmann, C. S., Fründ, I., and Lenz, D. (2010). Human gamma-band activity: a review on cognitive and behavioral correlates and network models. *Neuroscience & Biobehavioral Reviews*, 34(7):981–992.

- Ishii, A., Tanaka, M., and Watanabe, Y. (2014). Neural mechanisms of mental fatigue. *Reviews in the Neurosciences*, 25(4):469–479.
- Jongman, G. (1998). How to fatigue ACT-R. In Proceedings of the second European conference on cognitive modelling, pages 52–57. Nottingham University Press.
- Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal* of the American Statistical Association, 90(430):773–795.
- Kim, J.-H., Kim, D.-W., and Im, C.-H. (2017). Brain areas responsible for vigilance: an eeg source imaging study. *Brain Topography*, 30(3):343–351.
- Lovett, M. C. and Anderson, J. R. (1996). History of success and current context in problem solving: Combined influences on operator selection. *Cognitive Psychology*, 31(2):168–217.
- Mogensen, P. K. and Riseth, A. N. (2018). Optim: A mathematical optimization package for Julia. *Journal of Open Source Software*, 3(24):615.
- Morris, M. B., Haubert, A. R., and Gunzelmann, G. (2020). Beyond the vigilance end-spurt with event-related potentials. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 64, pages 1258–1262. SAGE Publications Sage CA: Los Angeles, CA.
- Ratcliff, R. (1993). Methods for dealing with reaction time outliers. *Psychological Bulletin*, 114(3):510.
- Tran, Y., Craig, A., Craig, R., Chai, R., and Nguyen, H. (2020). The influence of mental fatigue on brain activity: Evidence from a systematic review with meta-analyses. *Psychophysiology*, 57(5):e13554.
- Turner, B. M., Forstmann, B. U., Love, B. C., Palmeri, T. J., and Van Maanen, L. (2017). Approaches to analysis in model-based cognitive neuroscience. *Journal of Mathematical Psychology*, 76:65–79.
- Van Dongen, H. (2004). Comparison of mathematical model predictions to experimental data of fatigue and performance. Aviation, Space, and Environmental Medicine, 75(3):A15–A36.
- Veksler, B. Z. and Gunzelmann, G. (2018). Functional equivalence of sleep loss and time on task effects in sustained attention. *Cognitive Science*, 42(2):600–632.

# Modeling Short-term Fatigue Decrements in the Successive/Simultaneous Discrimination Task

**Taylor Curley (taylor.curley@cubic.com)** Cubic Defense Beavercreek, OH 45324 USA

Megan B. Morris (megan.morris.3@us.af.mil) Air Force Research Laboratory WPAFB. OH 45433 USA

#### Abstract

Previous research using goal-directed computational models has demonstrated that microlapses, or brief disruptions in effortful cognitive processing, are related to decreases in vigilance as a function of time-on-task in the psychomotor vigilance test (PVT) (Veksler and Gunzelmann, 2018). We extended these computational accounts of fatigue to model performance in two vigilance tasks that differ with respect to demands on working memory, i.e., successive vs. simultaneous discrimination (Davies and Parasuraman, 1982). While task performance was not affected by working memory demands, simulations show that fatigue moderators successfully capture decreases in vigilance over time. Additionally, participants showed greater individual differences in model parameters related to task performance, but not in the effects of fatigue across time. These results highlight the importance of fatigue moderators in computational accounts of vigilance tasks.

Keywords: ACT-R; fatigue; vigilance; microlapse

#### Introduction

The ability to direct and sustain attention over prolonged periods of time is essential to normal functioning in adults. Specifically, the ability to sustain conscious processing of a particular set of stimuli for periods longer than 10 s, or "vigilant attention" (VA; Robertson and Garavan, 2004; Robertson and O'Connell, 2010; Langner and Eickhoff, 2013), is directly linked to performance on continuous detection tasks such as the psychomotor vigilance test (PVT), where participants are asked to respond immediately upon presentation of a stimulus (Dinges and Powell, 1985). The PVT has traditionally been used to demonstrate decreases in VA under conditions of fatigue, where increases in degree of sleep loss are positively associated with response errors and latency (Doran et al., 2001; Dorrian and Dinges, 2004; Gunzelmann et al., 2009b). While most studies examine changes in PVT performance over the course of multiple days (typically coupled with sleep deprivation), researchers have also detected and modeled vigilance decrements over the course of single experiment sessions (e.g., 30-min tasks; Veksler and Gunzelmann, 2018). These methods successfully simulate the effects of fatigue on a few brief vigilance tasks, such as the PVT and the Mackworth Clock Task (Mackworth, 1948); however, it is unclear whether these methods can account for vigilance decrements in other related tasks.

In this paper, we describe a computational account of performance on two vigilance tasks in which participants are asked to view stimuli comprised of pairs of vertical lines and respond when the stimuli meet certain criteria. Our primary goals were to a) examine differences in processing and performance between successive and simultaneous discrimination tasks, b) determine whether computational accounts of fatigue provide a better fit to observed data compared to baseline models, and c) examine differences in parameter estimates across tasks and individuals.

### **Accounts of Vigilance Decrements**

Theoretical accounts of VA share the idea that attention modulates performance on vigilance tasks, but differ on the exact mechanism. Underload accounts argue that vigilance decrements stem from "drifts" of attention away from the task, motivated by the monotony of the task (e.g., Robertson et al., 1997; Smallwood and Schooler, 2006). Overload accounts, however, argue the opposite: The taxing nature of vigilance tasks induces fatigue, resulting in "lapses" of attention that negatively affect performance as a function of time-on-task (c.f. Warm and Dember, 1998). Most computational accounts of fatigue are inspired by overload hypotheses of VA and treat alertness as a resource that is exhausted with fatigue and replenished with rest (Gunzelmann et al., 2009a). Specifically, performance on simple vigilance tasks, such as the PVT, has been conceptualized as a balance between fatigue and compensation, where individuals offset decrements by changing response behavior, such as lowering the requirements needed to initiate a response. This performance is additionally affected by small lapses in attention, termed microlapses, which are positively related to fatigue and time-on-task (Gunzelmann et al., 2009b; Veksler and Gunzelmann, 2018).

#### Simultaneous vs. Successive Discrimination

An important topic in vigilance research is understanding how fatigue affects the different cognitive processes that support task performance. This is especially true for the role of working memory (WM) capacity, which has been shown to be strongly correlated to lapses in vigilance (Unsworth et al., 2010) and, more specifically, to PVT performance (Unsworth et al., 2021). One method for understanding the link between WM and vigilance is by contrasting performance on *simultaneous* versus *successive* discrimination tasks (Davies and Parasuraman, 1982; Caggiano and Parasuraman, 2004). In simultaneous discrimination tasks, all of the information that is needed to correctly classify a target item is included in the



Figure 1: Examples of target trials during the lines task. In the Simultaneous condition, targets were either pairs of lines with different (A) or identical (B) lengths. In the Successive condition, targets were either pairs of lines that were both short (C) or both long (D).

trial. During successive discrimination tasks, however, the response requirements are such that the stimuli presented during a trial must be compared to a template of the target item held in WM. The WM requirements of successive discrimination tasks make them particularly sensitive to the effects of fatigue, where task performance declines more rapidly across trials compared to performance in a comparable simultaneous discrimination task (See et al., 1995; although also see Gartenberg et al., 2018).

### The Current Study

We manipulated simultaneous vs. successive discrimination in the current study using a task in which participants are asked to view pairs of lines that are centrally-located on a computer screen (Figure 1). During each trial, participants were shown pairs of black vertical lines for 150 ms following a variable interstimulus interval (between 1.3 and 1.7 sec). The lengths of the two lines (either 14.6 or 18 mm) were randomly chosen during each trial. In the Simultaneous condition, participants were asked to respond only when both lines were the same length or different lengths. In the Successive condition, participants were asked to respond only if both lines matched and were of a particular size (short or long). Here, template-matching is not needed in the Simultaneous condition, as observers need only to determine differences between lines in order to provide a response. In the Successive condition, however, a template of either two "short" or two "long" lines is needed for a comparison. We modeled performance in both of these tasks to better understand differences in performance due to WM capacity and fatigue.

### Methods

### Behavioral

The models were based on data collected from 24 young adult volunteers ( $M_{age} = 21.17$ ,  $SD_{age} = 2.23$ ) recruited through the University of Dayton Research Institute and surrounding area. Participants were asked to complete two experiment

	Description	Bounds	BM?	Fixed?
υ	Initial utility value	[0.0, ∞]	Yes	No
τ	Initial utility threshold	[0.0, ∞]	Yes	No
λ	Microlapse penalty	[0.98]	No	Yes
ρ	Utility ToT penalty	[-1.0, 0.0]	No	No
κ	Threshold ToT penalty	[-1.0, 0.0]	No	No
γ	Conflict resolution time	[0.05]	Yes	Yes

Table 1: Parameters of the ACT-R lines models with indications as to whether they are a) included in the baseline model (BM) and b) if they are fixed values or freely estimated. "ToT" = "time-on-task".

sessions lasting 2 hr each, where part of the study was to complete the successive or simultaneous discrimination tasks on separate days. We counterbalanced the order in which participants completed these tasks to mitigate the influence of one discrimination task over the other. The discrimination tasks each consisted of 100 practice trials (which are excluded from the statistical analyses reported in this paper) and 1,600 test trials, and took approximately 45 min to complete. All participants gave written informed consent in accordance with the Declaration of Helsinki and were compensated for their participation.

### Computational

We developed the model using the Adaptive Control of Thought-Rational, or ACT-R (Anderson et al., 2004), cognitive architecture, with inspiration from previous models of the PVT (Gunzelmann et al., 2009b; Walsh et al., 2017; Veksler and Gunzelmann, 2018). ACT-R models behavior as emerging from a series of if/then rules that govern which actions (or "productions") are selected in a given situation, which itself is governed by a central cognitive system. The productions that are selected are a function of a) the amount of activation and noise for any given production (i.e., *utility values*) and b) the minimum activation required for a production to be selected (i.e., utility threshold). The strength of any given production can change as a function of baseline activation, the number of times a production is selected, and the match between the outside environment and production specifications. Here, utility values and thresholds will be determined by parameters related to fatigue. Table 1 briefly lists the critical parameters we use in our models, descriptions of the parameters, and the specific simulations that they are included in.

For the tasks in the current study, the production rules can be divided into four stages for any given trial:

• **Pre-attentive**: Prior to stimulus onset (i.e., lines appearing on the screen), participants must withhold a response as they anticipate a signal. Here, the model selects the *Wait* production to fire continuously until another production is selected, such as when lines appear on the screen. Under conditions of fatigue, however, the model may select and fire the *Respond* production, even in the absence of a valid

stimulus. This simulates false start responses when no lines are presented on the screen.

- Attentive: Immediately upon detecting a visual stimulus, the model will fire the *Attend* production, which represents the relatively automatic process of attending to and harvesting information about a visual cue. Similar to the preattentive stage, the model can erroneously choose the *Respond* production immediately after the *Attend* production, which simulates false start responses that are quicker than conscious processing.
- Decision: After moving attention to a visual cue, participants must decide whether the stimulus meets the response criteria (Match production) or not (Mismatch production). For the simultaneous discrimination task, the model is able to make this determination using only the stimuli presented on the screen. For the successive discrimination task, however, the model is required to compare test stimuli to a template held in WM, which requires more time and effort, i.e., about 50 ms extra. In either case, if the Match production is selected, then participants prepare to give a keyboard response; otherwise, the model will select the Wait production in anticipation of the next trial. Incorrect responses, which increase under conditions of fatigue, occur when a) the *Mismatch* production is selected given a target stimulus ("Miss") and b) when the Match production is selected given a non-target stimulus ("False Alarm").
- **Response**: When the model has decided to respond, it fires the *Respond* production, which simulates the physical act of pressing the "j" key on a keyboard. Consistent with Fitt's Law (Fitts, 1954), the model takes approximately 300 ms to execute the movement at the beginning of the experiment and becomes quicker as a function of practice throughout the task.

Additionally, the model can fire the *Microlapse* production, which is a brief interruption in model processing (50 ms). Microlapses occur when there are no productions with activations that exceed the production selection threshold and increase as a function of fatigue, simulating lapses in VA during continuous response tasks (Gunzelmann et al., 2009b).

In our full model, fatigue penalizes both the utility values (U) of these target productions and the threshold of the selection mechanism that controls which production is executed (UT). Specifically, utility values at a given time *t* are a function of both time-on-task and occurrence of microlapses, such that:

$$U(t) = \mathbf{v} \times [\lambda^{N_{ml}} \times (1+t)^{\mathbf{p}}], \tag{1}$$

where v is the initial utility value,  $\lambda$  is a penalty for microlapses,  $N_{ml}$  is the number of microlapses that have occurred in a given cycle,  $\rho$  is a time-on-task penalty specific to utility values, and *t* is the amount of time (in minutes) spent in the task.

Model	Cond	υ	τ	ρ	κ	-2LL
Baseline	Sim.	1.17	0.56	-	-	1546.88
	Succ.	1.90	0.35	-	-	2016.71
Fatigue	Sim.	1.43	0.81	-0.18	-0.21	1374.88
	Succ.	2.03	1.02	-0.24	-0.20	1497.86

Table 2: Best-fitting parameters and associated fit for models fit to all data.

The production selection threshold is affected much in the same way; however, only time-on-task, and not the occurrence of microlapses, has a direct effect on  $\tau$ :

$$UT(t) = \tau \times (1+t)^{\kappa}, \tag{2}$$

where  $\tau$  is the initial utility threshold value,  $\kappa$  is the timeon-task penalty specific to the utility threshold and, *t* is the amount of time spent on the task (scaled to minutes).

We fit the observed experiment data from both tasks to two models: One without fatigue moderators ("Baseline Model") and one with fatigue moderators ("Fatigue Model"). In both models, we freely estimated the starting utility values ( $\upsilon$ ) and utility thresholds  $(\tau)$ . In the Fatigue Model, we additionally estimated the time-on-task penalties for utility values ( $\rho$ ) and the utility threshold ( $\kappa$ ). We fixed the conflict resolution time ( $\gamma$ ) and microlapse penalty ( $\lambda$ ) parameters to 50 ms and 0.98, respectively<sup>1</sup>, although only  $\gamma$  is present in both Baseline and Fatigue models. The models were fit using maximum likelihood estimation and approximate Bayesian computation with differential evolution (Turner and Sederberg, 2012) against the joint log-likelihoods of the observed vs. simulated reaction times (RTs) (log-normal distribution), hit rates (Binomial distribution), and false alarm rates (Binomial distribution). All models were developed using the Julia language (Bezanson et al., 2017) and fit using the Optim. jl (Mogensen and Riseth, 2018) and DifferentialEvolutionMCMC.jl (2022) packages.

### Results

Here, we present only a few analyses regarding the behavioral data before discussing model fit indices. The results of the experiment are described in more detail elsewhere (c.f. Morris et al., 2022).

#### **Behavioral**

We conducted a 2 (Condition: Simultaneous [Sim] vs. Successive [Succ]) x 4 (Block: 4 blocks of 400 trials) withinsubjects ANOVA, with Greenhouse-Geisser corrections on degrees of freedom where assumptions of sphericity were violated. For RTs, there was only a main effect of Block, F(1.56,31.13) = 6.95, p < 0.05, reflecting a significant increase between Block 1, M = 0.58, SE = 0.02, and Block 2,

<sup>&</sup>lt;sup>1</sup>We did not freely estimate these values because these values have strong precedence in the extant literature (c.f. Veksler and Gunzelmann, 2018) and because early simulations indicated that model fit was not affected by these parameters.

Cond.	υ	τ	ρ	κ	-2LL
Sim	3.63	1.74	-0.21	-0.20	1525.61
SIIII.	(0.40)	(0.31)	(0.02)	(0.02)	(95.81)
Succ.	3.62	1.53	-0.23	-0.20	1580.08
	(0.38)	(0.39)	(0.02)	(0.01)	(103.00)

Table 3: Means and standard errors of the mean (in parentheses) of the best-fitting parameters and associated fit for individual participants.

M = 0.63, SE = 0.02, p < 0.05. RTs in Block 3, M = 0.64, SE = 0.02, and 4, M = 0.63, SE = 0.02, did not significantly differ from each other, ps < 0.05. A similar pattern emerged for accuracy, where there was also a main effect of Block, F(1.59,31.73) = 3.83, p = 0.02. A partial interaction contrast indicates that accuracy was significantly higher for Block 1, M = 0.91, SE = 0.01, compared to all other blocks, Ms = 0.89, SEs = 0.01,  $\chi^2(1) = 6473.10$ . p < 0.05. There were no significant interactions between Condition and Block, ps > 0.05.

### Computational

We first fit the aggregated experiment data to the separate Baseline and Fatigue models. For both experiment conditions, the models with fatigue penalties (Sim: -2LL = 1374.88, BIC = 1417.27; Succ: -2LL = 1497.86, BIC = 1540.25) fit the observed data better than the Baseline models (Sim: -2LL = 1546.88, BIC = 1568.07; Succ: -2LL = 2016.71, BIC = 2037.90). Table 2 lists the separate parameters that were recovered from this process.

Given the better fit, we also estimated model parameters separately for each participant, but only using the Fatigue model of task performance (Table 3). For both the Simultaneous and Successive conditions (Figure 3), the estimated initial utility values varied greatly across participants ( $M_{sim} = 3.63$ ,  $SE_{sim} = 0.40$ ,  $M_{succ} = 3.62$ ,  $SE_{succ} = 0.38$ ), while the corresponding initial utility thresholds were lower and were less varied ( $M_{sim} = 1.74$ ,  $SE_{sim} = 0.31$ ,  $M_{succ} = 1.53$ ,  $SE_{succ} = 0.39$ ). Interestingly, estimates for both the utility value ( $M_{sim} = -0.21$ ,  $SE_{sim} = 0.02$ ,  $M_{succ} = -0.23$ ,  $SE_{succ} = 0.02$ ) and utility threshold ( $M_{sim} = -0.20$ ,  $SE_{sim} = 0.02$ ,  $M_{succ} = -0.20$ ,  $SE_{succ} = 0.01$ ) time-on-task penalty parameters were similar and exhibited little variation. These estimates did not differ significantly between the two experiment conditions, ps > 0.05.

As expected, the initial utility value and threshold estimates were correlated, r = 0.77, t(43) = 8.01, p < 0.05, reflecting the need for productions to exceed the selection threshold. Initial utility values were significantly correlated with utility time-on-task, r = -0.42, t(43) = -3.03, p < 0.05, and threshold time-on-task, r = 0.40, t(43) = 2.84, p < 0.05, parameter estimates. Similarly, the initial threshold values were also significantly correlated with utility time-on-task, r = -0.70, t(43) = -6.34, p < 0.05, and threshold time-on-task, r = 0.32, t(43) = 2.19, p < 0.05, parameter estimates. The



Figure 2: Parameter estimates across participants for the initial utility and threshold values (a) as well as the utility and threshold time-on-task penalties (b) for the Simultaneous condition.



Figure 3: Parameter estimates across participants for the initial utility and threshold values (a) as well as the utility and threshold time-on-task penalties (b) for the Successive condition.

two time-on-task parameters were not significantly related to each other, r = 0.02, t(43) = 0.10, p = 0.92.

### Conclusions

These simulations support previous computational accounts of fatigue mechanisms (e.g., Gunzelmann et al., 2009b, 2015) and suggest that accounting for the effects of fatigue in a brief vigilance task provides a better fit to observed experiment data compared to models that do not account for fatigue, regardless of the WM requirements in the experiment task, i.e., Simultaneous vs. Successive conditions. They also suggest that penalties to both production utility values and production selection thresholds as a function of duration (Veksler and Gunzelmann, 2018) provide an accurate account of the decreases in response accuracy and increases in RTs in ACT-R models of the discrimination vigilance tasks. The parameters recovered from model-fitting indicate that while there are individual differences in factors related to general model performance, i.e., initial production utility values and production selection thresholds, this is not the case for parameters that describe decrements as a function of time-on-task, where all estimates for both the utility value and threshold penalty parameters showed little variation from -0.2.

The lack of differences between the two conditions in both behavioral and computational analyses contradict a resourcedepletion hypothesis of vigilance decrements (Caggiano and Parasuraman, 2004), where the additional WM requirements of the Successive lines task were expected to result in greater decreases in performance. The results are consistent, however, with a general resource-control theory of vigilance (Thomson et al., 2015), where greater decreases in vigilance are expected for tasks that are more difficult, but not for those that increase task engagement. In this particular task, requiring participants to hold a template of the target stimuli configured in WM might not have been sufficiently taxing in order to replicate the results of previous simultaneous/successive research (Parasuraman and Mouloua, 1987; Caggiano and Parasuraman, 2004). Alternatively, the Successive condition might have been sufficiently taxing, but also engaging enough to offset average differences in performance. Another possibility is that the stimuli used in the task (based on Parasuraman and Mouloua, 1987) were more taxing than previous speeded discrimination tasks, resulting in similar performance outcomes in both tasks. Regardless, the improvement in fit between the Baseline and Fatigue models implicates a performance decrement due to time-on-task, consistent with both theoretical and computational accounts of vigilance. Overall, these models extend previous accounts of fatigue and highlight the importance of accounting for decrements in brief vigilance tasks.

### 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-2022-1772. The authors thank Bella Veksler and Chris Fisher for their help with model development and comments on the paper.

#### References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060.

- Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98.
- Caggiano, D. M. and Parasuraman, R. (2004). The role of memory representation in the vigilance decrement. *Psychonomic Bulletin & Review*, 11(5):932–937.
- Davies, D. R. and Parasuraman, R. (1982). *The psychology* of vigilance. Academic Press, London.
- dfish (2022). itsdfish/DifferentialEvolutionMCMC.jl: v0.7.2.
- Dinges, D. F. and Powell, J. W. (1985). Microcomputer analyses of performance on a portable, simple visual RT task during sustained operations. *Behavior Research Methods*, *Instruments*, & *Computers*, 17(6):652–655.
- Doran, S. M., Van Dongen, H. P., and Dinges, D. F. (2001). Sustained attention performance during sleep deprivation: evidence of state instability. *Archives of Italian Biology: Neuroscience*, 139(3):253–267.
- Dorrian, J, R. N. L. and Dinges, D. F. (2004). Psychomotor vigilance performance: a neurocognitive assay sensitive to sleep loss. In Kushida, C., editor, *Sleep Deprivation: Clinical Issues, Pharmacology and Sleep Loss Effects*, pages 117–126. Marcel Dekker, Inc., New York.
- 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.
- Gartenberg, D., Gunzelmann, G., Hassanzadeh-Behbaha, S., and Trafton, J. G. (2018). Examining the role of task requirements in the magnitude of the vigilance decrement. *Frontiers in Psychology*, 9:1–13.
- Gunzelmann, G., Byrne, M. D., Gluck, K. A., and Moore Jr, L. R. (2009a). Using computational cognitive modeling to predict dual-task performance with sleep deprivation. *Human Factors*, 51(2):251–260.
- Gunzelmann, G., Gross, J. B., Gluck, K. A., and Dinges, D. F. (2009b). Sleep deprivation and sustained attention performance: Integrating mathematical and cognitive modeling. *Cognitive Science*, 33(5):880–910.
- Gunzelmann, G., Veksler, B. Z., Walsh, M. M., and Gluck, K. A. (2015). Understanding and predicting the cognitive effects of sleep loss through simulation. *Translational Issues in Psychological Science*, 1(1):106–115.
- Langner, R. and Eickhoff, S. B. (2013). Sustaining attention to simple tasks: a meta-analytic review of the neural mechanisms of vigilant attention. *Psychological Bulletin*, 139(4):870–900.
- Mackworth, N. H. (1948). The breakdown of vigilance during prolonged visual search. *Quarterly Journal of Experimental Psychology*, 1(1):6–21.
- Mogensen, P. K. and Riseth, A. N. (2018). Optim: A mathematical optimization package for Julia. *Journal of Open Source Software*, 3(24):615.
- Morris, M., Rhodes, J., Borghetti, L., Haubert, A., and Gunzelmann, G. (2022). Examining attentional and memory

mechanisms of the vigilance decrement with event-related potentials. Manuscript submitted for publication.

- Parasuraman, R. and Mouloua, M. (1987). Interaction of signal discriminability and task type in vigilance decrement. *Perception & Psychophysics*, 41(1):17–22.
- Robertson, I. H. and Garavan, H. (2004). Vigilant attention. In Gazzaniga, M. S., editor, *The cognitive neurosciences*, pages 631–640. MIT Press, Cambridge, MA.
- Robertson, I. H., Manly, T., Andrade, J., Baddeley, B. T., and Yiend, J. (1997). 'Oops!': performance correlates of everyday attentional failures in traumatic brain injured and normal subjects. *Neuropsychologia*, 35(6):747–758.
- Robertson, I. H. and O'Connell, R. (2010). Vigilant attention. In Nobre, A. C. and Coull, J. T., editors, *Attention and time*, pages 79–88. Oxford University Press, Oxford, UK.
- See, J. E., Howe, S. R., Warm, J. S., and Dember, W. N. (1995). Meta-analysis of the sensitivity decrement in vigilance. *Psychological Bulletin*, 117(2):230–249.
- Smallwood, J. and Schooler, J. W. (2006). The restless mind. *Psychological bulletin*, 132(6):946.
- Thomson, D. R., Besner, D., and Smilek, D. (2015). A resource-control account of sustained attention: Evidence from mind-wandering and vigilance paradigms. *Perspectives on Psychological Science*, 10(1):82–96.
- Turner, B. M. and Sederberg, P. B. (2012). Approximate Bayesian computation with differential evolution. *Journal of Mathematical Psychology*, 56(5):375–385.
- Unsworth, N., Redick, T. S., Lakey, C. E., and Young, D. L. (2010). Lapses in sustained attention and their relation to executive control and fluid abilities: An individual differences investigation. *Intelligence*, 38(1):111–122.
- Unsworth, N., Robison, M. K., and Miller, A. L. (2021). Individual differences in lapses of attention: A latent variable analysis. *Journal of Experimental Psychology: General*, 150(7):1303–1331.
- Veksler, B. Z. and Gunzelmann, G. (2018). Functional equivalence of sleep loss and time on task effects in sustained attention. *Cognitive Science*, 42(2):600–632.
- Walsh, M. M., Gunzelmann, G., and Van Dongen, H. P. (2017). Computational cognitive modeling of the temporal dynamics of fatigue from sleep loss. *Psychonomic Bulletin* & *Review*, 24(6):1785–1807.
- Warm, J. S. and Dember, W. N. (1998). Tests of vigilance taxonomy. In Hoffman, R. R., Sherrick, M. F., and Warm, J. S., editors, *Viewing psycholgy as a whole: The integrative science of William N Dember*, pages 87–112. American Psychological Association.

# Using a Cognitive Architecture to Consider Antiblackness in Design and Development of AI systems

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 USA

#### Abstract

How might we use cognitive modeling to consider the ways in which antiblackness, and racism more broadly, impact the design and development of AI systems? We provide a discussion and an example towards an answer to this question. We use the ACT-R/ $\Phi$  cognitive architecture and an existing knowledge graph system, ConceptNet, to consider this question not only from a cognitive and sociocultural perspective, but also from a physiological perspective. In addition to using a cognitive modeling as a means to explore how antiblackness may manifest in the design and development of AI systems (particularly from a software engineering perspective), we also introduce connections between antiblackness, the Human, and computational cognitive modeling. We argue that the typical eschewing of sociocultural processes and knowledge structures in cognitive architectures and cognitive modeling implicitly furthers a colorblind approach to cognitive modeling and hides sociocultural context that is always present in human behavior and affects cognitive processes.

**Keywords:** ACT-R; Concepnet; ACT-R/Φ; software engineering; sociogeny; sociogenic principle; antiblackness; AI

### Introduction

How might we use cognitive modeling to consider the ways in which antiblackness<sup>1</sup>, and racism more broadly, impact the design and development of AI systems? There has been a recent surge in scholarship approaching topics such as fairness, ethics, and equity in AI systems (e.g., see *AI*, *Ethics*, *and Society* or *Fairness*, *Accountability*, *and Transparency*, two conferences that were formed in recent years and focus on those topics.) Approaches in this space tend to focus on fairness and equity in the AI system itself, with solutions that detail ways to modify or test the AI system for forms of fairness (see Leben, 2020, for a discussion of *types* of fairness).

However, the current literature mostly fails to adequately consider two other important pieces of the equation:

- The person (or people) designing, developing, and/or deploying the AI system in question (especially from a *cognitive process* perspective)
- Sociocultural structures and institutions that mediate the way the AI system behaves and learns

within an environment. These same structures and institutions also mediate the individuals behind those systems.

When thinking about sociocultural systems and structures, we are pointing particularly to representations of the *Human* (Wynter, 2003; Wynter & McKittrick, 2015). Wynter (2003) traces how representations of the Human (that is, who is considered human and who is considered *other-than* human or *human-Other*) and how the dominant (socioculturally defined) modes of hierarchy that help define the Human have changed throughout recent history. Thinking in-terms of design and development, knowledge structures and representations used to design, develop, and deploy systems (and used by those systems to adapt or *learn*) exist within and re-present sociocultural contexts that designate some as *Other*.

Computational cognitive modeling offers an opportunity to develop computational accounts of the processes that lead to the creation and deployment of AI systems and related computational systems. Ritter (2019) makes a related point in their discussion of applications of cognitive modeling to the system design process. Though they particularly discuss the potential use in a spiral system development process (Pew & Mavor, 2007), the points generally apply to other development processes, especially those one might use in developing software (e.g., the Spiral model), often used to develop and deploy AI systems.

Beyond the typical cognitive models discussed by Ritter (2019), using computational cognitive modeling that includes physiological processes (e.g., Dancy, 2021) and those that include considerations of social processes (e.g., Orr et al., 2019) gives the representational space to consider, quantitatively and qualitatively, how social, cognitive, and physiological processes interact. The ability to understand the realistic interaction between these systems and their effect on behavior becomes especially important for questions related fairness, justice, and equity in the design, development, and deployment of AI systems (see Dancy & Saucier, 2022 for a related discussion of some of these questions that one should consider).

Orr et al. (2019) argues that cognitive architectures (and other systems in the "cognitive levels of scale") should be leveraged in concert with conceptual structures (and

<sup>&</sup>lt;sup>1</sup> Here, antiblackness refers to anti-Black racism. For more on connections between AI systems and antiblackness (albeit from a perspective sans cognitive-modeling), see Dancy and Saucier (2022)
dynamics) at the "social level of scale" to develop more complete simulations of human behavior with greater resolution. This paradigm of behavioral simulation, which they call the "Reciprocal Constraints Paradigm" (Orr et al., 2019), calls for cognitive agents to simulate social systems and abstract neurophysiology (upward constraints). The social systems should then constrain cognitive agents, while those cognitive agents constrain the interpretation of neurophysiological behavior (downward constraints).

Though this work differs in the representations that characterize the upward and downward constraints, there are similarities in recognizing the importance of physiological and social considerations in cognitive and behavioral processes. That is, despite the different *time scales* or *bands* (Newell, 1990) these processes have reciprocal effects albeit ones that differ depending on the scale of behavior. Given that we focus on cognitive models in the development process for this work, we focus the *rational band* of time through a cognitive model lens. The work spans both the cognitive and rational band in terms of generative model simulation (i.e., Ritter, 2019) considered here, but the representational space of the work is one that touches all four of the *biological, cognitive, rational, and social.* 

We use the ACT-R/ $\Phi$  hybrid cognitive architecture and the ConceptNet knowledge graph to consider computational representations that can be used to develop process models that span these bands. In the next sections, we provide more detail of these representations in rational and social bands that we use so that a cognitive model can be used to understand the effects of antiblackness on the design and development process (with a focus on a particular software engineering process). For more detail on the representations in the *biological* and *cognitive* band see (Dancy, 2021) for an understanding of the physio-affective and physio-cognitive connections, and see (Anderson, 2007; Ritter et al., 2019) for a more detailed look at the cognitive process representations.

# **Rational Band Representations**

Thinking through the design and development of AI systems at the rational band, is perhaps, the most natural fit for inquiry that aligns with cognitive modeling within cognitive architectures for the task of understanding. It is at this band that we start to think about behavior from the perspective of "knowledge-level" systems (e.g., see Newell, 1990 and also Lieto et al., 2018), at behaviors that span minutes to hours. Here, it is useful to use existing practices in design and engineering (particularly software engineering) as a guide for understanding the cognitive processes enacted within this space of time. We use a Software Engineering framework (Scrum, Schwaber & Sutherland, 2020) to think through the knowledge potentially used during AI system design and development. We also use work that connects processes from Data collection and use, AI development, and Software Engineering (Hutchinson et al., 2021) to move towards an understanding of design and development at this level.

In thinking through the knowledge that is used and enacted during the design and development of AI systems within the rational band of behavior, it's useful to consider the engineering framework that might be used to organize the development behavior and activities. Given the general popularity of agile methodologies and particularly Scrum, we use Scrum to think though behaviors within the scale of minutes to hours. Though Scrum can be thought of from the perspectives of social band as well, our considerations here are the behaviors that span minutes to hours (e.g., development of the product backlog, related agile practices such as the development of user-stories, or development of the system itself).

Hutchinson et al. (2021) argue that datasets used in AI and ML systems are a form of technical infrastructure and thus are produced by "goal-driven engineering" processes. Their discussion of Dataset development and curation as an engineering process becomes particularly useful in connecting their discussion of [AI and] ML datasets to considering the cognitive processing of the developer(s). Hutchinson et al. (2021) discuss datasets as forms of engineering models that represent "facts about the world that cannot be experienced directly, nor often replicated". These datasets are often collections of existing digital data, and thus pulled from existing digital knowledge infrastructure. Thus, one can think of these systems as providing a knowledgelevel representation (model) of the knowledge used by a person to enact actions within rational (and cognitive) timescales; relatedly, see (Sparrow et al., 2011) for a discussion on the increased importance of digital computational systems for human knowledge use. Thus, the use of some of these datasets can be extended beyond traditional AI/ML (e.g., Reinforcement learning, or Neural Network-based systems) systems, to generative cognitive models built within cognitive architectures and this may be warranted because these datasets can be thought of as a model of the (extended) knowledge systems used by humans to determine behavior, especially within the rational band. Using these datasets as a model of the knowledge used by designers and developers during cognitive processing and behavior within the *rational* band presents an opportunity to develop models that simulate multi-scale (or in this case, multi-band) representations. While we might use software engineering and design cognition perspectives to develop the task-focused procedural and declarative knowledge for a relevant cognitive model, some datasets can provide a useful engineering infrastructure for wider considerations of sociocultural knowledge (e.g., those knowledge structures that encode power structures and hierarchies) with those taskoriented procedures and knowledge.

# Social Band Representations

Understanding how existing social structures mediate behavior at the individual level is important for understanding contextualized behavior across time. In addition to physical structures and the affordances those structures may provide, it is also useful to consider the *knowledge* structures that are more directly related to behaviors in this band and how those may influence cognitive behavior, whether explicitly or implicitly. Given that such knowledge structures can be learned in diffuse ways across larger time-scales (indeed, knowledge and meaning taken from that knowledge can span generations), it is important to understand how this social (and cultural) knowledge might influence behavior during AI design and development. One can contend that a fundamental aspect of sociocultural knowledge is who is and is not seen as a part of the Human (Wynter & McKittrick, 2015) and as others have argued (e.g., Benjamin, 2019; Costanza-Chock, 2020; Noble, 2018) our ability to recognize the humanity is important in the way we design systems. The knowledge structures considered in racist hierarchies that perpetuate antiblackness are best thought of at the social band and time-scale because, though the context may change as environments change, these power structures and hierarchies represented in knowledge persist across time and space (McKittrick, 2006; Wynter, 2003).

To explore design and development from this perspective, large representations of digital knowledge (e.g., knowledge graphs) and large models that encode concepts and relations between concepts (e.g., word embeddings) can prove useful. These models can be thought of not only as technical infrastructure, but also as models of the world (as discussed in the previous section). The interest in the exploration of bias in these language/knowledge models, ultimately leading to a direct comparison to knowledge communicated by people (Caliskan et al., 2017), adds to the evidence that these models may be useful as a model of world, especially social, knowledge. Due to this primary concern of their connection to knowledge at time-scales in the social band, we discuss a particular model here. We use the ConceptNet knowledge graph and API (Speer et al., 2017) towards this aim of using an existing digital computational model of the world to consider antiblackness in design and development of AI systems. The open-source ConceptNet knowledge graph can be used by AI systems to attach meanings to words. Though the network itself is most robust in English and likely transfers some biases from English to other languages, it is a multilingual knowledge graph. The ConceptNet knowledge graph combines knowledge from several sources including crowd-sourcing, certain games, and some resources created by experts.

The ConceptNet API contains an integrated system that is a hybrid of several word embeddings and gives values of (among other things) *relatedness* between terms. Similar to previous work on connecting ACT-R to other sources of knowledge outside of the traditional declarative memory representation (e.g., Kelly et al., 2020; Salvucci, 2014), we are proposing to think through and model using a system that can represent declarative facts, but differs from the standard declarative memory system in ACT-R; that is, to use this *model of the world* to consider relations between concepts, how they encode social systems of power (such as those related to antiblackness), and how this might effect behavior during the engineering process (i.e., as discussed in the previous section).

# Considering Antiblackness in the Design and Development Process

Though ConceptNet has been through processes of "debiasing" (Speer, 2017), this has not necessarily resulted in the removal of representations of antiblackness if one audits the system with a more critical lens (e.g., see Dancy & Saucier, 2022). This "de-biased" representation of antiblackness is particularly interesting given that one can use the system to compare effects on computational cognitive models across versions (or perhaps, thinking from the human developer perspective, we can look at before and after bias training.) Thus, as also argued by Dancy and Saucier (2022), there exists an opportunity to think beyond just representation and bias by using this model of knowledge about the world and a cognitive model built within a cognitive architecture. We can begin to generate and better understand some ways that the infra-human (and other related racist ideas and concepts) may creep into decision-making. This is not to say that one can solely use these tools to explore antiblackness in AI design and development, but that they can serve as a complement to historical and sociocultural perspectives. existing Computational cognitive models can be used to help explore and probe the artifacts that digitize existing power structures that have produced (and continue to produce) these racist ideas, which are then consumed in a racist bootstrapping of knowledge and action. We also should emphasize that even if we are to move towards a potential process-based explanation of antiblackness at the cognitive and rational level, this does not relinquish the responsibility and agency of individuals and the groups that individual agents form: indeed, concepts such as ethical cognition (Bostrom & Yudkowsky, 2014) and racial literacy (Daniels et al., 2019) must remain an explicit goal even in the face of understanding the mediating cognitive processes.

# Interpreting relatedness in ConceptNet from a *cognitive* and *rational* perspective using ACT-R

Dancy and Saucier (2022) details the ways in which, despite debiasing processes, the system still shows problematic relatedness calculations between racialized concepts and particularly negative representations. As an example, when looking at relatedness between concepts related to humanity (or the lack of it) and racialized "man" (i.e., "black man", "white man"), the authors found "black man" to have a higher relatedness to terms such as "savage", "beast", and "inhuman". Racialized "woman" concepts (again. "black woman" and "white woman") are problematic, but (somewhat expectedly) in a different way. While "white woman" shows almost an exact match in relatedness as "woman", thereby making "woman" and "white woman" interchangeable, "black\_woman" is absent from the system (i.e., black woman is not a term in the whole knowledge\_graph and so relatedness is determined solely by a different algorithm than for the other concepts). As discussed below, these representations are further problematic when one considers the number of edges

between racialized concepts and other concepts within the knowledge graph.

From an ACT-R/Phi perspective, these differences in ConceptNet term relatedness values (and indeed edges between terms) are important when considering how a person (or cognitive model) may act given different situations. The relatedness can, essentially, be seen as an important component in a calculation of association strength. In a cognitive-process scenario (i.e., one which involves a typical affective and physiological state), this type of relatedness between terms may be important for Instance-based learning Gonzalez et al., 2003, as well as prospective memory and goal selection Altmann & Trafton, 2002, in decision-making (and also see Thomson et al., 2015). Furthermore, when combining these theoretical perspectives with more realistic physiological and affective variability (e.g., making those same decisions while sleep deprived or stressed), the effects may be multiplied.

Instance-based learning theory describes a feedback loop between retrieving declarative knowledge (instances) used to make a decision and the outcome of that decision. The stage of first recognizing the current situation is reliant upon using declarative memory systems. Within ACT-R, this means that the recognition of situations and the knowledge one uses in those situations is guided by the declarative knowledge most available, where the availability of knowledge concepts (typically chunks) is defined by the activation of declarative memory elements (see Anderson, 2007, pp. 91-134; Anderson et al., 2004 for a further discussion of declarative memory activation equations in ACT-R). Thus, a cognitive agent will rely partially on availability of potentially competing concepts to ultimately make a decision. Both the subsymbolic role of declarative memory (i.e., being driven by activation of a concept) and the symbolic role in making a decision mean that we not only may implicitly retrieve concepts related to human (or less-than human) capacities for understanding how we treat a representation of *black man*, but also that we may explicitly use these concepts to justify the decision to treat Black people as less-than (e.g., see Fincher et al., 2018). Relatedly, the availability of declarative memory (for our current example, the relatedness/similarity that ultimately affects declarative memory activation) also affects the choice of which goal to pursue (Altmann & Trafton, 2002). The potential goals (and thus problem space explored) by any cognitive agent will be limited by the ontological space that defines their concepts. Being more related to a brute, creature, or beast fundamentally changes the knowledge available for action, as well as the knowledge that will be used to justify and condone action; such knowledge relations help maintain an anti-Black space. Given that, similar to arguments made by Simon (1996) one can think of designing AI systems as being reliant on a designer and developer deciding the goal of the system, it's inner environment (including the technical infrastructure used to train a system and define it's state-action space), and it's outer environment (which the developer is often tasked with modeling or finding a model for and can be related to

the models used to train the inner environment), these conceptual relations become problematic even before considering a typical software engineering framework that will help to organize and guide such development.

Though the relatedness for both man and woman are high even for several of the less-than-human terms, this would prove less important for availability in most cases. This expectation stems from the fan effect (Anderson & Reder, 1999), which would signal that the large number of edges connected to man and woman means that it would be less potent in being used as a cue for other concepts and it is more likely that the more specific category (e.g., *black man*) would be applicable to many situations. Thus, the relatedness of man is less material to black man and in the case where black man is directly used as a concept, there is a stark contrast between the relatedness values for the human terms and the less-than-human terms. The equality between woman and white woman (in terms of relatedness) only means that the woman does not need to act as a cue to reach the same conclusions (as it appears the edge relations are such that woman has a heavy influence on the relatedness of white woman). This discussion says nothing for *black woman*, which must be extrapolated from other data as the concept is not connected to any other concepts in ConceptNet, not even woman; this signals the importance of bringing in *intersectional* (Collins, 2015; Crenshaw, 1989) analysis when understanding these systems.

These availability considerations from ConceptNet as a world knowledge model are intensified when one considers non-ideal physiological and affective states. Changes in affect and stress lead to differences in both declarative and procedural memory availability and selection (Dancy, 2021; Schwabe & Wolf, 2013) and can facilitate the switch between using more implicit memory strategies to guide decisionmaking and action selection (Schwabe et al., 2009). Thus, the implications of a less-than human ontological space are worsened by the fact that we can switch to more implicit memory strategies when under certain states, creating a higher potential to use the biased conceptual knowledge we've received from our environment. That is, any de-biasing attempts we might see in the form of training related to "diversity" or "equity", the developers are likely to be influenced by the dynamics of physiology and affect; most notably for our current purposes, those dynamics associated with stress. This becomes a practical issue for engineers and designers of AI systems as they are not likely to create these intelligent artifacts in a vacuum and under a perfect state, but very realistically while experiencing normal life stressors. Thus, without critically addressing these issues of anti-Blackness constantly and explicitly, we ensure the continuation of a cycle with a new justification.

Considering these results from the perspective of software engineering (particularly a Scrum/agile process for our purposes), we can think how this may affect the Scrum artifacts created. If a member of the team is creating *user-stories* (e.g., with the template of "As a *<user>*, I want *<to perform something>*, so that *<I can achieve some goal>*")



Figure 1. Sociocultural structures influence the Scrum process both through effects on team members during sprints, and through influence of peripheral development of the product backlog.

with these types of subsymbolic connections (not to say anything for explicit symbolic connections), this will shape the formation of these user stories. Typically, these user stories will then be the artifacts that make-up the product backlog, which is used to designate tasks within each sprint. Thus, these artifacts, which play a big role in design and development of a software (AI) system, will be heavily influenced by the knowledge of the developer, who themselves will have an internal memory/knowledge environment that represents an existing (social-scale) system of power and racial hierarchy. This is nothing to say of the other parts of meetings (e.g., daily scrums and sprint planning, reviews, and retrospectives) which themselves may result in problematic changes in development. Fig. 1 gives a high-level picture of how questions, choices, and artifacts created during processes within the Scrum framework will ultimately be will be mediated not by a cognitive system, which itself is influenced by the existing knowledge and also by stressors (partially mediated by physio-affective processes) experienced during this information processing.

This problem is further complicated when considering that, despite any de-biasing attempts we might see in the form of training related to "diversity" or "equity", the developers are likely to be influenced by the dynamics of physiology and affect; perhaps most notably for our current purposes, those dynamics associated with stress.

# "Look, a Negro" or Taking into Account the *Sociogenic Principle*.

Fanon (2008), the source of the quote in the section title, in discussing the experience of antiblackness in western contexts, and how fundamental, partially social definitions of

what it means to be human or other influence those placed in either categories within western sociocultural contexts, coins the term sociogeny. He puts forth this concept in addition to phylogeny and ontogeny as an additional layer that determines what it means to be human. Carrying this idea and argument forward, Wynter (2001) adapts the term to sociogenic principle. Wynter uses the sociogenic principle to theorize hybrid "nature/culture" modes of being human; Wynter and McKittrick (2015), and Wynter (2003), trace more recent (western) dominant modes of the Human through history. The sociogenic principle gives us an opportunity to seriously consider how our definitions of the cognitive architecture, or at least the treatment of architectures and cognitive models, may or may not encode fundamental, sociocultural specific aspects of human. As discussed by Wynter (2001), sociocultural knowledge (that operates at the timescales in the social-band) will have foundational effects on behavior causing a "sociocultural situation" to activate a "specific biochemical...correlate". Critically to our use of language-related models as digital models of the world, Wynter (2001) also links the sociocultural mode of the Human to language, particularly the "historico-raical schemas" which are elaborated through a "thousand anecdotes" (and also see a Dancy & Saucier, 2022 for a related discussion relating Fanon's treatment of sociogeny, language, and computational models like ConceptNet).

Nonetheless, the consistent effects of dominant sociocultural knowledge systems (especially those encoding systems of power and oppression such as race) have largely remained hidden and under explored, because cognitive architectures and cognitive models have tended to focus on behavior at the cognitive band of time (though simulating behaviors in the rational band)<sup>2</sup>. Systems and Models that encode world knowledge, such as ConceptNet, give another opportunity to consider how pervasive connections between what it means to be human and race may computationally mediate behavior within the *biological, cognitive, and rational time bands*. In some ways this perspective relates to SGOMS (West & Pronovost, 2009), Orr et al. (2019), and more generally Lieto et al. (2018), but we take aim specifically at racializing hierarchies as a fundamental organizing principle to the social world we operate within. Thus, we are perhaps a level above those in that we are thinking through how to fit (at least) one system of oppression (which is foundational to the current Western context that dominates, but is not exclusive to, the US) within an existing cognitive architecture.

# **Conclusion and Future Work**

Even with existing computational systems and models of knowledge, there remains work to be done in connecting these systems. In doing this, we seek to avoid multiple models in the rational and social band that ultimately, do not get us closer to understanding how sociocultural knowledge and systems fundamentally organize behavior at lower bands. Lieto et al. (2018) discusses this issue as something related to criticism put forth by Newell (1973), but at the rational band. Thus, in addition to the importance of grappling with our socioculturally contextualized definition of the Human and of the *other* [than human] as laid out by Wynter and McKittrick (2015), there lies an importance in specifying the organizing principles that we will use to develop computational models that span multiple levels.

We plan to continue this work through strengthening connections between ACT-R/ $\Phi$  and ConceptNet, with an exploration into better ways to combine the declarative memory equations present in ACT-R, the existing ConceptNet knowledge-graph, and the numberbatch system integrated into the ConceptNet API (work such as Salvucci, 2014, is instructive towards this goal). We also plan to explore related existing word embeddings to study how different underlying technical infrastructure (i.e., Hutchinson et al., 2021) and methods for determining vectors may affect models developed for the purposes of exploring antiblackness in AI design and development. We also plan to develop computational cognitive models that use world knowledge (starting with ConceptNet) to make decisions during software engineering processes.

Moving to sociocultural processes in models such as ConceptNet, this work would benefit from a more finegrained analysis of race (i.e., beyond Black and white); using theory posited by Bonilla-Silva (2015) may prove especially useful here. It would also be beneficial to expand beyond race to other sociocultural, power systems that intersect with race such as gender. As discussed in section *Interpreting*  relatedness in ConceptNet from a cognitive and rational perspective using ACT-R systems, we have explored some intersections, but more work is needed.

While there has been work in understanding bias in the development of AI systems, cognitive modeling with cognitive architectures has yet to be used to develop a computational process-level understanding of issues in that area. What's more specific focus of antiblackness in design and development, which itself has a "historico-social" context and is structural in ways we must understand, has rarely been explored. Additionally, when social systems have been approached in cognitive modeling, sociocultural systems of power that play a part in our sociocultural definition of the Human have been ignored, resulting in a colorblind approach to modeling. Using cognitive architectures in concert with existing knowledge (including language) models presents a promising method in which to computationally explore antiblackness in the development of AI systems.

## References

- Altmann, E. M., & Trafton, G. J. (2002). Memory for goals: an activation-based model. *Cognitive Science*, 26(1), 39-83.
- Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York, NY, USA: OUP.
- 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.
- Anderson, J. R., & Reder, L. M. (1999). The fan effect: New results and new theories. *Journal of Experimental Psychology: General, 128*(2), 186-197.
- Benjamin, R. (2019). *Race after technology: Abolitionist tools for the new jim code*. Medford, MA, USA: Polity Press.
- Bonilla-Silva, E. (2015). The Structure of Racism in Color-Blind, "Post-Racial" America. *American Behavioral Scientist*, 59(11), 1358-1376.
- Bostrom, N., & Yudkowsky, E. (2014). The ethics of artificial intelligence. In K. Frankish & W. M. Ramsey (Eds.), The Cambridge handbook of artificial intelligence (Vol. 1, pp. 316-334). Cambridge, UK: Cambridge University Press.
- Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, *356*(6334), 183-186.
- Collins, P. H. (2015). Intersectionality's Definitional Dilemmas. *Annual Review of Sociology*, *41*(1), 1-20.
- Costanza-Chock, S. (2020). *Design justice: Community-led practices to build the worlds we need*. Cambridge, MA, USA: MIT Press.
- Crenshaw, K. (1989). Demarginalizing the intersection of race and sex: A black feminist critique of

<sup>&</sup>lt;sup>2</sup> This is nothing to say for the ways in which diffuse systems of power and oppression are so foundational to sociocultural knowledge and behavior that they affect not only the perpetual

assumed decontextualization of behavior in studying it, but also *who* has an opportunity to study it.

antidiscrimination doctrine, feminist theory and antiracist politics. *University of Chicago Legal Forum, 1989*(1), 139-167.

- Dancy, C. L. (2021). A Hybrid Cognitive Architecture with Primal Affect and Physiology. *IEEE Transactions on Affective Computing*, 12(2), 318-328.
- Dancy, C. L., & Saucier, P. K. (2022). AI and Blackness: Towards moving beyond bias and representation. *IEEE Transactions on Technology and Society*, 3(1), 31-40.
- Daniels, J., Nkonde, M., & Mir, D. (2019). *Advancing racial literacy in tech*. Retrieved from New York, NY, USA: <u>https://datasociety.net/output/advancing-racial-literacy-in-tech/</u>
- Fanon, F. (2008). *Black skin, white masks*. New York, NY, USA: Grove Press.
- Fincher, K. M., Kteily, N. S., & Bruneau, E. G. (2018). Our humanity contains multitudes: Dehumanization is more than overlooking mental capacities. *Proceedings of the National Academy of Sciences*, 115(15), E3329-E3330.
- Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instancebased learning in dynamic decision making. *Cognitive Science*, 27(4), 591-635.
- Hutchinson, B., Smart, A., Hanna, A., Denton, E., Greer, C., Kjartansson, O., . . . Mitchell, M. (2021). Towards Accountability for Machine Learning Datasets: Practices from Software Engineering and Infrastructure. In proceedings of the Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 560–575.
- Kelly, M. A., Arora, N., West, R. L., & Reitter, D. (2020). Holographic Declarative Memory: Distributional Semantics as the Architecture of Memory. *Cognitive Science*, 44(11), e12904.
- Leben, D. (2020). Normative Principles for Evaluating Fairness in Machine Learning. *In proceedings of the Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society,* New York, NY, USA, 86–92.
- Lieto, A., Kennedy, W. G., Lebiere, C., Romero, O. J., Taatgen, N., & West, R. L. (2018). Higher-level knowledge, rational and social levels constraints of the common model of the mind. *Procedia Computer Science*, 145, 757-764.
- McKittrick, K. (2006). *Demonic Grounds: Black Women and the Cartographies of Struggle*. Minneapolis, MN, USA: University of Minnesota Press.
- Newell, A. (1973). You Can't Play 20 Questions with Nature and Win: Projective Comments on the Papers of this Symposium. *Visual information processing: proceedings*, 283-309.
- Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA, USA: Harvard University Press.
- Noble, S. U. (2018). Algorithms of oppression: How search engines reinforce racism. New York, NY, USA: NYU Press.
- Orr, M. G., Lebiere, C., Stocco, A., Pirolli, P., Pires, B., & Kennedy, W. G. (2019). Multi-scale resolution of neural,

cognitive and social systems. *Computational and Mathematical Organization Theory*, 25(1), 4-23.

- Pew, R. W., & Mavor, A. S. (2007). Human-system integration in the system development process: A new look. Washington, DC, USA: National Academies Press.
- Ritter, F. E. (2019). Modeling human cognitive behavior for system design. In S. Scataglini & G. Paul (Eds.), *DHM and Posturography*. San Diego, CA, USA: Academic Press.
- Ritter, F. E., Tehranchi, F., & Oury, J. D. (2019). ACT-R: A cognitive architecture for modeling cognition. *WIREs Cognitive Science*, 10(3), e1488.
- Salvucci, D. D. (2014). Endowing a Cognitive Architecture with World Knowledge. *In proceedings of the 36th Annual Conference of the Cognitive Science Society*, Austin, TX, USA, 1353-1358.
- Schwabe, L., Schächinger, H., de Kloet, E. R., & Oitzl, M. S. (2009). Corticosteroids Operate as a Switch between Memory Systems. *Journal of Cognitive Neuroscience*, 22(7), 1362-1372.
- Schwabe, L., & Wolf, O. T. (2013). Stress and multiple memory systems: From 'thinking' to 'doing'. *Trends in Cognitive Sciences*, 17(2), 60-68.
- Schwaber, K., & Sutherland, J. (2020). *The scrum guide*. Retrieved from <u>https://scrumguides.org/index.html</u>
- Simon, H. A. (1996). *The sciences of the artificial* (3rd ed.). Cambridge, MA, USA: MIT press.
- Sparrow, B., Liu, J., & Wegner, D. M. (2011). Google effects on memory: Cognitive consequences of having information at our fingertips. *Science*, 333(6043), 776-778.
- Speer, R. (2017). ConceptNet Numberbatch 17.04: better, less-stereotyped word vectors. Retrieved from <u>https://blog.conceptnet.io/posts/2017/conceptnet-</u> numberbatch-17-04-better-less-stereotyped-word-vectors/
- 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.
- Thomson, R., Lebiere, C., Anderson, J. R., & Staszewski, J. (2015). A general instance-based learning framework for studying intuitive decision-making in a cognitive architecture. *Journal of Applied Research in Memory and Cognition*, 4(3), 180-190.
- West, R. L., & Pronovost, S. (2009). Modeling SGOMS in ACT-R: Linking Macro- and Microcognition. *Journal of Cognitive Engineering and Decision Making*, 3(2), 194-207.
- Wynter, S. (2001). Towards the Sociogenic Principle: Fanon, Identity, the Puzzle of Conscious Experience, and What It Is Like to Be 'Black.'. In M. F. Duran-Cogan & A. Gomez-Moriana (Eds.), *National identities and sociopolitical changes in Latin America*. New York, NY, USA: Routledge.
- Wynter, S. (2003). Unsettling the Coloniality of Being/Power/Truth/Freedom: Towards the Human, After Man, Its Overrepresentation An Argument. *CR: The New Centennial Review*, 3(3), 257-337.

Wynter, S., & McKittrick, K. (2015). Unparalleled Catastrophe for Our Species? Or, to Give Humanness a Different Future: Conversations. In K. McKittrick (Ed.), *Sylvia Wynter: On being human as praxis*. Durham, NC, USA: Duke University Press.

# Argumentation-based Reasoning guided by Chunk Activation in ACT-R

Emmanuelle Dietz (emmanuelle.dietz@airbus.com)

Airbus Central Research & Technology, Hein-Sass-Weg 22 21129 Hamburg, Germany

#### Abstract

Argumentation is a widely studied topic in philosophy, psychology, and AI. In this paper, we are particularly interested in its psychological implications. According to Mercier and Sperber argumentation is the means for human reasoning. Here, we will investigate how the context plays a role in the argumentation process and bridges to lower levels of cognition. For this purpose the relevant knowledge within a given context determines the choice of the arguments by applying the spreading activation theory of memory. Relevant knowledge can be factual, conditional or hypothetical and, when in conflict, might have different strengths in relation to each other. We propose three comparison mechanism for choosing the winning argument for a given position. Different than in computational argumentation, we are not interested in an exhaustive search for arguments, but a guided process determined by the given context. By using the cognitive architecture ACT-R we specify this process through the spreading activation of chunks. Finally, we implement two models of conditional reasoning within the cognitive architecture ACT-R and evaluate them with the results of a famous reasoning task.

#### Introduction

Cognitive theories of reasoning investigate how humans reason to understand, model, and eventually predict their decisions. The adequacy of these theories is usually assessed by comparing their predictions to the experimental results of typical reasoning tasks (e.g., Byrne (1989), Wason (1968)) and by developing new experiments. Most of these reasoning tasks are designed as follows: Given some (causal) information, for instance in form of conditional sentences, such as "if A, then B" together with a set of given premises, humans are asked what can be concluded from this information.

According to Newell's (1990) classification of human experience and information processing mechanisms into the four bands of cognition, conditional reasoning might best be classified between the cognitive and rational bands. To facilitate the different aspects of human behavior into various levels (or bands) of cognition, Newell suggested the development of cognitive architectures. This proposal implied that different fields in the area of cognition need to link their work to each other. Cognitive architectures provide a formal specification of the structure of the brain, the functions of the mind, and how the structure explains the function, guided by the findings from decades of research. Within these cognitive architectures, the cognitive processes are organized as modular entities coordinated within one environment thus simulating human cognition. Even though bridging the gap between Newell's bands of cognition is still an open problem, the developed cognitive architectures (e.g. ACT-R (Anderson, 2007), SOAR (Laird, 2012)) had a significant contribution on providing formal methodologies.

In this paper, we will investigate conditional reasoning, where we are mainly interested in three aspects: (i) how do humans understand conditionals in the given context, (ii) how do they infer new information from that context, and (iii) how can (i) and (ii) be implemented such that they account for existing theoretical findings of lower levels of cognition. For addressing (i) and (ii), cognitive argumentation is chosen as the theoretical foundation, where well-known cognitive phenomena are formalized as cognitive principles and conclusions are derived based on the dialectic argumentation process. Arguments are usually understood symbolically. Yet, the process of building and choosing them, and then deciding which argument wins seems to be heavily guided by biases or heuristics, influenced by the given context, which might partially be modeled statistically. By exploiting the probabilistic functions in the cognitive architecture ACT-R (Anderson, 1990; Anderson, Byrne, Douglass, Lebiere, & Qin, 2004), we implement argumentation-based reasoning guided by chunk activation.

Finally, two models of argumentation-based reasoning in ACT-R will be presented and evaluated to data from the well-known Byrne's (1989) suppression task.

#### **Related Work**

Various (non-classical) logic-based approaches for conditional reasoning have been proposed in the past (Braine, 1978; Johnson-Laird, 1983; Johnson-Laird & Byrne, 1991; Rips, 1994; Polk & Newell, 1995; Stenning & van Lambalgen, 2008; Dietz, Hölldobler, & Ragni, 2012). However, only a few of them (Braine, 1978; Rips, 1994; Johnson-Laird, 1983; Johnson-Laird & Byrne, 1991; Chater & Oaksford, 1999) proposed a theory on the (internal) reasoning process itself. Up to now, only the (mental) model theory (Johnson-Laird, 1983; Johnson-Laird & Byrne, 1991) and some reasoning tasks have been embedded into ACT-R (Khemlani & Trafton, 2012; Ragni & Brüssow, 2010; Ghosh, Meijering, & Verbrugge, 2014).

Addressing the question of how humans integrate what is known and what is conjectured or observed to what is inferred to explain has been addressed by Weick (1995), who proposed the theory of Sensemaking. Sensemaking is about the process to search for contexts that make sense.

Lebiere et al. (2013) proposed computational models that specify how observed sensemaking behavior can be produced from elementary cognitive processes and modules. Among other aspects, they considered the process of information gathering and hypothesis updating. The authors' goal is to identify and understand the core mechanisms of cognitive biases generally. A sensemaking model for intuitive decision-making employing instance-based learning has been proposed by Thomson, Lebiere, Anderson, and Staszewski (2015). In the following section, we will briefly point to similarities between argumentation and sensemaking. A generally observed problem in the field of Cognitive Science is that many ad-hoc formulations of domain-specific models exist and therefore Thomson et al. (2015) suggest driving the field of cognitive modeling to the generalizability of models. Salvucci (2013) has addressed this aspect by integrating models through cognitive skill acquisition. In the PRIMs architecture, cognitive processes can be reused such that they are applicable in many different combinations (Taatgen, 2013). Serving a similar purpose for the case of reasoning, in this paper we will introduce cognitive principles, which are formalized task-independent assumptions made by humans.

# **Cognitive Argumentation**

Experiments by Mercier and Sperber (2011) have shown evidence that humans arriving at and justifying claims seems to be done through the construction of arguments. They state that arguments are the means for human reasoning. Without expanding on the formal details, we will here briefly introduce the theoretical foundation of our approach, Cognitive Argumentation (Dietz Saldanha & Kakas, 2019; Dietz & Kakas, 2020, 2021), where reasoning (or inference) is based on a dialectic argumentation process. In Cognitive Argumentation, argument construction is guided by cognitive principles. These arguments are built from argument schemes, which represent general links between information.

We will first introduce the relevant cognitive principles and then illustrate the dialectic argumentation process by an example.

#### **Cognitive Principles**

Cognitive principles are assumptions that humans make while reasoning. The specification of such principles helps us to explain why humans come to certain conclusions in particular when they diverge from valid conclusions in classical logic. Furthermore, the notion of a cognitive principle allows us to understand and distinguish between different types of reasoners.

The first two principles, **maxim of quality** and **maxim of relevance** are motivated by Grice's (1975) conversational implicature. The maxim of quality states that, if there is no reason to assume differently, humans believe that what they are told as factual information, is true ( $\Delta^{fact}$ ). The maxim of relevance states that humans believe what they are told is relevant ( $\Delta_{hvp}$ ). This maxim applies when humans perform some hypothesis generation to infer consequences, not based on facts, but based on what hypothetically could be true or false. The principles of necessary  $\binom{n}{\leadsto}$  and sufficient conditions  $(\stackrel{s}{\rightsquigarrow})$ , are motivated by Byrne, Espino, and Santamaría (1999) and Byrne (2005): Consider the conditionals If she meets a friend, then she will go to a play and If she has enough money, then she will go to a play. In the first conditional, she meets a friend is sufficient support for she will go to a play. This is a sufficient condition. For the second conditional, she has enough money can be understood as a necessary condition, i.e. the negation, she does not have enough money is plausible support for the negation of the conclusion, she will not go to the play. Together with the cognitive principle of hypothesis generation, the hypothesis that she does not have enough money functions as a disabling condition to the modus ponens conclusion that she will go to a play. Similarly, given that If she has free tickets, then she will go to the play, the hypothesis of the sufficient condition she has free tickets functions as an alternative condition and discards the condition she has enough money as necessary for the conclusion she will go to the play. This classification of necessary and sufficient conditions is dynamic and strongly depends on the context.

Different from valid inferences in classical logic, humans have the ability to reason from observations to explanations, which is sometimes called abduction. Abductive reasoning is motivated by the **maxim of inference to the best explanation** (Peirce, 1903). Additionally, the plausibility of explanations increases or decreases by setting them in contrast to the alternative explanations. So might the support for one explanation discount the support for the alternative explanations (Kelley, 1973; Sloman, 1994).

#### **Dialectic Argumentation Process**

We informally introduce the dialectic argumentation process (Baroni, Gabbay, Giacomin, & van der Torre, 2018): **Step 1.** Construct a root argument supporting the conclusion of interest, **Step 2.** Consider a counter-argument, **Step 3.** Find a defense argument, **Step 4.** Check if the defense argument is not in conflict with the root argument (in Step 1), **Step 5.** Add the defense argument to the root argument, **Repeat** 



Figure 1: The dialectic argumentation process is guided by cognitive principles. Acceptable arguments are in green and non-acceptable arguments in red.  $\uparrow$  shows attacks and/or weak defenses and  $\uparrow$  show strong attacks and/ or defenses.

from Step 2. This process is repeated until no other counterarguments (step 2) can be found. The extended root argument is then the acceptable argument supporting the conclusion of interest. Informally, conclusions follow credulously when they are supported by acceptable arguments. They follow skeptically when they are acceptable and there are no acceptable counter-arguments.

The intuition of this process will now be illustrated with the help of the previously introduced examples and Figure 1: Given If she meets a friend (f), then she will go to a play (p), assume that the condition is both sufficient  $(f \stackrel{s}{\rightsquigarrow} p)$  and necessary  $(\overline{f} \stackrel{n}{\leadsto} \overline{p})$ . Further, assume that we are given the factual information that She meets a friend (Figure 1, left). Let us start with the position that She will go to a play: 1. We build the (strong) argument  $\Delta_{f \xrightarrow{s} p}^{f}$  for p, from the fact that f and that f is a sufficient condition for p(1, p). 2. We build the counter argument  $\Delta_{\overline{f}\ \overline{f}\ mathcal{matrix}}$  from the hypothesis that *She does* not meet a friend  $(\overline{f})$  and that f is (also) a necessary condition for p. 3.-4. However,  $\Delta^f$  is a defense argument against  $\Delta_{\overline{f},\overline{f},\overline{n},\overline{p}}$ , as f is a (strong) fact. 5. The new argument for p stays  $\Delta_{f \leftrightarrow p}^{f}$  as f is already part of the root argument. The only counter-argument left is  $\overline{f}$  against which  $\Delta^{f}$  is trivially a strong defense (repeat). Finally, the root argument  $\Delta_{f,s,n}^{f}$  is an acceptable argument for the conclusion p.

Next, let us consider the arguments for  $\overline{p}$ , which can only be built from the hypothesis  $\overline{p}$ ,  $\Delta_{\overline{p}}$ , or  $\Delta_{\overline{f},\overline{f}} \overset{n}{\longrightarrow} \overline{p}$   $(1, \overline{p})$ .  $\Delta_{f}^{s} \overset{s}{\longrightarrow} p$  and  $\Delta^{f}$  are (strong) attacks against which  $\Delta_{\overline{p}}$  and  $\Delta_{\overline{f},\overline{f}} \overset{n}{\longrightarrow} \overline{p}$  cannot defend against. There is no acceptable argument for  $\overline{p}$ , thus pis a skeptical conclusion.

Let us consider the argumentation processes when we additionally receive the information that *If she has enough money* (*m*), she will go to a play (*p*), where she has enough money is a necessary condition for she will go to the play ( $m \stackrel{n}{\rightsquigarrow} p$ ). (Figure 1, right): 1. Starting with, (1, *p*)  $\Delta_{f \stackrel{s}{\rightsquigarrow} p}^{f}$  is a strong argument for *p*. 2. The attack  $\Delta_{m,\overline{m}\stackrel{n}{\longrightarrow} p}$  is built from the new conditional  $m \stackrel{n}{\rightsquigarrow} p$  and the hypothesis that she does not have enough money 3.-4. which can be defended against with the hypothesis that *She has enough money* ( $\Delta_m$ ). 5. This defense argument is added to the root argument, and defends against all its attacks  $(\Delta_{f \to p}^{f} \cup \Delta_{m})$ : This is an acceptable argument for *p*.

Consider now the process for the opposing position: 1. The (strong) argument for  $\overline{p}$  is  $\Delta_{\overline{m},\overline{m}}^{n} \overline{p}$ . 2.  $\Delta_{f^{s} \rightarrow p}^{f}$  attacks  $\Delta_{\overline{m},\overline{m}}^{n} \overline{p}$ , however 3.-4.  $\Delta_{\overline{m},\overline{m}}^{n} \overline{p}$  can defend itself against  $\Delta_{f^{s} \rightarrow p}^{f}$ , as necessary conditions ( $\overline{m}^{n} \overline{p}$ ) are stronger than sufficient conditions ( $f \xrightarrow{s} p$ ).  $\Delta_{\overline{m},\overline{m}}^{n} \overline{p}$  is also an acceptable argument for  $\overline{p}$ : Both p and  $\overline{p}$  are credulous conclusions.

**Sensemaking** We can draw parallels between the theory of sensemaking (Klein, Moon, & Hoffman, 2006) and the argumentation process, where sensemaking models can be analogously understood as arguments considering the description given by Klein, Phillips, Rall, and Peluso (2007)[115]. Initially, humans generate just-in-time mental models (i.e. local cause-effect connections) to explain events (Step 1). They then elaborate and question that model with inconsistencies (Step 2), fixate on the initial model, eventually discover inadequacies and compare alternative(s) (Step 3), reframe the initial model, and (if applicable) replace the model with another one (Step 4 and 5).

#### **Guided Argumentation Process**

It does not seem plausible, that humans rigorously follow such a step-wise procedure as described above but it is more likely that they are guided by some heuristics, which might depend on e.g. their simplicity, their strength, and their relevance in the context. In the following, we address this aspect by realizing a guided dialectic argumentation process in ACT-R.

# **Argumentation in ACT-R**

Two ACT-R models based on the theory of Cognitive Argumentation are presented in this section. The structure of both models is shown in Figure 2.

#### Tasks

The proposed models implement three tasks, *read*, *argue* and *respond*, where the last two is each specified with one control state. Model I follows sequentially the tasks, whereas the *read* and *argue* tasks in model II are intertwined.



Figure 2: Model I (left) and model II (right), where each (yellow) block in the middle (between the imaginal buffer and the declarative memory) represents a production rule. The background colors in the models correspond to the ACT-R modules on the right to top of the right model.

#### **Background Knowledge**

Model I (Figure 2, left) stores the conditions as either necessary or sufficient in the declarative memory whereas in model II (Figure 2, right) this information is derived from the production rules. This classification determines which arguments are going to be considered relevant in the *argue* task.

**Model I** The production rules activate fact and activate sentence contain the following structure:

=imaginal>	=imaginal>
fact =fact	sentence =sentence
==>	==>
+retrieval>	+retrieval>
word =fact	word =sentence

A chunk will be retrieved having a slot context containing either the chunk SUFFICIENT or NECESSARY. Figure 2, left, DECLARATIVE, gives two examples of such chunks (TEXT-SUF or TEXT-NEC). In the next step, this SUFFICIENT or NECESSARY chunk is placed in the imaginal buffer (Figure 2, left, IMAGINAL). This activation spreads to the chunk arguments (e.g. ARG-1 or ARG-2) which either contain the chunk SUFFICIENT or NECESSARY in their context slot.

**Model II** The read production rules in Figure 2, right, (e.g. read fact) all contain either the elements on the left or on the right:

=visual	>	=visual>	
value	•••	value	
==>		==>	
+retrie	val>	+retrieva	al>
value	NECESSARY	value	SUFFICIENT

where ... is a placeholder for a string value that is different for each production rule (e.g. "She will meet a friend"). After reading, the model interprets (or contextualizes) the sentence: Depending on which production rule matches and fires, a context chunk where either value NECESSARY or SUFFICIENT is retrieved and this retrieved chunk, either NEC or SUF (Figure 2, right, DECLARATIVE), is placed in the imaginal buffer. After that, the respective hypothesis chunk (either with value DISABLER or ALTERNATIVE) is retrieved and placed into the imaginal buffer.

#### **Argumentation Task**

The *argue* task can only start after the models have accomplished the *read* task (or at least once for model II).

Arguments as Chunks The chunks of type argument contains the slots fact, hypo and context which contain other chunks, respectively. Additionally, arguments contain the slots pos and neg-pos having string values, representing the position and the opposite position, and the slot str having a float value, denoting the argument's strength. Consider two

#### strong arguments from the example in the previous section:

```
(arg1 isa argument hypo NONE fact FRIEND
pos "YES" context SUF neg-pos "UNKNOWN" str 1)
(arg2 isa argument hypo DISABLER fact FRIEND
pos "UNKNOWN" context NEC neg-pos "YES" str 1)
```

arg1 represents the modus ponens argument, stating that *She will meet a friend* (fact FRIEND), together with the conditional being understood as sufficient (context SUF), being an argument for *She will go to play* (pos "YES"). arg2 represents the attacking argument including the final position: stating that, a disabling hypothesis (hypo DISABLER, e.g. *She does not have enough money*) and the conditional understood as necessary (context NEC), forms an argument for the position *She will not go to a play*. arg1 and arg2 are equally strong (str 1). As slot hypo in arg2 has a disabling hypothesis (DISABLER), it defends against arg1, and makes both arguments acceptable (thus we cannot conclude skeptically that *She will go to the play* and therefore the position is pos "UNKOWN).

Variations in Argumentation Process Humans differ in reasoning (c.f., (Khemlani & Johnson-Laird, 2016)): Some draw conclusions already based on one argument that supports a position, whereas others try to generate hypotheses to build (strong) counter arguments. The dialectic argumentation processes in model II (Figure 2, right) subsumes the one in model I and is as follows: In case an argument was successfully retrieved by search for argument, two production rules might apply, either (1) Respond with the position of that argument or (2) Search Counter argument. In the second case, three production rules might apply: (2a) there is a Retrieval Failure and the model Responds with the position of the current argument, (2b) there is a Retrieval Failure and the model Rereads the premises (which will increase either the activation of NEC or SUF) or (2c) Retrieval is Successful and both arguments are compared. The arguments can be compared in either one of the following ways: (2c,i) through their strengths (which argument is stronger?), (2c,ii) through their activation (which argument has the higher activation), or (2c,iii) based on their hypothesis (which argument has a disabling or alternative hypothesis?). Figure 2 only shows (2c,i), where depending on whether argument 1 or argument 2 is stronger, either one of the following production rules applies:

(p arg-1-	stronger	(p arg-2-	stronger
=goal>		=goal>	
state	argue	state	argue
=imagina	1>	=imagina	1>
strengt	h-1 =val	strengt	h-1 =val
< stren	gth-2 =val	> stren	gth-2 =val
arg-1	=pos	arg-2	=pos
==>		==>	
=imagina	1>	=imagina	1>
value	=pos	value	=pos
=goal>		=goal>	
state	respond )	state	respond )

When the argument taking the disabling or alternative hypothesis into account is chosen (2c,iii) then one of the following production rules applies:

(p arg-1-hypo =goal>	(p arg-2-hypo =goal>
state argue	state argue
=imaginal>	=imaginal>
- arg-1 nil	- arg-1 nil
- arg-2 nil	- arg-2 nil
arg-1 =pos	arg-2 =pos
- nypo-1 None	-nypo-2 None
==>	==>
=imaginal>	=imaginal>
value =pos	value =pos
=goal>	=goal>
state respond)	state respond)

In the current implementation, model II includes all options, except (2c,ii). Further, the utility to respond with the position of the firstly retrieved argument (thus not searching for a counter argument) is higher than for the other production rules.

#### **Evaluation**

We first show how the models perform with respect to a cognitive reasoning task and then discuss their results.<sup>1</sup>

#### **Application: Byrne's Suppression Task**

The application of Cognitive Argumentation in ACT-R is shown by means of a typical reasoning task. In the suppression task (Byrne, 1989) participants were asked whether they could derive conclusions given variations of a set of premises. The task consists of two parts, where in both parts, the conditionals are the same, but the factual information changes.

Part I Group I was given the following two premises: If she has an essay to finish, then she will study late in the library. and She has an essay to finish. (essay) The participants were asked what of the following answer possibilities follows assuming that the above premises were true: She will study late in the library, She will not study late in the library or She may or may not study late in the library. 96% of the participants in this group concluded that She will study late in the library (library). Group II of participants additionally received the following premise: If she has a textbook to finish, then she will study late in the library. which yield to the same result: 96% of the participants in this group concluded that She will study late in the library. Group III of participants instead additionally received the following premise: If the library is open then she will study late in the library. In this case, only 38% concluded that She will study late in the library. If instead She does not have an essay to finish was given as a fact, only 4% of Group II concluded She will not study late in the library, whereas for Group I and Group III, the percentage was 46% and 63%, respectively.

<sup>&</sup>lt;sup>1</sup>The models can be found here: https://github.com/eadietz/bst2actr.

**Part II** The second part of the experiment was similar, except that the given facts were different. The participants were given the fact that *She will study late in the library (library)* or *She will not study late in the library (not library)* and asked what of the following answer possibilities follows assuming that the given premises were true: *She has an essay to finish, She does not have an essay to finish* or *She may or may not have an essay to finish.* 

Fact	Group	Model I	Model II	Byrne	Dieussaert <sup>+</sup>
y	Ι	98	90	96	88
ssa	Π	98	90	96	93
<i>v</i>	III	52	37	38	60
	$\sim con$	cluded She	e will study l	ate in the	e library
		Model I	Model II	Byrne	Dieussaert+
say	Ι	47	31	46	49
t es:	II	5	10	4	22
iou	III	73	65	63	49
→ concluded She will not study late in the library					
		Model I	Model II	Byrne	Dieussaert+
2	Ι	46	31	71	53
bra	Π	4	10	13	16
<u>1</u>	III	72	64	54	55
→ concluded She has an essay to finish					
		Model I	Model II	Byrne	Dieussaert <sup>+</sup>
ury	Ι	95	90	92	69
ibra	Π	99	89	96	69
101 l	III	54	37	33	44
→ concluded She does not have an essay to finish					say to finish

Table 1: The percentages of model I and II after 100 simulations compared to the experimental results by Byrne (1989) and Dieussaert et al. (2000), abbreviated by Byrne and Dieussaert<sup>+</sup>, respectively. The first two columns are the cases and the groups. The highlighted rows show the suppression effects.

**Results** The results in Table 1 show that both, model I and model II account for the suppression effect in all four cases. The results that diverge most from the experimental data, are for cases II (*essay*) and III (*not essay*) for group I in model II. Model I fits better the data than model II, however which of the model's underlying mechanisms are more plausible?

# Discussion

Model I fits better the data than model II, but model II's implementation of background knowledge, divisions of tasks and individual differences, might better account for the underlying cognitive process. Through optimization via meta parameters or the utility modules, an eventual perfect fit of the models to the data seems feasible, however, maybe less interesting.

- **Background knowledge** In model I, background knowledge is stored in the declarative memory (where chunks differ in their base-level activation), whereas in model II, the knowledge is in the production rules.
- **Division of Tasks** Model I's tasks of *read*, *argue* and *respond* are strictly ordered. This might be plausible for the *respond* task, however the *read* and *argue* tasks seem intertwined, which makes model II more plausible: participants might re-read the sentences while they argue for or against some response.
- **Argument Selection** Chunks that are retrieved last have a higher activation than other chunks. Yet, for argumentative reasoning the strength or the attacking character (e.g. through disabling/ alternative hypotheses) might have stronger effects.
- **Individual Differences** Competing production rules in model II represent the different individual's responses. Another modeling approach could have been the implementation of a set of models.
- **Learning** Reasoning tasks usually do not consider learning, even though this is a relevant aspect for which cognitive architectures are well suited for.

## Conclusions

This paper shows how cognitive argumentation can be implemented into a cognitive architecture. In cognitive argumentation, cognitive principles specify task-independent assumptions humans might make while reasoning. A variation of the original dialectic argumentation process is formalized in ACT-R. Most importantly, an exhaustive search for arguments is avoided, and instead, the argumentation process is guided through chunk activation. Two argumentation-based reasoning models are evaluated to the experimental results of a famous reasoning task. The approach provides an ACT-R implementation of two models that solves a (conditional) reasoning task through cognitive principles where reasoning is a guided dialectic argumentation process. Still, a lot needs to be done to refine this approach. The current implementation takes the existence of arguments as granted and does not provide a mechanism of argument construction. Furthermore, we need to consider other reasoning tasks such as tasks that investigate learning. With the help of new experiments, we could evaluate and refine the dialectic argumentation process as currently implemented. Finally, the automation of the conditions' classification and the problem of prior knowledge is still an open problem. Natural language processing and argument mining (Lawrence & Reed, 2020) might be helpful for this purpose.

# References

- Anderson, J. R. (1990). *The adaptive character of thought*. Psychology Press.
- Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press.
- Anderson, J. R., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psy. Review*, 111(4), 1036-1050.
- Baroni, P., Gabbay, D., Giacomin, M., & van der Torre, L. (2018). *Handbook of formal argumentation*. College Publications.
- Braine, M. (1978). On the relation between the natural logic of reasoning and standard logic. *Psy. review*, 85(1), 1-21.
- Byrne, R. M. J. (1989). Suppressing valid inferences with conditionals. *J. of Memory and Language*, *31*, 61–83.
- Byrne, R. M. J. (2005). *The rational imagination: How people create alternatives to reality*. MIT press.
- Byrne, R. M. J., Espino, O., & Santamaría, C. (1999). Counterexamples and the suppression of inferences. *J. of Memory and Language*, 40(3), 347-373.
- Chater, N., & Oaksford, M. (1999). The probability heuristics model of syllogistic reasoning. *Cognitive Psy.*, *38*, 191-258.
- Dietz, E., & Kakas, A. (2021). Cognitive argumentation and the selection task. In *Proc. of the annual meeting of the Cognitive Science Society, 43* (pp. 1588–1594). Cognitive Science Society.
- Dietz, E., & Kakas, A. C. (2020). Cognitive argumentation and the suppression task. *CoRR*, *abs/2002.10149*.
- Dietz, E.-A., Hölldobler, S., & Ragni, M. (2012). A computational logic approach to the suppression task. In N. Miyake, D. Peebles, & R. P. Cooper (Eds.), *Proc. of the* 34th annual conference of the Cognitive Science Society (COGSCI) (pp. 1500–1505). Cognitive Science Society.
- Dietz Saldanha, E.-A., & Kakas, A. (2019). Cognitive argumentation for human syllogistic reasoning. *KI Künstliche Intelligenz*, *33*(3), 229–242.
- Dieussaert, K., Schaeken, W., Schroyens, W., & D'Ydewalle, G. (2000). Strategies during complex conditional inferences. *Thinking & Reasoning*, 6(2), 125–161.
- Ghosh, S., Meijering, B., & Verbrugge, R. (2014). Strategic reasoning: Building cognitive models from logical formulas. J. of Logic, Language and Information, 23(1), 1–29.
- Grice, H. P. (1975). Logic and conversation. In P. Cole & J. L. Morgan (Eds.), *Syntax and semantics* (Vol. 3). New York: Academic Press.
- Johnson-Laird, P. N., Girotto, V., & Legrenzi, P. (2004). Reasoning from inconsistency to consistency. *Psy. Review*, *111*(3), 640 - 661.
- Johnson-Laird, P. N. (1983). Mental models: towards a cognitive science of language, inference, and consciousness. Cambridge, MA: Harvard University Press.
- Johnson-Laird, P. N., & Byrne, R. M. J. (1991). *Deduction*. Hove, NJ: Lawrence Erlbaum Associates Ltd.

- Kelley, H. (1973). The processes of causal attribution. American Psychologist, 28(2), 107–128.
- Khemlani, S., & Johnson-Laird, P. N. (2016). How people differ in syllogistic reasoning. *38th Conference of the Cognitive Science Society*.
- Khemlani, S., & Trafton, J. G. (2012). mreactr: A computational theory of deductive reasoning. In N. Miyake, D. Peebles, & R. P. Cooper (Eds.), *Proc. of the 34th annual conference of the Cognitive Science Society* (pp. 581–586). Cognitive Science Society.
- Klein, G., Moon, B., & Hoffman, R. R. (2006). Making sense of sensemaking 2: A macrocognitive model. *IEEE Intelligent Systems*, 21(5), 88–92.
- Klein, G., Phillips, J. K., Rall, E. L., & Peluso, D. A. (2007). A data-frame theory of sensemaking. In R. R. Hoffman (Ed.), *Proc. of the sixth international conference on naturalistic decision making* (pp. 113–155). Lawrence Erlbaum Associates Publishers.
- Laird, J. E. (2012). *The soar cognitive architecture*. The MIT Press.
- Lawrence, J., & Reed, C. (2020, 01). Argument Mining: A Survey. Computational Linguistics, 45(4), 765-818.
- Lebiere, C., Pirolli, P., Thomson, R., Paik, J., Rutledge-Taylor, M., Staszewski, J., & Anderson, J. R. (2013). A functional model of sensemaking in a neurocognitive architecture. *Computational Intelligence and Neuroscience*, 2013, 921695:1–921695:29.
- Mercier, H., & Sperber, D. (2011). Why do humans reason? arguments for an argumentative theory. *Behavioral and Brain Sciences*, *34*(2), 57-74.
- Newell, A. (1990). *Unified theories of cognition*. USA: Harvard University Press.
- Peirce, C. S. (1903). Harvard Lectures on Pragmatism: Lecture VII. MS [R] 315.
- Polk, T. A., & Newell, A. (1995). Deduction as verbal reasoning. Psy. Review, 102, 533-566.
- Ragni, T., M. Fangmeier, & Brüssow, S. (2010). Deductive spatial reasoning: From neurological evidence to a cognitive model. In G. Salvucci D. D. und Gunzelmann (Ed.), *Proc. of the 10th international conference on cognitive modeling* (pp. 193–198). Philadelphia, PA. Drexel University.
- Rips, L. J. (1994). *The Psychology of Proof: Deductive Reasoning in Human Thinking*. The MIT Press.
- Salvucci, D. D. (2013). Integration and reuse in cognitive skill acquisition. *Cognitive Science*, 37(5), 829–860.
- Sloman, S. (1994). When explanations compete: the role of explanatory coherence on judgements of likelihood. *Cognition*, 52(1), 1 - 21.
- Stenning, K., & van Lambalgen, M. (2008). Human reasoning and cognitive science. In *Cambridge ma*. MIT Press.
- Taatgen, N. (2013, 06). The nature and transfer of cognitive skills. *Psy. review*, *120*, 439-471. doi: 10.1037/a0033138
- Thomson, R., Lebiere, C., Anderson, J. R., & Staszewski, J. (2015). A general instance-based learning framework

for studying intuitive decision-making in a cognitive architecture. Applied Research in Memory and Cognition, 4(3), 180-190. (Modeling and Aiding Intuition in Organizational Decision Making)

- Wason, P. (1964). The effect of self-contradiction on fallacious reasoning. *Quarterly J. of Exp. Psy.*, 16(1), 30-34.
- Wason, P. (1968). Reasoning about a rule. *Quarterly J. of Exp. Psy.*, 20(3), 273–281.
- Weick, K. (1995). *Sensemaking in organizations*. SAGE Publications.

# Modeling of Multi-Defender Collaboration in a Cyber-Security Scenario

Yinuo Du (yinuod@andrew.cmu.edu) Carnegie Mellon University Palvi Aggarwal (palvia@utep.edu) The University of Texas at El Paso Kuldeep Singh (palvia@utep.edu) The University of Texas at El Paso Cleotilde Gonzalez (palvia@utep.edu) Carnegie Mellon University

#### Abstract

While evidence shows that cyber attackers are good at coordinating and collaborating in their attacks, network defenders are notoriously poor at sharing information and collaborating among themselves. To help promote cooperation among defenders, one requires models that can explain and make predictions of emergent cooperation decisions of each defender in a cyber security scenario. We propose a Multi-Agent Instance-Based Learning (MDIBL-PD) cognitive model based on Instance-based Learning (IBL) theory, and founded on the Prisoner's Dilemma (PD) of cooperation. MDIBL-PD aims at explaining how collaborations emerge to share information with other defenders in a group. MDIBL-PD was created to interact in a Multi-Defender-Game (MDG) that was used in an experimental study with human participants, intended to determine the effect of different levels of information sharing on collaboration. MDIBL-PD uses an extension of the utility function in IBL theory to capture the emergence of cooperation with higher levels of social information. Through simulations with MDIBL-PD we collect synthetic data to compare to the data set collected in human studies. Our results help explain the emergence of cooperation at increasing levels of information regarding others' actions. We demonstrate the ability of MDIBL-PD to predict human cooperation decisions in the MDG in situations in which players have only their own information and in situations in which they have information about the sharing behavior of the other players.

**Keywords:** Cognitive Modeling; Multi-agent; Cooperation; Prisoners' dilemma; Cyber-Security

#### Introduction

In cybersecurity a major problem is the collaboration and coordination among defenders to share information on their vulnerabilities and experienced attacks. Sharing this information brings a major concern for companies and organizations: their privacy and competitive advantage can be damaged if other ill-intentioned people can take advantage of such information for their own benefit. In other words, organizations experience a social dilemma, in which there is a benefit to sharing information, but also put privacy at risk.

Singh, Aggarwal, and Gonzalez (2021) studied this social dilemma in cybersecurity using a Multi-Defender-Game (MDG) in human experiments, to learn about the conditions under which humans share information. MDG is a dynamic game in which sharing information may influence their future security and attack probability. Their experimental results

demonstrated a decreasing trend of the average proportion of group-level sharing. Human participants also tended to share less after being attacked, suggesting that instead of making sharing decisions solely based on reciprocity to their groupmates, participants may also base their decisions on the breach status, and might erroneously attribute the breach loss to groupmates.

As suggested by the Hierarchy of Social Information (HSI) in Gonzalez and Martin (2011), an increase in cooperation can be promoted by additional levels of information regarding the other players' actions and outcomes. Thus, knowledge about others' actions and outcomes might make the associations of reciprocity more clear and direct. The similarity of other's predicament to one's own can help strengthen a sense of reciprocity and thus lead to greater cooperation. The HSI proposed an increased level of social information from having no information about the others to an increased level of descriptive social information, where increased information about the complete interaction structure may result in more effective promotion of cooperation. Gonzalez and Martin (2011) argued that ongoing visibility of the payoff matrix can assist in clarifying the trade-off between short-term and long-term rewards. The cognitive modeling work in (Gonzalez, Ben-Asher, Martin, & Dutt, 2015) also suggests that humans tend to consider the outcome of their opponent, dynamically weighted by their interaction experience.

In cognitive science, most models focus on the individual behaviors. Many models aim at studying the cognitive processes of the attacker in order to inform the defense strategies (e.g., masking Aggarwal, Thakoor, et al., 2022; signaling Cranford et al., 2021; anti-phishing Singh, Aggarwal, Rajivan, & Gonzalez, 2020). Other models describe the recognition and comprehension processes of an individual defender (Dutt, Ahn, & Gonzalez, 2011) or the interaction between attacker and defender (Aggarwal, Moisan, Gonzalez, & Dutt, 2022). However, there's a lack of cognitive modeling for groups of defenders in the context of cybersecurity.

Mermoud, Keupp, Huguenin, Palmié, and Percia David

(2019) proposed a behavioral framework that theorizes the association between human behavior and their frequency and intensity to participate in security information sharing. However, their analysis focused on the individuals rather than the interaction among them. A recent review by (Ask, Lugo, Knox, & Sütterlin, 2021) suggests that research on cyber threat communication are mostly interview-based exploratory studies and focused more on individual-organization interaction and internal collaboration (Ahrend, Jirotka, & Jones, 2016; Hámornik & Krasznay, 2017).

In describe what follows. we first the Multi-Defender-Game (MDG) paradigm that reveals the dynamics of defenders' sharing tendency in groups of three over the course of 50 trials. We then formalize a cognitive model of a defender, built in SpeedyIBL (Gonzalez, Lerch, & Lebiere, 2003; Nguyen, Phan, & Gonzalez, 2021). Using the data set collected in a human experiment, we demonstrate that cognitive models of defenders can be useful for understanding the factors affecting the continuation and break down of collaboration and how humans account for the outcome of others.

#### **Multi-Defender-Game (MDG)**

We have developed a Multi-Defender-Game (MDG) for data collection through human experiments. The MDG is designed for group experiments. In the MDG there is a group of defenders (human participants) that play an information sharing game in a cyber-security scenario. The participants are assigned in groups of three players, in which they will be identified as defenders Defender 1, Defender 2, and Defender 3, each of them defending their own network. Initially, each defender receive 1000 points as an endowment, which can be used to invest in security to defend their network. Each defender's network is independent, some defenders may be attacked when the others are not and each may have a different chance of being attacked. Then defenders start the game and play 50 trials of decision making on sharing/not sharing information with other defender in the network. The goal of each defender is to maximize their points in the game.

In each trial t, the defender's network may or may not get attacked determined by his *Probability of Breach Pb<sup>t</sup>*. If the defenders' network gets attacked, then it costs them -30 points (*attack status*  $C_a^t = 1$ ). They need to choose to share or not to share information with other defenders in the network about the attack/not attack. They will then receive feedback information after the other two group members make their decisions.

The cost of information sharing (-15 points) is deducted from the available points if defenders choose to share information with others. The defender (receiver) gets rewarded (35 points) for receiving information from each other defender. Collectively, the sharing interaction between two defenders forms a prisoner's dilemma (table.1). For example, the payoff in the share-share cell is 20 = 35 - 15 for both the column player and the row player. Sharing points  $Z_i^t$  of defender *i* at trial *t* is the sum of receiving reward and sharing cost with the other two defenders in their group. The accumulated reward of player *i* at trial *t* of defender *i* is given by Eq.1. We assume the information shared is valuable and it helps the receiver to strengthen their security, thus information sharing also affects future probability of breach by Eq.2.

$$R_i^t = R_i^{t-1} + Z_i^t + (-30) \cdot C_a^t \tag{1}$$

$$Pb^{t+1} = Pb^t - \frac{0.95 \cdot Z_i^t}{2000} \tag{2}$$

Table 1: Payoff matrix

Defender 1 or Defender 2				
	Share Not-Share			
Defender 3	Share	20,20	-15,35	
	Not-Share	35,-15	0,0	

#### Human Dataset

As a baseline to compare the predictions of our IBL model, we used a data set collected from human participants who played together in groups using the MDG. This study recruited a total of 210 participants (about 46% female) from Amazon Mechanical Turk, to play a game in groups of 3 participants). On successful completion of the experiment, all participants received a base payment of \$3 and they could earn up to \$1.75 as additional bonuses based on the points available at the end. The average time taken to complete the experiment was 15 minutes.

The data set consists of two experimental conditions defined based on the information given to the participants regarding the sharing information of the other defenders in their group. The information levels were: *Own* and *Others*, where the *Own* condition provided only information on the actions of the other defenders in the group; while the *Others* condition also provided the outcomes of others and their breach status. Participants received this information in table 2 where the sharing decisions of each defender in the group, including the protagonist defender, were displayed in a separate column. The table also included their breach status, when this information was shared by the other defenders in the group. A total of 102 participants (34 groups) were in the *Own* condition, and 108 participants (36 groups) in the *Others* condition.

# Instance-Based Learning Model of Defender's Collaborations

We propose an Instance-Based Learning (IBL) cognitive model to make predictions about human sharing behavior in the MDG, at different levels of information. The model, Multi-Defender IBL - Prisoner's dilemma (MDIBL-PD),

Defender 3 Decision (Me)	Defender 1 Decision	Defender 2 Decision
Information not shared with Defender 1,	Defender 1 shared information with me,	Defender 2 didn't share any information
Information shared with Defender 2	He was attacked	Defender 2 didn't share any information
	My Payoff with Defender 1 : 35	My Payoff with Defender 2 : 0
	Defender 1's Payoff with me : -15	Defender 2's Payoff with me : 0

Table 2: An example output table provided as feedback in the Others condition of the (Du et al., n.d.) experiment

is based on a model of individual learning and decisions from experience in repeated two-player prisoner's dilemma (Gonzalez et al., 2015), and expands that concept to a multiplayer situation beyond a dyad. Like all IBL models, the MDIBL-PD model relies on the IBL Theory (i.e., IBLT) (Gonzalez et al., 2003), a well-known cognitive theory of experiential decision making. The key idea of this theory is that decisions are made by recognition of similar past experiences, their integration into the generation of expected utility of decision alternatives, and the selection of the alternative with the maximal expected utility. An IBL model can accurately represent the content of human memory, recognition, learning, and recall of experiences in decision making.

The IBLT process and mechanisms are general to every IBL model. These have been published in the past, but we repeat the mathematical formulations of the theory here for completeness. In IBLT, an "instance" is a memory unit that results from the potential alternatives evaluated. These memory representations consist of three elements that are constructed over time: a situation state s that is composed of a set of characteristics f; a decision or action a taken corresponding to an alternative in state s; and an expected utility or experienced outcome x of the action taken in a state. Concretely, for an IBL agent, an option k = (s, a) is defined by the action a in the state s. At time t, assume that there are  $n_{kt}$  different instances  $(k_i, x_{ik,t})$  for  $i = 1, ..., n_{kt}$ , associated with k. Each instance i in memory has an activation value, which represents how readily available this information is in memory (Anderson & Lebiere, 1998). Here, the equation captures recency, frequency, similarity, and noise in memory.

$$\Lambda_{ik_{i}t} = \ln\left(\sum_{t'\in T_{ik_{i}t}} (t-t')^{-d}\right) + \alpha \sum_{j} Sim_{j}(f_{j}^{k}, f_{j}^{k_{i}}) + \sigma \ln\frac{1-\xi_{ik_{i}t}}{\xi_{ik_{i}t}},$$
(3)

where d,  $\alpha$  and  $\sigma$  are the decay, mismatch penalty, and noise parameters, respectively, and  $T_{ik_it} \subset \{0, ..., t-1\}$  is the set of the previous timestamps in which the instance i was observed,  $f_j^k$  is the *j*-th attribute of the state *s*, and  $Sim_j$ is a similarity function associated with the *j*-th attribute. The rightmost term represents noise to capture individual variation in activation, and  $\xi_{ik_it}$  is a random number drawn from a uniform distribution U(0, 1) at each step and for each instance and option.

The activation of an instance i is used to determine the probability of retrieving an instance from memory. The probability of an instance i is defined by a soft-max function:

$$P_{ik_{i}t} = \frac{e^{\Lambda_{ik_{i}t}/\tau}}{\sum_{i=1}^{n_{kt}} e^{\Lambda_{jk_{j}t}/\tau}},$$
(4)

where  $\tau$  is the Boltzmann constant (i.e., the "temperature") in the Boltzmann distribution. For simplicity,  $\tau$  is often defined as a function of the same  $\sigma$  used in the activation equation  $\tau = \sigma \sqrt{2}$ .

The expected utility of option k is calculated based on *Blending* as specified in the choice tasks:

$$V_{kt} = \sum_{i=1}^{n_{kt}} P_{ik_i t} x_{ik_i t}.$$
 (5)

The choice rule is to select the option with the maximum blended value.

#### **MDIBL-PD model of Information Sharing**

The IBL model of the individual defender is primarily concerned with the learning processes determined by the various levels of information available to the model. We denote the within-group defender index as  $x \in \{1,2,3\}$  and their sharing decisions as  $D_x \in \{C(\text{Share}), D(\text{Not-Share})\}$ .

The new MDIBL-PD model was developed for both the own and others information conditions described above. Each IBL agent in the MDIBL-PD model makes decisions using the same procedure defined in the previous section. The human participants in the condition *Others* receive information on the outcome and the breach status of other players (Table 2). To capture this interdependence, we modified the blending equation (Eq.5) to account for the outcome of the other player, as suggested in (Gonzalez et al., 2015).

Actions *a*: In the MDG, the choice options are defined by the actions that each defender can take. The defender  $D_x$  can choose not to share information, to share information with one or both of the other defenders, denoted as None,  $D_{(x+1) \mod 3}$ ,  $D_{(x+2) \mod 3}$ , *Both*.

**State**  $S_i^t$ : The situation state of the defender consists of four attributes: the breach status  $A_x \in \{1(\text{attacked}), 0(\text{safe})\}$ , probability of breach (Pb<sub>i</sub><sup>t</sup>), and the expectation of receiving information from each player  $(E_{D_x}^t)$ . Thus, the situation state *s* of participant *i* (Defender *x*) at trial *t* is  $s_i^t = (A_i^t, Pb_i^t, E_{D_{(x+1) \mod 3}}^t, E_{D_{(x+2) \mod 3}}^t)$ .

Breach status  $A_i^t$  and probability of breach  $Pb_i^t$  have direct and indirect affect on the outcome of a trial, thus are included as the context information whose pure appearance might

affect human's information sharing tendency. As suggested by (Zhang, Lin, Jing, Feng, & Gu, 2019), beliefs and behavior correlate within rounds in repeated prisoners' dilemma game, and beliefs in one round vary with behavior in the previous round. Thus, we include  $E_{D_x}^t$  to capture the association between the expectation of receiving information from peers and the decision of whether to share information with them. It is approximated with the accumulated proportion of receiving information from  $D_x$  (Eq.6). Here, we assume that participants can keep track of the interaction experience with their peers. This assumption can be relaxed by manipulating the window of proportion calculation. After receiving the actual sharing decisions at the trial t, the  $E_{D_x}^t$  slots will be updated to  $T_{D_x}^t$  to store the real interaction experience in memory. When the expectation  $E_{D_x}^t$  is closer to 1, memory instances of receiving information from peer x  $(T_{D_x}^{t'})$  $1, t' \in [0, t)$  have greater similarities to the current situation, resulting in higher activation values (3), and higher likelihood to be recovered (4). Similarly, when the expectation  $E_{D_r}^t$  is closer to 0, memories of defected by peer x ( $T_{D_x}^{t'} = 0, t' \in$ [0,t)) are more likely to be retrieved. The similarity of these numeric attributes is calculated linearly and normalized to  $\nabla t - 1 \pi i$ [0,1].

$$E_{D_x}^t = \frac{\sum_{i=0}^t T_{D_x}^a}{t-1}$$
(6)

**Utility**  $U_x^t$ : Depending on the experimental condition, the players in the MDG received only information on their own actions (Own) or about the sharing decisions of other defenders and the effect on their outcome of themselves (Others). Therefore, the utility of the defender *x* in the trial *t* is the points gained or lost exclusively at that trial, constituted with the benefit of receiving information (35 points), the cost of sharing information (-15 points) and the cost of being attacked (Eq.7). The cost-benefit of information sharing forms the dyadic prisoner's dilemma as shown in Table 1. The cost of the breach is included as part of the utility, since the status of the breach has an effect on the sharing decisions of human defenders.

$$U_x^t = \Delta_x^t = Z_x^t + (-30) \cdot A_x^t \tag{7}$$

$$U_{x}^{t} = \Delta_{x}^{t} + w_{1}^{t} \cdot \Delta_{(x+1) \mod 3}^{t} + w_{2}^{t} \cdot \Delta_{(x+2) \mod 3}^{t}$$
(8)

$$w_1^t = \frac{1 - Surprise_1^t}{2} \tag{9}$$

$$w_2^t = \frac{1 - Surprise_2^t}{2} \tag{10}$$

To simulate how humans account for the outcome of others, the utility for the blended value calculations is set as the weighted sum of the point update of the defender  $D_x$  and his peers (Eq.8). Inspired by the notion of *Social value orientation (SVO)* (Balliet, Parks, & Joireman, 2009), *w* represents the degree to which a player is willing to consider the outcome of the other player for each option when making a decision that maximizes the gains in each trial.

Research in (Gonzalez et al., 2015) finds that the dynamic w dependent on individual experiences can best explain

human cooperation behavior. Under this hypothesis, a player will account for the outcome of the opponent as a function of a normalized gap between expected and actual outcomes (surprise). The value of  $w_i^t$  (with respect to the opponent's outcome in the trial t) will be reduced by surprise (Eq.9 and Eq.10). We assume that the players evaluate the benefit of sharing information with each other independently with different weights, updated according to separate *surprises* and *gaps*.

The normalization of *surprises* limited the value of *Surprise*<sup>t</sup><sub>i</sub> within the range of [0,1], the value of  $w_i^t$  within [0,0.5], and the sum of weights on the benefit of *others* within [0,1]. This formulation assumes that the way a player accounts for the opponent's outcomes will vary between *extreme selfish* when  $w_1^t = w_2^t = 0$  and extreme fairness when  $w_1^t = w_2^t = 0.5$ .

$$Surprise_{i}^{t} = \frac{Gap_{i}^{t}}{[Mean(Gap_{i}^{t}) + Gap_{i}^{t}]}$$
(11)

$$Gap_{i}^{t} = Abs(V_{j}^{t-1} - (X_{ij} + O_{ij}))$$
(12)

$$Mean(Gap_i^t) = Mean(Gap_i^{t-1})(1 - \frac{1}{50}) + Gap_i^t(\frac{1}{50}) \quad (13)$$

Pre-Population: From human data, we observed that more than 70% of the human participants chose to share information with both peers at the beginning. (Andreoni & Miller, 1993) show that some fraction of the population actually has altruistic motives. This ingrained tendency to share between human subjects can be the consequence of the experience of cooperation in recent years, or it could be an experimental effect of human participants who expected to cooperate in a Multi-Defender Collaboration Game. To capture this preference, and inspired by the conclusion in (Kelley & Stahelski, 1970) that there are two stable types of individuals that can be described as cooperative and competitive, we prepopulate the IBL agents with instances that represent these initial tendencies. 70% of IBL agents are prepopulated with Share instances with positive rewards (0, 20, 40 for zero, one, two sharing - receiving with peers), while 30% of IBL agents are prepopulated with Not-share instances with negative rewards (0, -15, -30) for zero, one, two sharing - not receiving with peers). Cooperatively biased agents and defectively biased agents are randomly formed groups of three. Each group contains random number (0 to 3) of cooperatively biased agents. The assumption is that the decrease in the proportion of information sharing is caused by the pairing of *cooperative* participants with defective participants.

**Simulation Procedure:** The MDIBL-PD model with default parameters was run for 100 simulated groups of players in each of the two information conditions. Each group plays the game for 50 trials. The utility assignment for *Own* condition follows Eq.7. The utility for *Others* condition follows Eq.8 with  $w_1, w_2$  defined by Eq.9 and Eq.10.

**Dependent Measures:** We calculate the overall proportion of sharing in Own and Others conditions, the proportion of sharing with Both, One, or None of the other defenders, and the sequential dependencies that emerged from the interaction between IBL agents in a group (Martin, Gonzalez, Juvina, & Lebiere, 2014). Sequential dependencies measures include: *Mistrust*, the decision a player makes to defect at time t, after both players mutually defected at time t-1; Forgiveness (Not Share - Share), the decision to continue cooperating at time t, although mutual cooperation was not achieved due to the defection of the other at time t - 1; Abuse (Share - Not Share), the decision to continue defecting at time t after a profitable defection at t - 1; and *Trust*, the decision to continue cooperating at time t, after successful mutual cooperation at time t - 1. To assess the precision of the predictions of the model with respect to human data, we calculated the mean squared deviation (MSD) using the average of the dependent measure (e.g., the average proportion of cooperation per trial) and using the Pearson correlation coefficient (r) to assess the similarity of time trends between the model and human data.

#### Results

#### **Overall Information Sharing**

Figure 1 illustrates the proportion of sharing for the MDIBL-PD model compared to human data in the conditions *Own* and *Others* conditions over the course of 50 trials. The proportion of sharing in human data is higher in the *Others* condition (Mean=0.74, SD=0.44) than in the *Own* condition(Mean=0.59, SD=0.49). As shown in Fig.1, the MDIBL-PD model captures these observed trends very accurately. The MSD between human data and model data in *Own* condition is 0.0029, with r = .86, p < 0.001. The MSD in Others condition is 0.0022, with r = .76, p < 0.001.



Figure 1: Overtime Sharing Proportion for the *Own* condition (left panel) and the *Others* condition (right panel)

## Proportion of sharing with None, One or Both

Figure 2-Top panel, represents the proportion of information sharing with both one and none of the other players in the Own condition. More than 70% human participants choose to share with *Both* peers at the beginning. The proportion decreases over time, and some participants shift to sharing

with *One* of the peers, and more participants choose to share with *None*. Most importantly, in the *Own* condition, where participants only receive feedback about their own actions and outcomes, the proportion of sharing with none of the other players increases over the 50 trials.

The model is able to approximate the trends of three types of options accurately. As shown in Fig.2, the deviation between human and model in the proportion of sharing with *Both, One,* and *None* is trivial, especially for the *None* option with (MSD = 0.0029, r = .86, p < 0.001). We note that the model seems to show a stronger preference for sharing with *One*, while human participants share more with *Both*. A possible explanation is that a fraction of human participants are altruistic or are trying to build an altruistic reputation by indiscriminately sharing with *Both*. The model's decisions, driven by the utility exclusively, converge relatively quickly to the more rewarding options, i.e., sharing with the more reciprocal peer.



Figure 2: Sharing proportions with Both, One, or None of the other players for the *Own* condition (top panel) and the *Others* condition (bottom panel)

Figure 2-Bottom panel, represents the proportion of information sharing with both, one and none of the other

players in the Others condition. The model can account for the dynamics of choosing three types of option (*Both*: MSD =0.0.0041, r = .86, P < 0.001, *One*: MSD = 0.0069, r =.58, p < 0.001, *None*: MSD = 0.0049, r = .64, p < 0.001). Similar to *Own* condition, human participants demonstrate an initial preference to share with *Both* other players. Although still increasing, the upward trend of sharing with *None* is more flat, indicating that the information of the actions and results of others is effective in maintaining cooperation.

#### **Sequential Dependencies**

Fig.3-Left panels demonstrate the comparison between human and model in terms of sequential dependency metrics in *Own* condition. The model fits *Mistrust*, *Trust*, and *Forgiveness* reasonably well with a significant positive correlation with human data (*Trust*: MSD = 0.0203, r =.55, P < 0.001, *Mistrust*: MSD = 0.0052, r = .92, p < 0.001, *Forgiveness*: MSD = 0.0207, r = .83, p < 0.001), but exhibits approximately 25% more *Abuse* than human players (MSD =0.0708, r = .16, p > 0.05).

Similarly, Fig.3-Right panels show that the model matches human behavior for the *Others* condition in terms of *Mistrust* (MSD = 0.0158, r = .85, p < 0.001) and *Forgiveness* (MSD = 0.0404, r = .80, p < 0.001), but deviates on *Trust* (MSD = 0.0291, r = .12, p > 0.05) and *Abuse* (MSD = 0.0583, r = .36, p > 0.05). The model is still more likely to *Abuse* and *Forgive* than humans. *Defect* is getting increasingly rewarding as the game progresses, and it becomes more affordable to lose a cooperator.

#### Discussion

In this paper, we propose a cognitive model that represents the dynamics of cooperation among defenders in a multi-defender game. The MDIBL-PD model builds on and advances the model proposed in (Gonzalez et al., 2015) for a dyad playing the PD game. The model proposes that direct information on the actions of others, whether they share or not with the own player, will influence the emergence of cooperation in the group. The outcomes of the other players in the group are used by each player to make their own decisions. However, the outcomes of the other players are only considered to a certain extent (i.e., "w"). The main insight from (Gonzalez et al., 2015) is that such "w" is dynamic and depends greatly on how the other players behave with the own player in each round of the game. That is, the regard that the self gives to others depends on the dynamic behaviors of others. This idea was used in the MDIBL-PD model and simulation results were produced to replicate the conditions of an experiment carried out with human data.

The results demonstrate that the model performed similarly to the actions taken by humans. First, with more information on Others, individuals share information more often in the MDG. Second, humans tend to decrease the proportion of sharing with both players and increase the proportion of their no-sharing behavior over time. This happens particularly in the Own condition. There are also some differences between



Figure 3: Sequential dependencies in the *Own* condition (top panel) and the *Others* condition (bottom panel), showing Trust, Mistrust, Abuse, and Forgiveness behaviors of the model and human participants

the model's predictions and human data. For example, in the Own condition, the model initially tends to share more with one of the other players. The model also shows a higher proportion of "abuse" of the other players, defined as the proportion of defections (not sharing) the model makes after the other player has cooperated (shared). It seems that the model is more "selfish" than humans are regardless of the level of information, as clearly the level of abuse in the model is higher than that of human participants.

Sequential dependencies also indicate that humans have difficulty sharing information with other players, increasing the level of mistrust of other players over time. This pattern is particularly strong in the Own condition, and the model replicates such trends.

Future research will explore more of how to account for others' decisions while making decisions, for example the surprise and *w* values to explain human behavior. We will also look at the triads in more detail and see the proportion of sharing with each of the two other players.

## Acknowledgements

This research was sponsored by the U.S. Army Combat Capabilities Development Command Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Combat Capabilities Development Command Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

#### References

- Aggarwal, P., Moisan, F., Gonzalez, C., & Dutt, V. (2022). Learning about the effects of alert uncertainty in attack and defend decisions via cognitive modeling. *Human Factors*, 64(2), 343–358.
- Aggarwal, P., Thakoor, O., Jabbari, S., Cranford, E. A., Lebiere, C., Tambe, M., & Gonzalez, C. (2022). Designing effective masking strategies for cyberdefense through human experimentation and cognitive models. *Computers & Security*, 117, 102671.
- Ahrend, J. M., Jirotka, M., & Jones, K. (2016). On the collaborative practices of cyber threat intelligence analysts to develop and utilize tacit threat and defence knowledge. In 2016 international conference on cyber situational awareness, data analytics and assessment (cybersa) (pp. 1–10).
- Anderson, J. R., & Lebiere, C. J. (1998). The atomic components of thought (J. R. Anderson & C. J. Lebiere, Eds.). New York: Psychology Press. doi: https://doi.org/10.4324/9781315805696
- Andreoni, J., & Miller, J. H. (1993). Rational cooperation in the finitely repeated prisoner's dilemma: Experimental evidence. *The economic journal*, 103(418), 570–585.
- Ask, T. F., Lugo, R. G., Knox, B. J., & Sütterlin, S. (2021). Human-human communication in cyber threat situations: A systematic review. In *International conference on human-computer interaction* (pp. 21–43).
- Balliet, D., Parks, C., & Joireman, J. (2009). Social value orientation and cooperation in social dilemmas: A meta-analysis. *Group Processes & Intergroup Relations*, 12(4), 533–547.
- 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.
- Du, Y., Aggarwal, P., Kuldeep, S., & Gonzalez, C. (n.d.). Multi-defender collaborations in a cyber-security scenario. *Human Factors*.

- Dutt, V., Ahn, Y.-S., & Gonzalez, C. (2011). Cyber situation awareness: Modeling the security analyst in a cyber-attack scenario through instance-based learning. In *Ifip annual conference on data and applications* security and privacy (pp. 280–292).
- Gonzalez, C., Ben-Asher, N., Martin, J. M., & Dutt, V. (2015). A cognitive model of dynamic cooperation with varied interdependency information. *Cognitive science*, *39*(3), 457–495.
- Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science*, 27(4), 591–635.
- Gonzalez, C., & Martin, J. M. (2011). Scaling up instance-based learning theory to account for social interactions. *Negotiation and Conflict Management Research*, 4(2), 110–128.
- Hámornik, B. P., & Krasznay, C. (2017). A team-level perspective of human factors in cyber security: security operations centers. In *International conference on applied human factors and ergonomics* (pp. 224–236).
- Kelley, H. H., & Stahelski, A. J. (1970). Social interaction basis of cooperators' and competitors' beliefs about others. *Journal of personality and social psychology*, *16*(1), 66.
- Martin, J. M., Gonzalez, C., Juvina, I., & Lebiere, C. (2014). A description–experience gap in social interactions: Information about interdependence and its effects on cooperation. *Journal of Behavioral Decision Making*, 27(4), 349–362.
- Mermoud, A., Keupp, M. M., Huguenin, K., Palmié, M., & Percia David, D. (2019). To share or not to share: a behavioral perspective on human participation in security information sharing. *Journal of Cybersecurity*, 5(1), tyz006.
- Nguyen, T. N., Phan, D. N., & Gonzalez, C. (2021). Speedyibl: A solution to the curse of exponential growth in instance-based learning models of decisions from experience. *CoRR*, *abs/2111.10268*. Retrieved from https://arxiv.org/abs/2111.10268
- Singh, K., Aggarwal, P., & Gonzalez, C. (2021, 06). A social dilemma for cybersecurity: Sharing information among defenders..
- Singh, K., Aggarwal, P., Rajivan, P., & Gonzalez, C. (2020). What makes phishing emails hard for humans to detect? In *Proceedings of the human factors* and ergonomics society annual meeting (Vol. 64, pp. 431–435).
- Zhang, D., Lin, Y., Jing, Y., Feng, C., & Gu, R. (2019). The dynamics of belief updating in human cooperation: findings from inter-brain erp hyperscanning. *NeuroImage*, *198*, 1–12.

# Combining EEG and a Cognitive Model to Infer the Time Course of Game Play

Jon M. Fincham (fincham@cmu.edu)

Department of Psychology, 5000 Forbes Ave Pittsburgh, PA 15213 USA

Caitlin S. Tenison (ctenison@ets.org)

Educational Testing Service Princeton, NJ 08541 USA

## John R Anderson (ja@cmu.edu)

Department of Psychology, 5000 Forbes Ave Pittsburgh, PA 15213 USA

#### Abstract

We have developed an analysis stream for integrating a cognitive model with EEG data to reconstruct the cognition of individual subjects. A critical component of this method is the Sketch level that combines cognitive modeling and classification of EEG data using an HSMM to identify and place critical events over the timeline of a task. Multiple factors can influence sketch accuracy. In this study, we investigated the effect of game play elements on sketch accuracy across two EEG experiments where subjects interacted with the Space Fortress video game. Experiment 1 consisted of elaborate interface elements that accompanied game events (multiple sound effects, visual explosions). Subjects in Experiment 2 performed the same task, but audio and visual feedback elements were greatly reduced. We find that sketch accuracy while still much better than chance in Experiment 2, was significantly worse than in Experiment 1.

**Keywords:** EEG, cognitive modeling, cognitive reconstruction, HSMM, MVPA, Space Fortress, video game, BCI

#### Introduction

Considerable studied research has classifying electroencephalography (EEG) signals and the results have been applied to a number of domains such as braincomputer interfaces (BCI; Lotte et al., 2018), emotion recognition (Kim et al., 2013), understanding human memory (Noh et al., 2014), estimating workload (Brouwer et al., 2012), among others. Much of this research is conducted using a limited set of interaction paradigms (Abiri et al., 2019; Saeidi et al., 2021). In active BCI systems, classification methods are used to identify specific brain signals consciously and purposefully generated by the participant. Reactive BCI systems involves tasks where the experimenter has control over the presentation of stimuli and examines activity in predefined intervals, typically locked to the presentation of these stimuli. Research on passive BCI focuses on the classification of brain states that occur within complex, operational environments such as driving or aviation. Within passive BCI systems the sequence of events emerges as an interaction between the subject and the environment. These events can reflect a complex interplay between the cognitive process and task context and the uncertain timing of these events adds an additional challenge to their detection. Although this research is often conducted within realistic situations, the focus of the detection is often limited to considering only a few, highly distinguishable cognitive states(Aricò et al., 2016). The ability to decode diverse, time-variable events has valuable implications for enabling the development of neuroadaptive technologies to support complex tasks and greater interactivity (Krol et al., 2018)

Video games can provide a rich testbed that begin to bridge the gap between doing traditional EEG experiments in tightly controlled lab studies and the complex tasks in which people routinely engage every day. In recent research, Anderson, et al, (2020) decoded cognitive, perceptual, and motor events from EEG data gathered from participants playing the video game Space Fortress (Donchin, 1989; Frederiksen & White, 1989; Gopher et al., 1989). In that work, they presented a Sketch and Stitch method that was successful in reconstructing an entire sequence of actions to capture the play of a subject in a game. The Sketch component of that procedure was used to infer a chronology of the critical events of a subject's gameplay by using a hidden semi-Markov model (HSMM) to combine cognitive modeling and EEG data. The critical events they tried to identify were

- 1. Kills: when a player succeeds in destroying fortress;
- 2. Deaths: when a player's ship is destroyed;
- **3. Resets:** when a player slips in trying to build up the vulnerability of the fortress and is reset to 0.

They exploited the fact that such events during gameplay tend to produce robust EEG signals while a cognitive model can provide probabilities of various transitions between critical events as well as the distribution of intervals between these events. The approach identifies the most probable sequence of critical events and when they happened.

While Anderson et al (2020) had success identifying critical events in a subject's game play, the Space Fortress interface accompanies these critical events by special visual

and auditory effects, raising the question if this success just depended on detecting perceptual responses in the EEG. For example, the destruction of a ship was accompanied by a sound effect and an elaborate visual element meant to indicate an explosion. In this paper, we explore the question of how well the method would work in a situation where these events occurred without the strong perceptual correlates. We ran an experiment that replicated the one described in Anderson et al (2020) but reduced the audio and visual events that accompanied game play. Necessarily, something in the interface must change to indicate to the subject that the event has happened, but we eliminated strong visual and auditory signals. We will compare the results with this reduced interface to the prior results with the original Space Fortress interface.

#### **Space Fortress Game**

Figure 1 illustrates the critical elements of the game. Players are instructed to fly a ship between the two hexagons. They are firing missiles at a fortress in the middle, while trying to avoid being hit by shells fired by the fortress. The ship flies in a frictionless space. To navigate, the player must combine thrusts in various directions to achieve a path around the fortress. Mastering navigation in the Space Fortress environment is challenging; while subjects are overwhelmingly video game players, most have no experience in navigating in a frictionless environment. We use the Pygame implementation of Space Fortress (Destefano, 2010)where all actions are key presses.



Figure 1: Snapshot of ship (nearest outer hexagon) shooting missile (arrow) at fortress (inside inner hexagon).

We used the Autoturn version of the game introduced in Anderson et al. (2019) and described in detail in that paper. In this variant of the game, the ship is always aimed at the fortress and subjects do not have to turn it. There are only two relevant keys: A left-hand press of the W key to add thrust to the ship and a right-hand press of the space bar to fire at the fortress. The ship begins each game aimed at the fortress, at a 9:00 starting position (Figure 1), and flying at a moderate speed parallel to the upper left diagonal segment of the outer hexagon. To avoid having their ship destroyed, subjects must avoid hitting the inner or outer hexagons, and they must fly fast enough to prevent the fortress from aiming, firing at, and hitting the ship. When subjects are successful the ship goes around the fortress in a clockwise They can destroy the fortress by shooting direction.

missiles at it to build up its vulnerability and then destroying it with a "kill shot" (two shots in rapid succession). If the fortress is destroyed, it leaves the screen for 1 second before respawning. If the ship is destroyed, it respawns after 1 second in the starting position flying along the starting vector. The replay site (http://andersonlab.net/reconstruction/) offers examples of game play.

Anderson et al. (2019) found that subjects can achieve relatively high and stable performance within an hour of playing AutoTurn (much faster than in original Space Fortress where subjects are also responsible for turning their ship among other things). To maintain a constant challenge of game play, a staircase procedure decreased the separation between the inner and outer hexagons as subjects got better. Subjects played 1-minute games. During the first 10 games the inner corners were 40 pixels from the center and the outer corners were 200 pixels from the center producing a width of 160 pixels. After the tenth game, the border width was reduced by 10 pixels if the subject had 0 or 1 deaths in the prior game and it was increased by 30 pixels (to a maximum width of 160 pixels) if they had 2 or more deaths. In this way the death rate in the game was maintained at about 1 death per 1-minute game. For each 10 pixels the border is reduced, subjects get an additional 10 points for each fortress they destroy. Navigation becomes more difficult as one has to fly between narrower borders, with many deaths resulting from thrusting into the inner hexagon, a rare event with the original 160-pixel width.

The Sketch procedure combines classification results from the EEG signal with information about the expected distribution of critical events from a cognitive model of the subject. The cognitive model we use was the ACT-R model that was described in Anderson (2019). We simulated 100 subjects by running the model 100 times on 60 games under the same game conditions as humans to generate behavioral results. We ran the model in over 35,000 games to generate statistics used in the Sketch procedure.

#### Methods

Here we describe data collection, pre-processing and procedures. We will refer to the reduced interface experiment as "Experiment 2" to contrast it with "Experiment 1" in Anderson, et al (2020).

#### **Subjects**

A total of 20 subjects (6 male, 14 female) were recruited from the CMU population of students and researchers between the ages of 18 and 40. None reported a history of neurological impairment. Subjects were paid between \$60 and \$75 for participation, depending on task performance. The duration of the experiment, including setup and task execution was less than 2 hours. All participants signed a written informed consent form. The experimental protocol was reviewed and approved by the Carnegie Mellon University Institutional Review Board.



Figure 2. Mean values (line) and standard errors (area around lines) per game for subjects and models as a function of game (a) border width; (b) points before bonuses for kills at narrow borders; (c) number of fortress destructions; (d) number of deaths.

#### Task

Subjects were given a verbal overview of the time course of the experiment and how to play the game, after which they interacted with the software at their own pace. After reviewing instructions displayed onscreen, they played 60 1minute games. Each 1-minute game yielded 1800 1/30 sec time frames or game ticks. The full game state is recorded by the software on every game tick. The record of game state included the keyboard (keys down/up) and all aspects of the display screen (direction, speed and location of the ship if alive, fortress orientation, presence of shells or missiles, etc.).

Three changes made from the game used in Anderson et al (2020). First, as already noted, we eliminated all explosions (visual and auditory effects). Second, in the original game one auditory tone accompanied each shot and another auditory tone accompanied a reset. This resulted in a quick double tone when there was a reset. In this version to eliminate the double tone, we used one tone when a shot resulted in an increment to vulnerability and another tone when there was a reset of vulnerability. Half of the subjects had one pairing of tones to the vulnerability changes while this was switched for the other half. Third, we changed the awarding of points. In the original game, as soon as the borders began to narrow (game 11) subjects received double the 100 points for a fortress kill. As described above, in this game they received an additional 10 points for each 10-pixel reduction of the border width. This change was introduced to keep subjects motivated to play at a higher level of difficulty.

#### **EEG** Analysis

The EEG was recorded from 128 Ag-AgCl sintered electrodes (10-20 system) using a Biosemi Active II System (Biosemi, Amsterdam, Netherlands). The EEG signal was recorded continuously for the entire experimental session and broken into 1-minute games. Portions of the game periods that included poor signal were excluded. Individual channels within an epoch were flagged based on having extreme values for mean absolute deviation, drift, or range. Flagged channels were interpolated. Epochs that still

contained channels with extreme values after these steps were flagged and rejected. This resulted in loss of the signal for an average of 2.3 seconds per game for games used in the decoding (44.4% of the games had no lost signal).

In order to get simple correspondence with the game state data, the 512 Hz data were then down-sampled to 30 Hz with default EEGLab anti-aliasing filtering applied. A one-second window around each game tick (14 game ticks before, the game tick, and 15 game ticks after) was used to classify whether a game tick contained a critical event. Thus each game tick had associated with it a vector of 30\*128=3840 electrode readings, representing regional effects, frequency effects below 30 Hz, and their interactions. The first 1000 components of the PCA of these vectors were used for classification.

#### Classification

We replicated the Sketch procedure described in Anderson, et al (2020). We focused our analysis on the last 55 games for each subject where performance is relatively stable while also employing the same game exclusion criteria used in experiment 1. Of the 1100 games, we excluded 10 games because of border width or relative inactivity by the subjects (the one and only game where the staircase procedure resulted in a border width of 30 pixels, 3 games where subjects failed to destroy a fortress without resetting or being killed, and 6 further games with 12 or fewer critical events) leaving 1090 games.

Classification was performed on the 1000-element vectors produced by the PCA to identify the critical events that determine the critical sketch of game activity. We used a leave-one-game-out method where for a given target game of a particular subject, linear discriminant classifier training was done using all remaining games for that subject and all games from the remainder of the subjects. The classifier was trained to label the EEG activity vectors with the critical event corresponding to the game tick the vector describes. To reflect the point that a subject's own data are likely the most relevant, the training games for each subject are weighted 15 times more than the games of other subjects. This leave-one-game-out procedure was repeated for every game to generate event probabilities across all 1090 games.



Figure 3. (a) EEG activity around the destruction of the fortress with the scalp profiles ranging from -4 to 4  $\mu$ V. (b) EEG activity around the death of a ship with the scalp profiles ranging -10 to 10  $\mu$ V. (c) EEG activity around a vulnerability reset with the scalp profiles ranging -10 to 10  $\mu$ V. Shaded areas represent a standard error of the mean calculated from the standard deviation of the subject means.

#### **Results**

#### **Behavioral Results**

The time course of various performance measures over 60 games are shown in Figure 2. Data shown include those from Experiment 1 labeled as 'Subjects Exp1', the reduced-interface-element Experiment 2 described above labeled as 'Subjects, labeled as Models. Games 1-10 all had a fixed border width of 160 pixels between the small inner hexagon that contains the fortress and the outer hexagon. After game 10, the staircase procedure was employed: border widths for successive games would continue to decrease at 10 pixel decrements until a subject's ship was destroyed 2 times or more, at which point the next game would reset to a larger width.

Part a shows border width. Subject behavior in both experiments results in slightly tighter border widths than those from model gameplay. Considering only games 11-60 where border width could vary according to the staircase procedure, Exp2 subjects attained somewhat tighter border widths (M = 98.2, SD = 13.46) than Exp1 subjects (M = 107.6, SD = 15.59, t(38)=2.04, p=.049 reflecting the change of scoring scheme from Experiment 1. Figure 1b shows canonical point scores by game. Canonical points show what subjects would achieve with the original 100 points per kill without the further bonuses they get for kills at narrow widths. Points were comparable for models and subjects over the course of the experiment, and there was a not a significant difference in points scored between Exp2 subjects (M = 627.1, SD = 122.08) and Exp1 (M = 655.5, SD = 106.76), t(38) = 0.782, p=.44. A similar pattern holds for fortress kills shown in Figure 1c, with roughly 9.5 kills per game in both Exp2 (M = 9.4, SD = 1.67) and Exp1 (M =9.7, SD = 1.32). Similarly, there was no difference in ship deaths (Figure 1d) between Exp2 (M = 0.9, SD = 0.12) and Exp1 (M = 0.9, SD = 0.13), both averaging just under 1 death per game which was the goal of the staircase manipulation.

#### **Generating a Sketch**

While the above performance measures show that behavioral performance is comparable between the enhanced and reduced versions of the game, the essential question we want to answer is whether and how features of the gaming interface affect the ability of the Sketch procedure to accurately assign the identity and timing of critical events throughout a game. There are five critical events that occurred during gameplay:

- 1. Kills. Player destroys the fortress and scores 100+ points.
- 2. Fortress Respawns. 1 second after the fortress is killed, it reappears and normal gameplay can resume.
- 3. Deaths. The player's ship is destroyed and the player loses 100 points.
- 4. Ship Respawns. The ship is absent for 1 second after death, then reappears and normal gameplay can resume.
- 5. Resets. If the interval between ship missile firing is less than 250ms and the vulnerability is less than 11, the fortress vulnerability will be set back to zero and the subject must begin rebuilding the vulnerability from scratch.

	Experi	ment 1	Experi	ment 2
Event	Hear	See	Hear	See
Ship Death	Whoosh	Explode		
Fortress Kill	Whoosh	Explode		
Missile Fired	HF Beep	$\rightarrow$		$\rightarrow$
Fortress Fired	LF Beep	$\diamond$		$\diamond$
Vuln Increase			Beep 1/2	
Vuln Reset	Beep		Beep 2/2	

Table 1: Interface Elements for Game Events

Table 1 shows the interface elements associated with various game events in both experiments. Experiment 2 has eliminated all unnecessary sounds and visual effects. Missile and shell firing are still accompanied by the visual display of the missile or shell flying across the space. Increments and decrements of vulnerability are indicated by distinctive tones so the subject does not have to be constantly looking at vulnerability on a different part of the screen. In addition, ship deaths and fortress deaths are

accompanied by a 1-second removal of the fortress or ship so the subject does not waste actions.

While the classification component of the Sketch procedure is multivariate in nature, it is useful to have a sense of the mean EEG activity around events that will be classified. We show the activity around a subset of critical events in Figure 3. Each of the panels shows a full second of activity (the same time-window used in the classification procedure), from 500 ms before the event to 500 ms after.

There seems to be a post-event positivity that is common to kills, deaths, and resets in both experiments, though in the current experiment, kills show only a return to baseline from negativity as opposed to positivity. Consistent with results reported in Anderson, et al (2020), the magnitude of this positivity in both experiments varies with the rarity of the event. Kills are most frequent and show the smallest positivity while deaths are the least frequent event and show the greatest return to positivity. This is consistent with what would be expected from a P300 (Polich, 2012).

## **Classification Results**

As in Anderson et al (2020), the leave-one-game-out cross validation procedure to predict labels for the 5 classes of critical events also requires inclusion of a sixth class containing null events. To avoid being overwhelmed by null events, for every critical game tick in a single game, 2 non-critical game ticks were chosen randomly to include in the classifier training phase. The overall discriminability d-prime was 1.76. Average accuracy was 54.0% and the average pairwise AUC was .915. This was slightly lower than Anderson et al. (2020) where d-prime was 2.0, average accuracy was 59.6% and the average pairwise AUC was .942.

As detailed in Anderson, et al (2020), the classification results themselves would not give us very good critical event sketches. For example, many of the null events are labeled as being critical events. Further, even if we managed to achieve unrealistically good classification accuracy, an unconstrained critical event sketch would contain sequences of events that are unlikely within the dynamics of the Space Fortress game. We need a way to tell the real critical events from the false labels and sequence events realistically. The Sketch method was developed for this purpose. This procedure combines statistics about what critical events are likely to occur when. This is calculated from a large library of model runs with output from the classifier to produce a critical sketch. The model games are used to estimate probabilities for a critical event transition matrix as well as latency distributions for time elapsed between events. The transition matrices and latency distributions are used to parameterize an HSMM.

The HSMM can efficiently combine the model-based statistics and conditional probabilities from the EEG classifier to estimate the most likely sequence of events in a game. Any sequence of events can be denoted  $a_1, a_2, ..., a_n$  occurring at game ticks  $t_1, t_2, ..., t_n$  where  $a_1$  is game start (and so  $t_1$  is game tick 1),  $a_n$  is the end ( $t_n$  is the 1800th game

tick), and  $a_2$ , ...,  $a_{n-1}$  are fortress kills and respawns, ship deaths and respawns, and vulnerability resets. Anderson, et al (2020) derived the following proportionality describing the probability of any such sequence relative to the probability of other sequences:

$$\Pr{ob(a_1, a_2 \cdots a_n)} \approx \prod_{i=1}^{n-1} trans(a_i, a_{i+1}) * f(t_{i+1} - t_i \mid a_i, a_{i+1}) * \frac{P(EEG(t_{i+1}) \mid a_{i+1})}{P(EEG(t_{i+1}) \mid Null)}$$

where  $trans(a_{i},a_{i+1})$  is the probability of transition between the events  $a_i$  and  $a_{i+1}$  estimated from the model runs,  $f(t_{i+1} - t_i | a_i, a_{i+1})$  is the probability of the  $t_{i+1} - t_i$  game ticks between the events  $a_i$  and  $a_{i+1}$ , instantiated with the distributions computed from the model runs, and  $P(EEG(t_i+1, t_{i+1}) | a_{i+1})$  is the conditional probability of the EEG signal for this period if it ends in  $a_{i+1}$  where the conditional probabilities are generated from the classifier.

The Viterbi algorithm (Rabiner, 1989) for hidden semi-Markov models was used to find the assignment of events (event identity and timestamp) that maximized  $Prob(a_1, a_2, a_3)$  $\dots$ ,  $a_n$ ). This produced for each game a critical event sketch: a set of inferred events and the time ticks when they occurred. We use two measures to evaluate the goodness of match between sketch and actual game events: recall and precision (Buckland & Gey, 1994). We focus only on kills, deaths and resets (ignoring respawns of ship and fortress as they were directly tied to kills and deaths with a 1 second lag). The recall measure considers all actual game events that occur and the identity of the closest sketch event to each. If the identity of the closest sketch event matched the actual game event, the assigned recall score would be the distance in time ticks between them. If the sketch and actual event time tick were identical, the score would be 0. If the sketch event was further than 2.5 seconds (75 time ticks) away, or if the identity of the sketch event did not match, a score of 75 was assigned. The precision measure used the same scoring procedure but was anchored to predicted sketch events and evaluated match to the closest game events.



Figure 4: Event placement rating distributions for both experiments and chance performance.

Figure 4 shows the distribution of recall and precision scores for Experiments 1 and 2 and provides a comparison to chance (reconstructions randomly paired with games). The mean recall and precision was 14.1 and 11.8 for Experiment 1, 18.4 and 18.5 for Experiment 2, and 48.1 and 47.0 for chance. While the reconstructions for both experiments are far better than chance, the difference in recall is significant (t(38)=2.25, p<.05) as is the difference in precision (t(38)=2.85, p<.01).

### Conclusion

A straightforward conclusion seems to emerge when comparing sketch results from the embellished Experiment 1 to relatively impoverished Experiment 2: While there remains enough information in the cognitive response to events to achieve a fairly high-quality sketch of the events in Experiment 2, the sketch accuracy is somewhat lower than in Experiment 1, reflecting the slightly poorer classification performance, likely a result of reduced game feedback elements. As Figure 2 shows, the current ACT-R model only approximately matches subject performance. A direction for improvement of reconstruction in either experiment would be a further improvement in that model.

#### Acknowledgments

This material is based upon work supported by the Office of Naval Research Grant N00014-15-1-2151 and USAF AFRL under Contract No. FA8649-20-C-0008. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ONR or USAF AFRL.

#### References

- Abiri, R., Borhani, S., Sellers, E. W., Jiang, Y., & Zhao, X. (2019). A comprehensive review of EEG-based brain– computer interface paradigms. *Journal of Neural Engineering*, 16(1), 11001.
- Anderson, J. R., Betts, S., Bothell, D., Hope, R., & Lebiere, C. (2019). Learning Rapid and Precise Skills. *Psychological Review*, 126, 727–760.
- Anderson, J. R., Betts, S., Fincham, J. M., Hope, R., & Walsh, M. W. (2020). Reconstructing fine-grained cognition from brain activity. *NeuroImage*, 221. https://doi.org/10.1016/j.neuroimage.2020.116999
- Aricò, P., Borghini, G., di Flumeri, G., Colosimo, A., Bonelli, S., Golfetti, A., Pozzi, S., Imbert, J.-P., Granger, G., Benhacene, R., & others. (2016).
  Adaptive automation triggered by EEG-based mental workload index: a passive brain-computer interface

application in realistic air traffic control environment. *Frontiers in Human Neuroscience*, *10*, 539.

- Brouwer, A. M., Hogervorst, M. A., van Erp, J. B., Heffelaar, T., Zimmerman, P. H., & Oostenveld, R. (2012). Estimating workload using EEG spectral power and ERPs in the n-back task. *Journal of Neural Engineering*, 9(4), 45008.
- Buckland, M., & Gey, F. (1994). The relationship between recall and precision. *Journal of the American Society for Information Science*, 45(1), 12–19.
- Delorme, A., & Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134(1), 9–21.
- Destefano, M. (2010). *The mechanics of multitasking: The choreography of perception, action, and cognition over 7.05 orders of magnitude*. Rensselaer Polytechnic Institute.
- Donchin, E. (1989). The learning-strategies project: Introductory remarks. *Acta Psychologica*, 71(1–3), 1– 15.
- Frederiksen, J. R., & White, B. Y. (1989). An approach to training based upon principled task decomposition. *Acta Psychologica*, 71(1–3), 89–146.
- Gopher, D., Weil, M., & Siegel, D. (1989). Practice under changing priorities: An approach to training of complex skills. *Acta Psychologica*, *71*, 147–178.
- Kim, M. K., Kim, M., Oh, E., & Kim, S. P. (2013). A review on the computational methods for emotional state estimation from the human EEG. *Computational and Mathematical Methods in Medicine*.
- Krol, L. R., Andreessen, L. M., & Zander, T. O. (2018). Passive brain–computer interfaces: a perspective on increased interactivity. In *Brain–Computer Interfaces Handbook* (pp. 69–86). CRC press.
- Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., & Yger, F. (2018). A review of classification algorithms for EEG-based brain– computer interfaces: a 10 year update. *Journal of Neural Engineering*, 15(3), 31005.
- Noh, E., Herzmann, G., Curran, T., & Sa, V. R. (2014). Using single-trial EEG to predict and analyze subsequent memory. *NeuroImage*, *84*, 712–723.
- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257–286.
- Saeidi, M., Karwowski, W., Farahani, F. v, Fiok, K., Taiar, R., Hancock, P. A., & Al-Juaid, A. (2021). Neural Decoding of EEG Signals with Machine Learning: A Systematic Review. *Brain Sciences*, 11(11), 1525.

# On the Limits of Spreading Activation in ACT-R: Predictions and Testability

Christopher Fisher<sup>1\*</sup> (christopher.fisher@cubic.com), Brent D. Fegley<sup>2\*</sup> (bfegley@aptima.com),

Christopher A. Stevens,<sup>3</sup> & Christopher W. Myers<sup>3</sup>

<sup>1</sup>Cubic Defense, <sup>2</sup>Aptima, Inc., USA, <sup>3</sup>Air Force Research Laboratory, USA

#### Abstract

In the fan effect, reaction time (RT) increases as a function of fan size (i.e. the number of associations of a fact). Spreading activation in ACT-R provides a good account of the fan effect at low fan size (i.e., 1–4). However, little is known about the predictions of ACT-R at ecologically valid scales. We developed a general guessing mixture model (GMM) within ACT-R in which a guessing process is triggered by retrieval failures, and analyzed the predictions for fan sizes much larger than those used in laboratory experiments. Our analysis revealed the following properties of the GMM: RT increased as a function of fan size, but stays within a plausible range (< 2 seconds) as long as the retrieval threshold is not excessively low, and, in the limit, accuracy asymptotes at the value of the guessing bias parameter. We discuss practical challenges with testing the predictions at larger fan sizes.

**Keywords:** ACT-R; spreading activation; fan effect; simulation study; declarative memory; retrieval threshold

## Introduction

One goal of cognitive architectures is to develop unified theories of cognition that scale to complex tasks in realistic environments (Newell 1990). Part of this larger goal is identifying memory processes and representations that support the retrieval of information from an extensive knowledge base. To achieve this goal, it is necessary to stress test existing theories and identify boundary conditions where the predictions may breakdown. Confidence in a theory will invariably increase if it survives rigorous stress testing. However, a failure provides an opportunity to revise the theory or develop alternatives. In either case, pushing the limits of a theory can provide important scientific insights and serve as a catalyst for scientific progress.

One theoretical question concerning the ACT-R cognitive architecture (Anderson 2007) is whether there are limits in the ability of spreading activation to account for the classic fan effect as the fan size increases. The fan effect is a phenomenon whereby retrieval time increases as the number of associations with a fact i.e. the fan size increases (Anderson 1974). For example, it takes longer to verify whether the hippie was located in the house if he or she was known to be in three places rather than one place. According to ACT-R, the fan effect arises through spreading activation in which a fixed quantity of activation, evenly distributed among associations in memory, spreads through a semantic network. As the fan size increases, the amount of activation distributed to each memory decreases, leading to slower retrieval time.

In a typical fan experiment, the fan size ranges from 1 to 4 (Anderson 1974; Sohn et al. 2004). ACT-R provides an accurate description of the fan effect within this limited range of fan size. Whether the fan effect increases with larger fan

size and whether ACT-R continues to provide an accurate account remain open questions. From a theoretical standpoint, these questions are interesting because spreading activation may greatly inhibit the retrieval of requested information at large fan sizes, leading to low accuracy. Nonetheless, humans seem to retrieve information effectively even though the fan size in certain knowledge domains might be large, such as autobiographical memory. From a practical standpoint, this question is interesting for modeling human knowledge in applied domains. Given that human knowledge is extensive and associations among some facts may be high, what are the implications for predicting retrieval time and accuracy? Our goal is to analyze ACT-R's predictions at these boundary conditions.

#### **Overview**

The remainder of the paper is organized as follows. First, we describe how the fan effect is typically studied in a paired associates recognition memory task. Next, we present a general model of the fan effect and analyze several submodels including the model presented in the ACT-R tutorial (ACT-R Research Group n.d., Tutorial Unit 5). We compare the submodels at fan sizes much larger than have been examined in the laboratory. Finally, we detail some practical limitations in testing the predictions of spreading activation at scale. We conclude with a discussion of the theoretical and practical implications of our findings.



Figure 1: A bipartite graph of person-place pairs in a typical fan experiment (ACT-R Research Group n.d., Tutorial Unit 5). Nodes represent persons or places and edges represent associations between nodes. The number of edges connecting to a node represents the fan size.

#### Fan Effect

The fan effect is typically studied in a paired associates recognition task (Anderson 1974; Anderson and Reder 1999). During the learning phase, subjects study a series of word pairs that vary in fan size. For example, consider the network diagram in Figure 1. Each node represents either a person or a place, each edge represents connections between two nodes, and the number of edges connected to a node corresponds to its fan size. In Figure 1, *hippie* has a fan size of f = 3, whereas *earl* has a fan size of f = 1. During the test phase, subjects must indicate "yes" if the word pair was studied or "no" if the word pair was not studied. Half of the test trials are *targets* in which the person and place were studied together as a pair, such as (earl, castle) in Figure 1. The remaining test trials are *foils* formed by switching person and place values across studied pairs such that the person and place in the new pair were not studied together. An example of a foil based on Figure 1 is (earl, cave).

Typically, fan size is manipulated factorially across a small range of values for the person and place attributes. Considering that we are interested in the predictions at large fan sizes, we will simplify the design by setting f equal for both attributes. At minimum, this design requires two sets of pairs, with  $f^2$  pairs in each set for a total of  $2 \cdot f^2$ . Having two sets of pairs ensures that sufficient pairs exist for creating foils with equal f.

## **General Model**

Our analysis of ACT-R is organized around a general model of the fan effect which we term the guessing mixture model (GMM). In the GMM, responses are determined by a mixture of a retrieval process and a guessing process. As described below, the fan model presented in ACT-R Tutorial 5 is a special case of the GMM. Figure 2 depicts the structure of the GMM as a processing tree in which each node represents a state and each branch represents a transition between states. Each path-defined as a series of branches-terminates in a "ves" or "no" response. The probability of traversing a path is the product of branch probabilities within the path. The marginal probability of a response is computed as the sum of all branch probabilities that map to the response. For example, the probability of responding "yes" on a target trial is  $Pr(yes | target) = t_m + (1 - t_m - t_{mm}) \cdot g$ , which is composed of two paths: a path in which the matching chunk is retrieved and a path in which a retrieval failure occurs and the response "yes" is produced through a guessing process. Below, we will show how the transition probabilities in Figure 2 can be expressed in terms of ACT-R's memory retrieval mechanisms.

#### **Knowledge Representation**

In ACT-R, declarative memory consists of a set of chunks  $M = {\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n}$ . A chunk is a basic unit of declarative knowledge. For a given fan size f, we assume that declarative memory consists of a minimum required  $2 \cdot f^2$  chunks corresponding to each studied pair. Formally, a given chunk m is a collection of slot-value pairs denoted as  $\mathbf{c}_m = {(s_i, v_i)}_{i \in I_m}$ , where  $s_i$  and  $v_i$  are the slot and value of pair i, and  $I_m$  is the index set for the elements (slot-value pairs) of chunk m. An example of a chunk in a typical fan experiment is  $\mathbf{c}_m = {(person, hippie), (place, park)}$ , which indicates the hippie is in the park. We will represent the mapping from slots to values as  $c_m(s) = v$ , which is empty or null if slot s is not in  $\mathbf{c}_m$ . Continuing with the example above, we can express the mapping between place and park as  $c_m(place) = park$ .



Figure 2: A tree diagram of the guessing mixture model. Panel [a]: process tree for target trials where  $t_m$  is the probability of retrieving matching chunk on target trial,  $t_{mm}$  is the probability of retrieving mismatching chunk on target trial, and g is the probability of guessing "yes". Panel [b]: process tree for foil trials where  $f_{mm}$  is the probability of retrieving mismatching chunk on a foil trial

#### **Retrieval Process**

Upon submitting a retrieval request r to declarative memory, a set of matching chunks R compete for retrieval and the chunk with the highest activation is retrieved if it exceeds the retrieval threshold,  $\tau$ . A retrieval failure occurs if the highest activation is less than  $\tau$ . The retrieval request is a mixture of retrieving from the person slot-value pair with probability w or the place slot-value pair with 1 - w. Although in the tutorial w = .5, in our design, the value of w does not matter because f is equal for both attributes. On this basis, we can simplify the model by setting the mixture probability to w = 1 so that the person slot-value pair is always used as the retrieval request. We represent a retrieval request similarly to a chunk, which is defined as  $\mathbf{r} = \{(\text{person}, v)\}$  where v is the value associated with the person slot. Upon submitting the retrieval request to declarative memory, a set R of candidate chunks compete for retrieval:

$$R = {\mathbf{c}_m \in M : c_m(\text{person}) = r(\text{person})}$$

The number of chunks matching the retrieval request is f. On a target trial, 1 chunk in R matches the stimulus on the person and place slot-value pairs. The remaining f - 1 chunks match only on the person slot-value pair. On foil trials, each of the f chunks match only on the person slot-value pair.

#### Activation

In ACT-R, each chunk has a memory activation value representing the log odds it will be encountered or needed (Anderson 2007). As activation increases, the probability and speed with which the chunk is retrieved also increases. Activation for chunk m is defined as

$$a_m = \beta + \mathbf{S}\mathbf{A}_m + \mathbf{\varepsilon}_m \tag{1}$$

where  $\beta$  is the base level constant, SA is the spreading activation term,  $\varepsilon \sim \text{normal}(0, \sigma)$  is activation noise, and  $\sigma$  is the standard deviation. We will use  $\beta$  to represent activation associated with relatively stable, asymptotic learning. Based on the assumptions we introduced, we can simplify the spreading activation term for the following two cases. The spreading activation term for chunk *c* which matches the stimulus on both the person and place slot value pairs is defined as

$$SA_{c} = \frac{1}{2}[\gamma - \ln(f+1)] + \frac{1}{2}[\gamma - \ln(f+1)] = \gamma - \ln(f+1)$$
(2)

where  $\gamma$  is the maximum association parameter. The other case occurs when the chunk only matches on one slot-value pair of the retrieval request, which is given by:

$$SA_i = \frac{1}{2} [\gamma - \ln(f+1)] \tag{3}$$

According to the ACT-R documentation, SA is truncated at zero by default, stating that undesirable behavior may occur with negative values (Bothell 2020, December 21, p. 287). Negative values occur when  $f > e^{\gamma} - 1$ . However, to be consistent with the theoretical interpretation of activation as log odds, which ranges between  $-\infty$  and  $\infty$ , we do not impose any restrictions on SA. In ACT-R, retrieval time is the following inverse function of activation:  $t_m = Fe^{-a_m}$  where *F* is the latency factor parameter with a default value of 1.

#### **Response Mapping**

As shown in Figure 2, the GGM uses the following response mapping: if the retrieved chunk matches the stimulus, the model responds "yes"; if the retrieved chunk does not match the stimulus, the model responds "no"; if a retrieval failure occurs, the model guesses "yes" with probability g.

#### **Response Probabilities**

Although the results we report below are based on Monte Carlo simulations of ACT-R, we will express the model in terms of approximate equations to provide a deeper understanding of the factors that determine the predictions. Using  $\mu_c$  and  $\mu_i$  as the expected activation for the cases based on Equations (2) and (3), the probability of correctly responding "yes" on a target trial can be approximated with the following softmax function (Weaver 2008):

$$Pr(yes \mid target) = \frac{e^{\mu_c/\sigma}}{e^{\mu_c/\sigma} + (f-1) \cdot e^{\mu_i/\sigma} + e^{\tau/\sigma}} + \frac{e^{\tau/\sigma}}{e^{\mu_c/\sigma} + (f-1) \cdot e^{\mu_i/\sigma} + e^{\tau/\sigma}} \cdot g \quad (4)$$

where *s* is the logistic scalar parameter,  $\sigma = s\sqrt{2}$ , and  $\tau$  is the retrieval threshold. Accuracy initially decreases as *f* increases because the the preponderance of chunks eligible for retrieval (f - 1 out of f) do not match the target. However, in the limit, responding is driven entirely by guessing because activation becomes much lower than  $\tau$ . We can see this behavior in Equation (4) where the term for retrieval failures,  $e^{\tau/\sigma}$ , has the largest exponent and thus determines the limit. As *f* increases, the first term on the right approaches zero whereas the second term approaches *g*. Setting Equation (4) to h(f), we can state:  $\lim_{f\to\infty} h(f) = g$ .

On foil trials, the probability of a correctly responding "no" is given by:

$$\Pr(\text{no} \mid \text{foil}) = \frac{f \cdot e^{\mu_i/\sigma}}{f \cdot e^{\mu_i/\sigma} + e^{\tau/\sigma}} + \frac{e^{\tau_i/\sigma}}{f \cdot e^{\mu_i/\sigma} + e^{\tau/\sigma}} \cdot (1 - g)$$
(5)

Setting Equation (5) to z(f), the limiting behavior of the GMM is  $\lim_{f\to\infty} z(f) = 1-g$  based on the same logic used for target trials.

#### **Simulation Study**

In this section, we analyze the predictions of two special cases of the GMM: the fan effect model from the tutorial, and an extension of the tutorial model with noise added to memory activation and the retrieval threshold. Table 1 lists the parameter values used for both submodels. For each combination of parameters, we repeated the simulation 5,000 times to ensure that stable predictions were generated.

Table 1: Parameter symbols, their descriptions, and values used in the submodels: tutorial model (TM), tutorial model plus noise (TM+N).

parameter	description	ТМ	TM+N
β	base level constant	0	1.5
τ	retrieval threshold	0	0, 2
γ	max associative strength	1.6	1–4
S	activation noise	0	0.2
F	latency factor	0.63	1
g	guess yes	0.5	0.5
t <sub>er</sub>	non-retrieval time	.845*	.845*

\*0.050 seconds is added for guessing.

#### **Tutorial Model (TM)**

The ACT-R Tutorial Unit 5 model for the fan effect (TM) is a special case of the GMM with parameter values specified in Table 1. One important characteristic of the TM is that retrieval time is deterministic because s = 0. In the tutorial, specifying a guessing process was unnecessary because the maximum fan size of 3 ensured that all  $a_m > \tau$ . However, a guessing process must be incorporated into the model

to handle retrieval failures at larger fan sizes, which incurs an additional overhead of .050 seconds. Incorporating the guessing process requires adding two production rules with noisy utility values selected to produce the desired guessing probability. We assume that guessing is unbiased (i.e., g = .50).

The TM predicts an instantaneous drop in accuracy from 100% to 50% when fan size forces activation below the retrieval threshold. Rounding to the next integer, this occurs at f = 4 with the specific parameters in Table 1. In general, the shift in accuracy occurs on target trials when

$$f > e^{\gamma + \beta - \tau} - 1 \tag{6}$$

and on foil trials when

$$f > e^{\gamma + 2(\tau - \beta)} - 1 \tag{7}$$

RT predictions are also affected by an abrupt shift from retrieving chunks to retrieval failures. In Figure 3, RT increases with fan size on both target and foil trials until activation decreases below  $\tau$  at f = 4. When  $f \ge 4$ , retrieval failures trigger a guessing process that produces the same constant RT for correct and incorrect responses regardless of increases in f. In general, the switch to guessing occurs when Equation (6) and Equation (7) are true, in which case the predicted RT becomes  $Fe^{-\tau} + t_{er} + .05$  seconds regardless of increases in f. In addition, when  $f \le 4$ , RTs for correct responses on foil trials are greater than the RTs for the corresponding responses on target trials. The reason is that only one source of spreading activation contributes to retrieved chunks on foil trials whereas two sources contribute to retrieved chunks on target trials.



Figure 3: RT predictions of the TM model paneled by target and foil trials.

#### **Tutorial Model Plus Noise (TM+N)**

The TM suffers from the following limitations: (1) RTs are unrealistically deterministic, and (2) accuracy drops instantaneously from 100% to 50% once activation decreases below  $\tau$ . In light of these limitations, we investigate a less restrictive special case of the GMM that we term the Tutorial Model Plus Noise (TM+N). The TM+N differs from the TM in one important way: noise is added to both memory activation and the retrieval threshold. Adding noise to activation and the retrieval threshold improves the model in two ways. First, the TM+N predicts a distribution of times for retrievals and retrieval failures rather than a deterministic time. Given that human RTs are variable, some have argued that adding noise to the retrieval threshold makes the model more plausible (Weaver 2008; Nicenboim and Vasishth 2018). Second, the TM+N predicts a gradual decrease in accuracy as a function of fan size rather than an immediate drop from 100% to 50%.



Figure 4: The probability of a correct response for trial types across fan sizes for TM+N as a function of  $\tau$  and  $\gamma$  pair. Note that the x-scale from 1-50 is stretched to prevent overplotting.

Four noteworthy patterns for accuracy can seen in Figure 4. First, all other things being equal, accuracy is higher for larger values of  $\gamma$ . Second, on target trials, accuracy is a non-monotonic function of fan size, beginning above the asymptote at g = 0.50 and decreasing below g = 0.50 before increasing to g = 0.50. If activation is sufficiently larger than  $\tau$ , accuracy will decrease to zero before converging on the asymptote. Third, on foil trials, accuracy starts high and decreases towards the asymptote at 1 - g = .50 as fan size increases. Fourth, the speed with which the trends change increases with smaller differences between  $\gamma$  and  $\tau$ . In other words, guessing dominates responding sooner when activation begins closer to the retrieval threshold.

Several important trends for RTs are present in Figure 5. First, RT increases as a function of fan size, but stays within a plausible range of approximately .90 to 1.6 seconds. Second, unlike the TM, the TM+N shows smooth curves for correct and incorrect RTs rather than an abrupt change from correct to incorrect RTs. Third, as expected, RTs were faster for higher values of  $\gamma$ . Fourth, RTs are faster when  $\tau = 2$  than when



Figure 5: Median RT by fan size, trial type (target or foil), and response for TM+N as a function of  $\tau$  and  $\gamma$ . Vertical lines are interquartile ranges. Horizontal lines depict the trajectory of median RT. Note that x-scale is nonlinear to prevent overplotting.

 $\tau = 0.$ 

## **Practical Challenges**

Testing the boundary conditions of spreading activation is fraught with several practical challenges. One challenge is that the time complexity for completing the study phase is quadratic (i.e.,  $O(f^2)$ ) because the fan experiment requires at minimum  $2 \cdot f^2$  pairs. Consequently, the duration of the study phase will quickly become impracticable as fan size increases. For example, suppose subjects must complete *b* practice blocks to reach a target learning criterion. Suppose further that each pair will require *t* seconds on average to study. Thus, the learning phase will require  $t_{\text{total}} = t \cdot b \cdot 2 \cdot f^2$  seconds to complete. Figure 6 shows the duration of the study phase as a function of *f* and *b* with t = 2 seconds. Depending on *b*, the duration of the study phase ranges between approximately .7 and 2.1 hours for f = 25, and quickly increases to a range of 11 to 33 hours for f = 100.

A second challenge is counteracting memory decay by increasing practice blocks. As shown in Figure 7, the time between consecutive presentations of the same word pair grows in a non-linear fashion with respect to fan size. With a fan size of 25, the time difference between presentations is nearly .70 hours which poses difficulties for learning. As fan size increases, so will the number of practice blocks required to counteract increasing amounts of memory decay between consecutive presentations of the same word pair.

The test phase, by contrast, offers more flexibility because it is not necessary to test the entire stimulus set. Instead, one could sample a random subset for testing. Although this would reduce the duration of the test phase and mitigate the effects of decay, it would come at the cost of lower statistical power.



Figure 6: Study time as a function of fan size and the number of practice blocks.

Based on our analysis, it is clear that increasing fan size beyond 10–15 in a single experimental session would become prohibitively difficult. One way to increase fan size beyond 10–15 is to distribute practice across multiple sessions. Although using multiple practice sessions would make the time of a single session manageable, it suffers from inter-session decay effects and the potential for attrition. It is worth noting that researchers would likely vary fan size across several values to test the functional form of the fan effect, in which case the time demands would be even greater. For example, an experiment with fan sizes 2, 5, and 10 would require a one hour study phase assuming t = 2 seconds, b = 3, and an equal number of pairs per fan size:  $\frac{3\cdot 2f^2 \cdot t \cdot b}{60^2} = \frac{3\cdot 2\cdot 10^2 \cdot 2\cdot 3}{60^2} = 1$  hour.

#### Discussion

Previous research has supported ACT-R's predictions for the fan effect within a small range of fan size. However, little is known about how ACT-R's predictions scale to ecologically



Figure 7: The difference in hours between two consecutive times a pair is studied. Medians are depicted as points, the interquartile ranges depicted as whitespace around the median, and minimum and maximum values at the outer extremes of the bounding lines.

valid domains in which fan size is likely large. In light of this gap in the literature, we set out to accomplish two goals: (1) to analyze the predictions of ACT-R at an ecologically valid range of fan size, and (2) to assess the practical challenges with studying the fan effect within ecologically valid parameters.

In service of the first goal, we analyzed the properties of a general guessing mixture model (GMM), with an emphasis on the two special cases: the tutorial model (TM) and the tutorial model plus noise (TM+N), an extension of the TM with noise in the retrieval process. Across a broad range of conditions, three findings emerged: (1) RT stayed within a plausible range (.9 - 1.6 seconds) despite low memory activation at high fan size, (2) RT decreased with increases in the maximum association parameter, and (3) accuracy eventually reaches an asymptote equal to the guessing parameter, g, on target trials, and 1 - g on foil trials. The TM suffered from two limitations due to its assumption that memory retrieval is deterministic: (1) unrealistically deterministic RTs, and (2) instantaneous switching from perfect accuracy to guessing. Adding noise to the retrieval process to produce the TM+N eliminated these limitations. Interestingly, the TM+N can produce non-monotonic behavior where accuracy drops below the asymptote-sometimes as low as 0% accuracybefore increasing to the asymptote.

Our analysis revealed that the time complexity for running a fan effect experiment is quadratic, meaning that the time demands quickly become prohibitive as fan size increases. This is exacerbated by the fact the time between consecutive presentations of the same study pair also grows quickly with fan size, leading to substantial decay. Additional study blocks would be needed to counteract memory decay during increasingly long study sessions, putting tests at large fan size even further out of reach.

One of the most interesting findings from our analysis is that accuracy drops to guessing levels somewhat quickly under a wide range of parameter settings. The primary factor in determining how quickly accuracy drops is the difference between activation and the retrieval threshold. As this difference decreases, accuracy decreases more quickly. Given that the predictions depend on this relationship, it is necessary to empirically test ACT-R at large fan sizes. As our analysis revealed, doing so will be challenging and there are practical limits to the maximum fan size that can be tested. Nonetheless, testing ACT-R at larger fan sizes—even if only as large as 10 or 20—will be important in assessing ACT-R's robustness, and determining ACT-R's scalability in practical situations with large knowledge domains.

#### 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. The contents have been reviewed and deemed Distribution A. Approved for public release. Case number: AFRL-2022-2020

#### References

- ACT-R Research Group. (n.d.). *ACT-R* » *Software* (7.21.6-<3099:2020-12-21>). Retrieved December 31, 2021, from http://act-r.psy.cmu.edu/software/
- Anderson, J. R. (1974). Retrieval of propositional information from long-term memory. *Cognitive Psychology*, *6*(4), 451–474. https://doi.org/10.1016/0010-0285(74)90021-8
- Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press.
- Anderson, J. R., & Reder, L. M. (1999). The fan effect: New results and new theories. *Journal of Experimental Psychology: General*, 128(2), 186–197. https://doi. org/10.1037/0096-3445.128.2.186
- Bothell, D. (2020, December 21). ACT-R 7.21+ Reference Manual. https://act-r.psy.cmu.edu/actr7.x/ reference-manual.pdf
- Newell, A. (1990). *Unified theories of cognition*. Harvard University Press.
- Nicenboim, B., & Vasishth, S. (2018). Models of retrieval in sentence comprehension: A computational evaluation using bayesian hierarchical modeling. *Journal* of Memory and Language, 99, 1–34. https://doi.org/ 10.1016/j.jml.2017.08.004
- Sohn, M.-H., Anderson, J. R., Reder, L. M., & Goode, A. (2004). Differential fan effect and attentional focus. *Psychonomic Bulletin & Review*, 11(4), 729–734.
- Weaver, R. (2008). Parameters, predictions, and evidence in computational modeling: A statistical view informed by ACT-R. *Cognitive Science*, 32(8), 1349– 1375.

# A Comparison of Quantum and Multinomial Processing Tree Models of the Interference Effect

Christopher R. Fisher (christopher.fisher.27.ctr@us.af.mil)

Cubic Defense Beavercreek, OH 45324 USA

Joseph W. Houpt (joseph.houpt@utsa.edu)

University of Texas at San Antonio San Antonio, TX 78249 Lorraine Borghetti (lorraine.borghetti.1.ctr@us.af.mil)

Air Force Research Laboratory Wright Patterson AFB, OH USA

Leslie M. Blaha (leslie.blaha@us.af.mil) Air Force Research Laboratory Wright Patterson AFB, OH USA

#### Christopher Stevens (christopher.stevens.28@us.af.mil)

Air Force Research Laboratory Wright Patterson AFB, OH USA

#### Abstract

We compare the qualitative predictions of an existing quantum model and a novel multinomial processing tree (MPT) model of the interference effect using parameter space partitioning (PSP). An interference effect occurs when categorizing a stimulus changes the marginal probability of a subsequent decision, leading to a violation of the LOTP. The PSP analysis revealed that our MPT model can produce the same qualitative patterns as the quantum model. Further analysis, however, revealed that the models differ in several important ways. First, a larger volume of the MPT model's parameter space produces a smaller number of interference effects compared to the quantum model. Second, the distribution of volume across patterns is more diffuse for the MPT model, indicating it is more flexible than the quantum model. We discuss limitations and future directions.

**Keywords:** Multinomial processing trees; Quantum cognition; Interference effects; Categorization; Model flexibility

## Introduction

An interference effect occurs when an action or judgment changes the marginal probability of a subsequent decision (Wang & Busemeyer, 2016; Busemeyer et al., 2011). One reason interference effects are interesting from a theoretical perspective is that they violate a law of classical probability theory (CPT) called the *law of total probability* (LOTP). Adherence to the LOTP means that for decision D and set of categories  $\{C_i\}_{i\in I}$ , the marginal distribution of D is given by  $Pr(D) = \sum_{i=1}^{n} Pr(D | C_i) Pr(C_i)$ . Previous research has demonstrated that categorizing face interferes with the subsequent decision to attack, such that  $Pr(D) \neq \sum_{i=1}^{n} Pr(D | C_i) Pr(C_i)$  (Wang & Busemeyer, 2016; Busemeyer et al., 2011).

Interference effects present a challenge for many models that are based on CPT because they violate the LOTP. For example, two models based on CPT—a Markov model and a signal detection model—are unable to account for the entire pattern of interference effects that have been observed empirically (Wang & Busemeyer, 2016). By contrast, a quantum model called the belief-action entanglement (BAE) model provides an account of the interference effect (Wang & Busemeyer, 2016). The reason that the BAE model is successful in accounting for interference effects is that the less restrictive axioms of quantum probability theory allow for the violation of the total law of probability.

Our primary goal is to demonstrate as a proof of concept that a model based on CPT can produce interference effects. Specifically, we show that a multinomial processing tree (MPT; Riefer & Batchelder, 1988) composed of a categorization process, a category revision process, and a decision process is sufficient to produce interference effects. Our second goal is to compare the qualitative patterns of interference effects that the new model and the BAE model can produce. Understanding the prediction space of a model is important for understanding its behavior, assessing flexibility, and identifying diverging predictions between different models. An overly flexible model provides a less persuasive account of the data than a less flexible model (Roberts & Pashler, 2000).

The remainder of this article is organized as follows. In the next section, we describe the categorization-decision paradigm used to study interference effects. Next, we provide a brief overview of the BAE quantum model of interference effects. We then introduce a new MPT model which can also produce the empirical pattern of interference effects. We compare the qualitative patterns of interference effects each model can produce using a method called parameter space partitioning (Pitt et al., 2006). We conclude with a discussion of the limitations of the proposed model and the need for a unified account of interference effects, order effects and other phenomena based on CPT.

#### **Categorization-Decision Paradigm**

One popular paradigm for studying interference effects is the categorization-decision sequential choice paradigm (Wang & Busemeyer, 2016). Prior research with this paradigm has demonstrated that inclusion of an explicit categorization stage interferes with subsequent decision making (Wang & Busemeyer, 2016). On each trial, subjects are presented with a face and must decide whether to attack or withdraw. Each face is either a good guy, who is likely to be friendly, or a bad guy who is likely to be hostile. Although subjects do not know the category associated with each face (good vs. bad), they can use facial features, such as width, as cues to

aid in the decision process. For simplicity, we define type-b and type-g faces as faces most likely to be in the bad or good category, respectively. The extended paradigm involves three conditions (Wang & Busemeyer, 2016). In the decision-only condition (d), subjects make a single decision: to attack or withdraw from each face. In the categorize and decide condition (cd), subjects categorize each face as good or bad before proceeding to the attack/withdraw decision. In the third condition (xd), subjects are given the true category of each face prior to making a decision.

According to many models based on CPT, the marginal probability of attacking (irrespective of category membership) should be equal in each condition as required by the LOTP, which states:

$$\Pr_{d}(A = a \mid F = x) =$$

$$\Pr_{cd}(A = a \mid F = x, C = g)\Pr_{cd}(C = g \mid F = x) + (1)$$

$$\Pr_{cd}(A = a \mid F = x, C = b)\Pr_{cd}(C = b \mid F = x)$$

where random variables A, F and C represent the action, facial feature, and category, respectively. Possible actions are a for attack and w for withdrawal; possible values for facial feature are tb for type-b and tg for type-g, and possible categories are b for bad and g for good. Each probability statement is subscripted by its condition; for example, cd is the categorize and decide condition. The left-hand-side represents the case in which no category judgment is made, and the right-hand-side represents two possible cases-one in which the face is categorized as bad, and another in which the face is categorized as good. Because good and bad are mutually exclusive and exhaustive states of the world, the probability of each state should sum to the probability in which neither state is known. If this equation is true, the LOTP holds, and no interference effect occurs. However, if the LOTP does not hold, it follows that the act of categorizing the face interferes with the subsequent decision.

An example of a typical interference effect pattern can be found in Table 1. The pattern is typified by interference effects of approximately equal magnitude but opposite direction in the xd condition, a positive interference effect for typeb faces in the cd condition, and the absence of an interference effect for type-g faces in cd condition. This asymmetrical pattern in cd has been challenging for CPT models, such as signal detection and Markov models, to predict (Wang & Busemeyer, 2016).

Table 1: Interference effects reported in Experiment 2 of Wang & Busemeyer (2016). Values are computed as the difference of the left and right hand side of Equation 1.

	ĸd	с	d
type-b	type-g	type-b	type-g
0.03	-0.03	0.04	0.00

## **Belief-Action Entanglement Model**

The belief-action entanglement (BAE) model is a quantum model of interference effects (Wang & Busemeyer, 2016). Importantly, the axioms on which quantum models are based allow for the violation of certain rules in classical probability, such as the LOTP. In the BAE, beliefs are represented by four orthornormal basis vectors corresponding to the four combinations of category (good vs. bad) crossed with action (attack vs. withdraw). Prior to making a decision, a person is in an indefinite state called a superposition, which is a linear combination of the four basis vectors.

During the deliberation process, a person's indefinite state evolves according to a wave function with different potentials to attack or withdraw. The decision dynamics are governed by four utility parameters which represent the utilities of attacking under different conditions. For example,  $\mu_{tg,b}$  is the utility of attacking a type-g face that has been categorized as bad. The utility parameters are are assumed to be symmetric for type-g faces:  $\mu_{tg,g} = -\mu_{tg,b}$ . However, for type-b faces, the utilities can be asymmetrical, which allows interference to occur. In the d and cd conditions, entanglement aligns beliefs and actions to be consistent with each other. A parameter,  $\gamma$ , controls the degree of entanglement as well as its direction. Importantly, the entanglement and the utility parameters interact to produce interference effects. An interference effect will occur whenever the entanglement parameter is nonzero and the utility parameters for a given feature type (e.g., type-b) are asymmetrical (e.g.,  $\mu_{bg,b} \neq -\mu_{tb,g}$ ). The BAE also includes a parameter *j*, which represents the probability of categorizing a face into its most likely category (e.g., type-g categorized as good).

#### **Judgment Revision Model**

We developed a novel multinomial processing tree (MPT) model of the categorize-decide paradigm called the Judgment Revision model (JRM). Although the JRM is based on CPT, it can produce interference effects under specific conditions. A MPT characterizes how latent cognitive processes map onto categorical responses which follow a multinomial distribution (Riefer & Batchelder, 1988). As the name implies, MPTs are organized as a tree-like structure in which nodes represent cognitive states or processes and branches that connect nodes represent the transition from one cognitive state or process to another. Each branch is associated with a parameter representing a transition probability between cognitive states or processes. A series of transitions ultimately terminates at a response node representing a specific response category. The probability of following a specific path to a response node (i.e., a series of connected branches) is computed as the product of transition probabilities. In a MPT, several paths can terminate at a response node representing the same response category; in this case, the marginal probability of a specific response is the sum of all path probabilities linked to the response category.

The JRM assumes interference effects emerge from the in-
teraction of three cognitive processes. The first cognitive process is the decision to attack a face, which is represented by parameter a. The probability of attacking depends on both the face type and the category of the face, leading to the use of two indices: (1) the first index represents the feature type (tg for type-g and tb for type-b), and (2) the second index represents the category (g for good and b for bad). The second cognitive process is the categorization of a face as good or bad. The parameter *i* represents the probability of categorizing a face into its most likely category (e.g., type-g as good). The third cognitive process is the decision to continue with the initial category judgment or to revise it, which is captured by parameter c. With probability c, a person is certain in the initial category judgment and continues to the decision process without revising the initial category. With probability 1-c, a person is uncertain in the initial category judgment and revises it from good to bad (or vice versa) before continuing to the decision process. As we detail later, if one can assume that certainty in the categorization (i.e., c) can vary across some conditions, the JRM can produce the observed interference effect pattern.

#### Predictions

**Category Given Condition** In the xd condition, subjects are given both the feature and the category cues prior to making a decision to attack or withdraw. Parameters j and c play no role in this condition because the correct category information is provided, thus leading to simplified equations. The probability of attacking a type-b face in category b is:

$$\Pr_{\mathsf{xd}}(A = a \mid F = \mathsf{tb}, C = b) = a_{\mathsf{tb},b}.$$

The probability of attacking a type-b face in category g is:

$$\Pr_{\mathrm{xd}}(A = a \mid F = \mathrm{tb}, C = g) = a_{\mathrm{tb},g}$$

The probability of attacking a type-g face in category *b* is:

$$\Pr_{\mathrm{xd}}(A = a \mid F = \mathrm{tg}, C = b) = a_{\mathrm{tg},b}$$

The probability of attacking a type-g face in category g is:

$$\Pr_{\mathrm{xd}}(A = a \mid F = \mathrm{tg}, C = g) = a_{\mathrm{tg},g}$$

To compute the marginal probability of attacking in the *xd* condition, it is necessary to multiply the conditional attack probabilities by the objective category probabilities, p. For this, we assume that p is the same for both b and g faces; thus p is the probability that a face belongs to the most probable category (e.g., type-g is in category g). Formally,

$$p = \Pr(C = g \mid F = \mathsf{tg}) = \Pr(C = b \mid F = \mathsf{tb}).$$

The marginal probability of attacking a type-g face in the



Figure 1: Example trees for the cd condition. The top tree represents the categorization process for a type-b stimulus in the cd condition. The bottom tree represents the decision process for a type-b stimulus categorized as bad

xd condition is:

$$\begin{aligned} \Pr_{\mathrm{xd}}(A &= a \mid F = \mathrm{tg}) \\ &= \sum_{n \in \{g, b\}} \Pr_{\mathrm{xd}}(A = a, C = n \mid F = \mathrm{tg}) \\ &= \sum_{n \in \{g, b\}} \Pr_{\mathrm{xd}}(A = a \mid C = n, F = \mathrm{tg}) \Pr(C = n \mid F = \mathrm{tg}) \\ &= p \cdot a_{\mathrm{tg}, g} + (1 - p) \cdot a_{\mathrm{tg}, b}. \end{aligned}$$

Similarly, the marginal probability of attacking a type-b face in the *xd* condition is:

$$\Pr_{\mathrm{xd}}(A = a \mid F = \mathrm{tb}) = p \cdot a_{\mathrm{tb},b} + (1-p) \cdot a_{\mathrm{tb},g}$$

**Categorize and Decide Condition** In the *cd* condition, subjects are instructed to categorize the face before deciding whether to attack or withdraw. The first tree in Figure 1 illustrates the categorization process for a type-b face. In the first branch, a type-b face is categorized as good with probability 1 - j. In the second branch, a type-b face is categorized as bad with the complementary probability *j*. The probability of categorizing a type-b face as good is given by:

$$\Pr_{\rm cd}(C=g \mid F={\rm tb})=1-j$$

The probability of categorizing a type-g face as good is:

$$\Pr_{\rm cd}(C = g \mid F = tg) = j$$

After categorizing the face, a person must decide to attack or withdraw. As shown in Figure 1, there are two paths leading to a decision to attack. In the first path, a person is certain with probability c and continues with the initial category judgment of bad. The face is then attacked with probability  $a_{\text{tb},b}$ . In the second path, a person is uncertain with probability 1 - c and revises the initial category judgment from bad to

good. Next, the face is attacked with probability  $a_{tb,g}$ . This process can be represented mathematically with the following equation:

$$\Pr_{cd}(A = a \mid F = tb, C = b) = c \cdot a_{tb,b} + (1 - c) \cdot a_{tb,g}$$

One important point to note is that the JRM does not require certainty in category judgments to be equal in all conditions. In particular, we assume that c is higher in the cd condition in which a type-b face is categorized as good. The c parameter in this condition is denoted as  $c_k$  to distinguish it from c in the other conditions. Importantly, when  $c_k > c$ , the JRM can produce a positive interference effect for type-b faces in the *cd* condition. Without this assumption, the JRM can only produce interference effects in the xd conditions. Justification for this assumption can be found in Table 2 where certainty is measured as the degree to which conditional attack probabilities are close to the boundaries 0 or 1. As expected, we tend to see more certainty in xd because all information is provided. However, this pattern is reversed for type-b face categorized as good in the *cd* condition. Thus, we assume  $c_k > c$ . The probability of attacking a type-b face categorized as good is:

$$\operatorname{Pr}_{\operatorname{cd}}(A = a \mid F = \operatorname{tb}, C = g) = c_k \cdot a_{\operatorname{tb},g} + (1 - c_k) \cdot a_{\operatorname{tb},b}.$$

The probability of attacking a type-g face categorized as bad is given by:

$$\Pr_{cd}(A = a \mid F = tg, C = b) = c \cdot a_{tg,b} + (1 - c) \cdot a_{tg,g}$$

The probability of attacking a type-g face categorized as good is given by:

$$\Pr_{cd}(A = a \mid F = tg, C = g) = c \cdot a_{tg,g} + (1 - c) \cdot a_{tg,b}$$

Table 2: Conditional attack probabilities reported in Wang & Busemeyer (2016) Experiment 2.

	Good		Bad		
	type-g	type-b	type-g	type-b	
Certain ( <i>xd</i> )	0.28	0.40	0.58	0.69	
Uncertain (cd)	0.33	0.37	0.53	0.61	

The marginal probability of attacking is found by combining the equations for category judgment and decision processes. The marginal probability of attacking a type-b face in the *cd* condition is given by:

$$Pr_{cd}(A = a \mid F = tb) = (1 - j) \cdot [c_k \cdot a_{tb,g} + (1 - c_k) \cdot a_{tb,b}] + j \cdot [c \cdot a_{tb,b} + (1 - c) \cdot a_{tb,g}]$$

The marginal probability of attacking a type-g face in the *cd* condition is given by:

$$\Pr_{cd}(A = a \mid F = tg) = j \cdot [c \cdot a_{tg,g} + (1 - c) \cdot a_{tg,b}]$$
$$+ (1 - j) \cdot [c \cdot a_{tg,b} + (1 - c) \cdot a_{tg,g}]$$

**Decision Only Condition** In the d condition, subjects simply make the decision to attack or withdraw from each face. The JRM assumes that an implicit categorization precedes the decision to attack. The marginal probability of attacking a type-b face in the d condition is given by:

$$\Pr_d(A = a \mid F = \mathsf{tb}) = (1 - j) \cdot [c \cdot a_{\mathsf{tb},g} + (1 - c) \cdot a_{\mathsf{tb},b}]$$
$$+ j \cdot [c \cdot a_{\mathsf{tb},b} + (1 - c) \cdot a_{\mathsf{tb},g}].$$

The equation above provides four paths leading to a decision to attack. The first two paths begin with categorizing a type-b face as good with probability 1 - j. In the first path, a person is certain in the category judgment with probability c and continues without revision. From there, a person attacks with probability  $a_{\text{tb},g}$ . In the second path, a person is uncertain in the initial category judgment with probability 1 - c and revises it from good to bad. From there, a person attacks with probability  $a_{\text{tb},b}$ .

The other two paths begin with categorizing a type-b face as bad with probability j. In the third path, a person is certain in the category judgment with probability c and continues without revision. From there, a person attacks with probability  $a_{tb,b}$ . In the fourth path, a person is uncertain in the initial category judgment with probability 1 - c and revises it from bad to good. From there, a person attacks with probability  $a_{tb,g}$ . The marginal probability of attacking a type-g face in the d condition is given by:

$$\Pr_d(A = a \mid F = tg) = j \cdot [c \cdot a_{tg,g} + (1 - c) \cdot a_{tg,b}] + (1 - j) \cdot [c \cdot a_{tg,b} + (1 - c) \cdot a_{tg,g}].$$

### **Parameter Space Partitioning**

We found that the JRM and BAE provide similar quantitative fits to the data, so we focus instead on comparing their prediction spaces. A model that predicts any pattern provides little evidence for a theory, no matter how well it fits a particular data set (Roberts & Pashler, 2000). Thus, it is important to know the range of patterns a model can and cannot produce. For this reason, we compare the prediction space of both models using a qualitative model comparison method called parameter space partitioning (PSP; Pitt et al., 2006). PSP explores the parameter space of a model to identify regions associated with different qualitative data patterns. In contrast to model fitting which assess the quantitative fit of a model to a specific data set, the goal of PSP is to understand the behavior of the model across its entire parameter space. In addition, PSP uses volume estimation to determine the prevalence of various patterns in the parameter space.

In total, the paradigm can produce a maximum of 81 possible interference effect patterns. Specifically, the interference effect is computed as the difference between the left hand and right hand side for the definition of the LOTP in Equation 1. The resulting difference yields three types of interference effects: positive, negative and absent (i.e. a approximate difference of zero). An interference effect is computed in four conditions by crossing face type (type-g,type-b) and condition (xd, cd). Thus, in total, there are  $3^4 = 81$  possible patterns in the present paradigm. Our criteria for classifying an effect as absent was a small effect:  $|\Pr_d(A = a|F = x) - \Pr_z(A = a|F = x)| \le 0.01$ , where  $x \in \{\text{tg}, \text{tb}\}$  and  $z \in \{\text{xd}, \text{cd}\}$ .

We analyzed two versions of the BAE and the JRM:

(1) a relatively constrained version denoted by subscript c, and (2) a relatively unconstrained version denoted by subscript u. In the JRM<sub>c</sub>, we constrained the judgment certainty parameters to be equal:  $c_k = c$ . In the JRM<sub>u</sub>, we allowed  $c_k > c$ . In the BAE<sub>c</sub> model, we constrained  $\mu_{tg,b} = -\mu_{tg,g}$  as described the original paper (Wang & Busemeyer, 2016). In the BAE<sub>u</sub> model, no such constraint was imposed. Except where constraints apply, the allowable parameter ranges were  $j \in [0, 1]$  and  $\mu_{tb,b}, \mu_{tb,g}, \mu_{tg,g}, \mu_{tg,b}, \gamma \in [-2, 2]$  in the BAE, and [0, 1] for all parameters in the JRM.

#### Results

# Flexibility

One way to assess flexibility is to count the number of patterns a model can produce. As expected, Table 3 shows that the constrained BAE<sub>c</sub> model produced  $3^3 = 27$  patterns because it cannot produce interference effects for type-g faces in cd. By contrast, the BAE<sub>u</sub> can produce all 81 possible patterns. As expected, the JRM<sub>c</sub> only produced the 9 interference effects in xd condition. However, the JRM<sub>u</sub> can produce the same 27 patterns as the BAE<sub>c</sub> model.

One limitation with using pattern counts to assess flexibility is that it does not take into account the volume of regions associated with a data pattern. Although two models may produce the same number of data patterns, one model may concentrate most of its volume on a small subset of patterns whereas a highly flexible model might produce a uniform distribution of volume across patterns. We used the Gini coefficient (Gini, 1921)-an economic measure of income inequality-to better quantify the flexibility of the models. A value of 0 corresponds to maximal flexibility (i.e., a uniform distribution) whereas a value of 1 indicates minimal flexibility (i.e., all volume assigned to one pattern). As shown in Table 3, the Gini coefficient varies markedly across models, but all models are far from maximal flexibility. Although the JRM<sub>c</sub> is the least flexible model, it cannot account all empirical patterns (e.g, it cannot produce an interference effect in *cd* for type-b faces). In agreement with the pattern count, the  $BAE_u$  model is the most flexible model. Although the  $BAE_c$ model and the  $JRM_u$  model produce the same patterns, the  $BAE_c$  model is less flexible.

#### Volume

Next, we analyze the volume of regions associated with different patterns, which are normalized as a percentage of the volume for the entire parameter space. One challenge with comparing the volume of patterns between the models is the large number of patterns (81). Our solution to this problem is to analyze volume according to three factors: the type of interference effect (positive, negative, or absent), the number

Table 3: A summary of the qualitative pattern of interference effects produced by the BAE and JRM models. n is the number of possible patterns for the model. Gini is a coefficient of inequality. Volume % for patterns with at least one positive interference effect, at least one negative interference effect, and at least one absent effect.

model	n	Gini	positive	negative	absent
$BAE_c$	27	.868	80.1%	76.6%	100.0%
$BAE_u$	81	.656	81.4%	83.8%	72.3%
JRM <sub>c</sub>	9	.910	58.1%	56.1%	100.0%
JRM <sub>u</sub>	27	.791	73.0%	73.2%	100.0%

of interference effects, and the condition.

Table 3 shows the volume associated with positive, negative and absent interference effects. For example, a pattern was considered positive if at least one interference effect in the four conditions was positive. Volume for positive and negative interference effects were similar within each model. Volume for positive and negative interference effects was higher for BAE models compared the JRM models. The volume for at least one absent interference effect was high across all models.

Across all models, the volume estimates in Table 4 indicate that volume for interference effects in the *xd* condition was larger than for the *cd* condition. The volume in the *xd* condition was greater for the BAE models compared the the JRM models. As expected, the JRM<sub>c</sub> did not produce any interference effects in the *cd* condition. Only the BAE<sub>u</sub> model had sufficient flexibility to produce interference effects in the *cd* condition for type-g faces.

Table 4: Volume % as a function of condition and face type.

model	xd type-b	xd type-g	cd type-b	cd type-g
$BAE_c$	94.6%	97.1%	44.7%	0.0~%
$BAE_u$	94.1%	95.4%	46.3%	46.3%
JRM <sub>c</sub>	72.6%	71.3%	0.0%	0.0%
JRM <sub>u</sub>	70.2%	70.4%	63.3%	0.0%

Table 5 shows the estimated volume as a function of number of interference effects (positive or negative) for each model. As expected, the  $JRM_c$  produced a maximum of two interference effects; the  $JRM_u$  and the  $BAE_c$  produced a maximum of three interference effects, and the  $BAE_u$  produced a maximum of four interference effects. Generally speaking, the JRM models tend to predict a smaller number of interference effects than the BAE models.

### Discussion

Our goal was to develop a MPT model of the interference effect and compare its qualitative predictions to those of the

interference effects	BAE <sub>c</sub>	BAE <sub>u</sub>	JRM <sub>c</sub>	JRM <sub>u</sub>
0	0.6%	0.8%	14.2%	9.4%
1	5.1%	5.0%	27.7%	18.5%
2	51.8%	33.1%	58.1%	30.8%
3	42.6%	33.4%	0.0%	41.3%
4	0.0%	27.7%	0.0%	0.0%

Table 5: Volume % as a function of number of interference effects for each model.

BAE quantum probability model. Our MPT model, termed the JRM, is based on three cognitive processes: a categorization process, a category revision process, and a decision process. Although the JRM is based on CPT, it can produce interference effects if the judgment certainty can differ across conditions.

We used PSP to compare the models in terms of the data patterns they can and cannot produce. This is important because a model's ability to account for an observed data pattern is less impressive if it can predict many rather than few patterns (Roberts & Pashler, 2000). Our PSP analysis produced three noteworthy findings. First, an unconstrained version of the BAE can produce all qualitative interference effect patterns, and the JRM with constraints fails to produce the observed pattern of interference effects in the cd condition. Second, although the unconstrained JRM and the constrained BAE produce the same patterns of the interference effect, the BAE is less flexible because the volume across patterns is less diffuse compared to the JRM. Third, the volume analysis indicates that the JRM tends to generate fewer interference effects compared to the BAE. In summary, the JRM shows promise as an alternative to the BAE, as it can also produce the empirical pattern of interference effects. However, the BAE has the advantage of being less flexible according to the PSP analysis.

#### Limitations

We note a few limitations. One limitation is that PSP implicitly assumes the prior distribution across parameters is uniform. An extension of PSP incorporating information about the prior probability of parameters may yield different conclusions. The JRM has at least one limitation. In contrast to the BAE, the JRM does not generalize to experiments with different reward rates or associations among features and categories because it uses a parameter for each decision probability. One possible solution to this problem would be replacing the attack probability parameter *a* with a utility function mapping stimulus inputs to decision probabilities.

#### Conclusion

One advantage of quantum cognition is its ability to account for a wide range of phenomena, such as order effects and interference effects, with similar mechanisms (Busemeyer et al., 2011). A unified account of these phenomena based on CPT has yet to emerge. Instead, modeling efforts, including this one, have focused on demonstrating that models based on CPT can produce effects that are relatively easy for models of quantum cognition to produce. Recently, for example, several CPT-based models of order effects (which violate the commutative law of CPT) have been proposed, including a MPT model (Kellen et al., 2018), an ACT-R model (Fisher et al., 2021), and a Bayesian network model (Moreira & de Barros, 2021). The wide variety of models in these demonstrations indicates that the current challenge is not one of feasibility. Indeed, models based on different assumptions can produce the effects. Instead, this lack of consensus points to a deeper theoretical challenge in providing an alternative unified account of order effects, interference effects, and other phenomena. A viable alternative to quantum cognition must ultimately seek to provide a unified account. Nonetheless, developing an alternative model of interference effects is a necessary first step in this direction.

#### 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 by Air Force Office of Scientific Research grant 21RHCOR080. Distribution A Approved for public release. Case number: AFRL-2022-2015.

#### References

- Busemeyer, 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.
- Fisher, C. R., Borghetti, L., Houpt, J. W., & Stevens, C. (2021). An ACT-R model of order effects. In *Proceedings* of the 19th international conference on cognitive modeling.
- Gini, C. (1921). Measurement of inequality of incomes. *The Economic Journal*, *31*(121), 124–126.
- Kellen, D., Singmann, H., & Batchelder, W. H. (2018). Classic-probability accounts of mirrored (quantum-like) order effects in human judgments. *Decision*, 5(4), 323– 338.
- Moreira, C., & de Barros, J. A. (2021). Order effects in Bayesian updates. In *Proceedings of the annual meeting of* the cognitive science society (Vol. 46).
- 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.
- Riefer, D. M., & Batchelder, W. H. (1988). Multinomial modeling and the measurement of cognitive processes. *Psychological Review*, 95(3), 318–339.

Proceedings of the 20th International Conference on Cognitive Modelling (ICCM 2022)

- Roberts, S., & Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. *Psychological Review*, *107*(2), 358–367.
- Wang, Z., & Busemeyer, J. R. (2016). Interference effects of categorization on decision making. *Cognition*, *150*, 133–149.

# Informational Trade-offs of Learning from Expert Demonstration

Rebekah Gelpi\* (rebekah.gelpi@mail.utoronto.ca) Department of Psychology, University of Toronto Toronto, ON, Canada Yikai Tang\* (yikai.tang@mail.utoronto.ca) Department of Psychology, University of Toronto Toronto, ON, Canada

William A. Cunningham (wil.cunningham@utoronto.ca)

Department of Psychology, University of Toronto

Toronto, ON, Canada

Keywords: deep learning; reinforcement learning; cultural transmission; pedagogy; social learning

# Introduction

Human beings develop in a highly complex social and physical environment. Behaving appropriately in this environment requires learning detailed action sequences, where intermediate actions do not provide obvious instrumental rewards. Alongside a high degree of general-purpose intelligence, humans have adapted to this computational challenge through a deep reliance on learning through the cultural transmission of information from teachers or other social sources (Boyd et al., 2011; Mesoudi et al., 2006). This deep cognitive adaptation is expensive, requiring a large investment of each generation of humans in providing for and teaching the subsequent generation, and an extended period of childhood longer than that observed in other animals (Gopnik, 2020). During this time, children are both dependent on caregivers for resources, and spending a large amount of energy on brain development.

Nevertheless, learning from expert demonstrators obviates the need to engage in time-consuming and even possibly dangerous exploration to discover solutions already known by other members of society, and allows for cultures to develop new tools and technologies by allowing its members to build upon previous knowledge cumulatively (Tennie et al., 2009).

Teaching provides many opportunities for learning above and beyond serving as another source of information for a learner. Because teachers are intentional agents, it is possible to make strong assumptions behind the rationale for their behavior, leading to stronger inferences about the data than if it had been independently discovered (Shafto et al., 2014). However, here we focus on a simpler phenomenon: teachers tend to be more skilled, and observing an expert demonstrator can improve learning by providing learners with access to examples of success before they are able to succeed themselves. Indeed, prior work has found that using expert demonstrations to pretrain or guide exploration can substantially improve learning speed and performance in RL agents (e.g. Gulcehre et al., 2019; Zhang & Ma, 2018).

To investigate the benefits of expert demonstration, we develop and test a simple grid world game in which an agent

either learns through self-directed exploration, observation of a pre-trained expert demonstrator, or a combination of both of varying proportions.

# Method

We implement a 10  $\times$  10 grid world in which one agent, two bushes, and one wolf are located at coordinates in space. All the objects are randomly distributed throughout the world. The agent and the bushes have a certain energy level when they are instantiated. The agent's action space involves basic movements (up, down, left, and right) and eating, each consuming energy to perform. When the agent eats while adjacent to a bush, its energy level increases and the bush's energy level decreases. When an agent's energy level decreases to zero, the agent will 'die'; bushes with an energy level of zero no longer provide energy. Unlike the agent and the bushes, the wolf has unlimited energy. It intermittently hunts the agent with a predetermined action policy. The agent is rewarded when it eats bushes and when it survives for 50 turns, but it is punished when eaten by the wolf or when it starves.

# **Model Architecture**

The agent contains a deep Q-learning neural network (DQN) that takes in the location and identity of nearby objects as well as its own hunger level as its observation of the world. Observations are first input into an LSTM followed by a linear policy that outputs the estimated Q-value of the five possible state-action pairs (four cardinal directions plus eating). The agent also contains a replay buffer that stores past experiences, either from self-directed exploration or from a pre-trained expert demonstrator. After each epoch, the neural network samples a batch of multi-state game sequences, and updates its policy estimates based on the rewards obtained in these states.

# **Experimental Conditions**

We trained the agent for 200,000 games in one of five conditions. Each game is initialized with varying agent energy levels (between 15 and 100) and ends after 50 steps or when the agent dies. Individual games sometimes include a wolf, and sometimes do not. As a result, agents learn

about games that have differing optimal policies for survival (e.g. seek out food first, or avoid the wolf first).

We generated data for 5 agents, corresponding to differing levels of experience received from a pre-trained expert demonstrator. In Condition 1, the agent learns solely through its own experiences of interacting with the environment, and does not receive any expert demonstration. In Conditions 2–5, a gradually increasing proportion of the agent's learning trials correspond to a game played by an expert demonstrator (12.5%, 25%, 50%, and 100%, respectively). Every 1000 epochs, the agent is presented with 900 test games with an initial energy level of 15 in the grid world. We test agents' performance by recording the number of steps survived on the test trials.

# **Results and Discussion**

To assess the final performance of the model, we conducted a series of *t*-tests with Bonferroni correction for multiple comparisons to evaluate the performance of each fully trained model on 10000 new test games. We found that a proportion of 25% expert trials had a better performance than all other models (all p < .001), but also that models mixing both learning strategies outperformed the two that used only one or the other (all p < .001). Notably, the size of the performance increase from 25% expert trials compared to 100% expert trials (Cohen's d = 1.08) and self-directed learning (Cohen's d = 1.40) were both very large.



**Figure 1.** Average turns survived by agents for self-directed exploration (red), as well as 12.5% (yellow), 25% (purple), 50% (blue), and 100% expert demonstration (green) conditions. Results are averaged over 5 model runs. Shaded region indicates standard error value.

Overall, all pedagogical models substantially outperformed learning from self-directed exploration alone. Exposure to expert demonstrations led all agents to quickly improve well beyond the maximum average survival of the self-directed learning model. Nevertheless, not all forms of demonstrations were equally valuable. For example, being presented with only expert trials led agents to quickly stop improving their performance, with a ceiling achieved after 15 turns. This outcome reflected highly robust learning of how to avoid being eaten by a wolf, but an inability to reliably generalize a policy that included eating from the bushes to avoid starvation. In contrast, while other agents displayed a higher proportion of being eaten by a wolf, this was traded off against an ability to use self-directed learning to learn how to eat and thus survive longer on average.

# **Conclusions and Ongoing Research**

These simulations suggest that learning from an expert can provide an immediate advantage over learning from one's own error-prone first attempts, and that even small amounts of expert guidance can provide a lasting boost to one's total learning (e.g. Gulcehre et al., 2019). Nevertheless, it also shows that relying too heavily on an expert can limit one's learning—serving as a "double-edged sword" (e.g., Bonawitz et al., 2011) that limits one's capacity for future exploration. Instead, success requires balancing expert knowledge with exploration, echoing the iterative innovation process that is characteristic of human cumulative culture (Tennie et al., 2009).

We are currently investigating how dynamically shifting reliance on an expert can optimize its benefits. For example, when one has little idea of the best action policy, heavily drawing from an expert is highly beneficial; as one gains more personal experience, however, relying on one's own innovations becomes progressively more advantageous.

# Acknowledgments

We acknowledge the support of the Social Sciences and Humanities Research Council of Canada [SSHRC-506547] and the Natural Sciences and Engineering Research Council of Canada [RGPIN-2018-05946].

## References

- Bonawitz, E., Shafto, P., Gweon, H., Goodman, N. D., Spelke, E.,
  & Schulz, L. (2011). The double-edged sword of pedagogy: Instruction limits spontaneous exploration and discovery. *Cognition*, 120(3), 322–330.
- Boyd, R., Richerson, P. J., & Henrich, J. (2011). The cultural niche: Why social learning is essential for human adaptation. *Proceedings of the National Academy of Sciences*, 108, 10918–10925.
- Gopnik, A. (2020). Childhood as a solution to explore–exploit tensions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 375(1803), 20190502.
- Mesoudi, A., Whiten, A., & Dunbar, R. (2006). A bias for social information in human cultural transmission. *British Journal of Psychology*, 97(3), 405–423.
- Shafto, P., Goodman, N. D., & Griffiths, T. L. (2014). A rational account of pedagogical reasoning: Teaching by, and learning from, examples. *Cognitive Psychology*, 71, 55–89.
- Tennie, C., Call, J., & Tomasello, M. (2009). Ratcheting up the ratchet: On the evolution of cumulative culture. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1528), 2405–2415.
- Gulcehre, C.\*, Paine, T. L.\*, et al. (2019). Making Efficient Use of Demonstrations to Solve Hard Exploration Problems. arXiv:1909.01387
- Zhang, X., & Ma, H. (2018). Pretraining Deep Actor-Critic Reinforcement Learning Algorithms With Expert Demonstrations. arXiv:1801.10459

# Action Sequencing, Timing, and Chunking in Space Fortress

Pierre G. Gianferrara (<u>pgianfer@andrew.cmu.edu</u>) John R. Anderson (<u>ja0s@andrew.cmu.edu</u>)

Department of Psychology, Carnegie Mellon University, 5000 Forbes Avenue

Pittsburgh, PA, 15213 USA

**Keywords:** skill acquisition; practice; keypress; sequencing; timing; sensorimotor learning; motor behavior; chunking

# Introduction

Recent cognitive modeling research has been uncovering the complex mechanisms whereby humans learn to combine instruction and experience to acquire rapid and precise complex skills (Anderson *et al.*, 2019). Two key aspects of the learning include the proceduralization of declarative instructions (also known as "production compilation") and the progressive tuning of controllable movement properties to environmental features that predict success in a given task (i.e., internal model; see Anderson *et al.*, 2019).

One promising way of exploring sensorimotor learning during skill acquisition is to look at the details of motor behavior. For instance, it has been shown that motor timing and sequencing variability predicted skill acquisition in a simplified version of the *Space Fortress* (SF) video game (Gianferrara, Betts & Anderson, 2020, 2021). In this project, we focus on action timing and action sequencing in a *SF* video game instantiation with more complex dynamics called *YouTurn* (see Anderson *et al.*, 2019).

In *SF YouTurn*, players are flying a spaceship in a frictionless environment while shooting missiles at a fortress and avoiding shells. To navigate the spaceship, players use four possible keypress actions: "Fire" (F – space bar), "Turn Left" (L – 'A' key), "Turn Right" (R – 'D' key), and "Thrust" (T – 'W' key). To earn points, players accumulate fortress kills over 40 games of 3 minutes. To do so, players need to aim at the fortress and fire a sequence of 10 consecutive shots with intershot intervals of at least 250 ms, and conclude each game cycle with a final quick double shot (with an intershot interval faster than 250 ms). Each fortress kill was rewarded with 100 points, each fired missile cost 2 points, and players lost 100 points for each ship death.

# **Keypress Chunks over the Games**

The notion of motor chunking has been proposed as part of motor skill learning to account for the progressive increase in fluency and accuracy that is usually characteristic of skill acquisition (Diedrichsen & Kornysheva, 2015). Specifically, motor chunks can be thought of in terms of a hierarchical representation of motor skills in which groups of consecutive motor actions are fired collectively as motor units instead of separately as serial actions (Beukema & Verstynen, 2018; Diedrichsen & Kornysheva, 2015). Evidence for motor chunking comes from motor learning experiments, such as the serial reaction time task, in which participants' behavior progressively includes idiosyncratic sequential and temporal groupings, resulting in gradually higher response time autocorrelation at early lags (Verstynen *et al.*, 2012).

We explored the *SF YouTurn* video game dataset from Anderson *et al.* (2019) with N = 29 and looked for evidence of action chunking in terms of action sequencing and action timing. Based on past experimental evidence (e.g., Sakai, Kitaguchi & Hikosaka, 2003), we considered that groups of two consecutive keypresses  $K_l$  and  $K_2$  were more likely to be "chunked" when their inter-press interval (IPI) was lower, and when their relative frequency was higher. We thus expressed chunk propensities as follows:  $p(chunk_i) = \frac{X(chunk_i)}{Z_k^{n=16}X(chunk_k)}$  where  $X(chunk_i) = \frac{Freq \,\% (chunk_i)}{IPI(K_1K_2)}$ . We computed this propensity for each of the 16 2-keypress chunks over the 40 games.

Figure 1 depicts the progression of each 2-keypress chunk propensity over the 40 games. Figure 2 depicts the average keypress transition probabilities across all 16 chunks. The main result is that as players acquired skills, they tended to preferentially select chunks with a "fire" action while purely navigational chunks became less frequent over the games.



Figure 1: Progression of all 16 2-keypress chunk propensities over the 40 3-min. *SF YouTurn* games.



Figure 2: *SF YouTurn* chunk probability network (created in *Cytoscape<sup>1</sup>*). Estimated transition probabilities are shown with edges. Thicker and redder edges are more probable. Edge labels indicate Markov transition probabilities relative to their respective source keypress nodes.

<sup>&</sup>lt;sup>9</sup> <sup>1</sup>Institute for Systems Biology. (2019). *Cytoscape*. Retrieved from https://www.cytoscape.org

In the context of the *SF YouTurn* video game, shots are particularly important since points are awarded based on participants' ability to pace their "fire" keypress actions. However, the game's frictionless space and speed requirements (i.e., ships get killed if they are too slow) impose additional navigational constraints which must be dealt with simultaneously. The results from Figures 1 and 2 suggest that participants increasingly bound shots with other navigational keypresses as part of action chunks in order to build up skill over the games.

This example of chunking is reminiscent of past incidental learning work on artificial grammars in which participants were asked to remember unfamiliar string sequences, and unintentionally learned strings' environmental statistical regularities (Servan-Schreiber & Anderson, 1990; Perruchet & Pacton, 2006). In such work, chunk formation and hierarchical representational structures were shown to provide an advantage in terms of memory consolidation and recall during learning (Servan-Schreiber & Anderson, 1990). Applied to motor skill learning, a growing body of evidence suggests that elementary movements that are bound into chunks may be retrieved faster and more accurately than individually selected movements (Diedrichsen & Kornysheva, 2015; Beukema & Verstynen, 2018).

# **Motor Correlates of Skill Acquisition**

We next broke down motor skill learning into separate measures of action sequencing and action timing variability. Following the methodology introduced by Gianferrara, Betts & Anderson (2021), we plotted the entropy which measured keypresses' sequential variability in *SF YouTurn*. With 4 keys, there are  $4^3 = 64$  keypress triples<sup>2</sup>. The entropy was computed as  $H(X) = -\sum_{i=1}^{64} p_i . log_2 p_i$  and ranged from 0 to 6. We also plotted players' action timing variability in terms of the logarithmic coefficient of variation of the inter-shot intervals (ISI) such that  $log CV(ISI) = log(\sigma(ISI)/\mu(ISI))$  where  $\sigma(ISI)$  refers to the standard deviation of the ISIs, and  $\mu(ISI)$  refers to their mean. Figure 3 shows the progression of skill in terms of players' performance score (3a), action sequential variability (3b), and shot timing variability (3c).



Figure 3: a) Performance over time. b) Action sequencing variability (Entropy) over time. c) Shot timing variability (Log CV ISI) over time. Shaded areas indicate the S.E.M.s.

Note that we filtered out games with no completed game cycles. The current results show data from 1064 individual subjects' games (~92% of all game data).

We then averaged each of the three above measures within subjects across all 40 games to investigate inter-individual skill differences (see Figures 4a and 4b). The main result is that lower action sequencing and shot timing variability are correlated with higher scores.





# Predicting Skill based on Motor Behavior

Finally, we fit a linear mixed-effects model (LMEM) on game data to assess each measure's ability to predict skill over the games. In R, the model was written as lmer(Score ~ Entropy + logCV + (1|Subject) + (1|GameNb)). Note that 8 observations out of 1064 observations (~ 0.75%) were removed because of model residuals that were more than 3 SDs away from the mean and acted as high-leverage observations. Another model was fit to inter-individual skill data (across games) and was written as lm(Score ~ Entropy + LogCV). Results from both models are shown on Table 1. The main result is that lower measures of action sequencing and shot timing variability significantly predict higher scores across subjects and games.

Table 1: Predicting skill in the SF YouTurn video game.

	Skill predictions in YouTurn						
	LMEM across games			Inter-individual skills			
	Estimate	95 %	CI	Estimate	95 %	% CI	
Entropy	$-560^{***}$	(-631,	-492)	$-905^{***}$	(-1380,	-430)	
Log CV ISI	$-216^{***}$	(-258,	-174)	$-765^{***}$	(-1181,	-349)	
Adjusted $\mathbb{R}^2$		0.87			0.76		

#### Conclusion

We showed that our measures of action timing variability and action sequencing variability also predicted skill in a more complex video game closer to the original *Space Fortress* environment. This finding suggests that as players are acquiring skills, they also learn to chunk actions which results in more consistent and fluent motor behavior.

## Acknowledgments

This research was supported by the Office of Naval Research Grant N00014-15-1-2151 and AFOSR/AFRL award FA9550-18-1-0251.

<sup>&</sup>lt;sup>2</sup>For purposes of visual illustration, we only represented 2-keypress chunks on Figure 1. We expanded chunks' size to include all keypress triples for entropy computations based on the results from past motor skill learning research (Ariani *et al.*, 2021)

Proceedings of the 20th International Conference on Cognitive Modelling (ICCM 2022)

## References

- Anderson, J. R., Betts, S., Bothell, D., Hope, R., & Lebiere, C. (2019). Learning rapid and precise skills. *Psychological review*, 126, 727-760.
- Ariani, G., Kordjazi, N., Pruszynski, J. A., & Diedrichsen, J. (2021). The planning horizon for movement sequences. *Eneuro*, 8.
- Beukema, P., & Verstynen, T. (2018). Predicting and binding: interacting algorithms supporting the consolidation of sequential motor skills. *Current opinion in behavioral sciences*, 20, 98-103.
- Diedrichsen, J., & Kornysheva, K. (2015). Motor skill learning between selection and execution. *Trends in cognitive sciences*, 19, 227-233.
- Gianferrara, P. G., Betts, S., & Anderson, J. R. (2020). Timerelated Effects of Speed on Motor Skill Acquisition. In Stewart, T. C. (Ed.). *Proceedings of the 18th International Conference on Cognitive Modelling* (pp. 64-70). University Park, PA: Applied Cognitive Science Lab, Penn State.
- Gianferrara, P. G., Betts, S., & Anderson, J. R. (2021). Cognitive & motor skill transfer across speeds: A video game study. *PloS one*, 16, e0258242.
- Perruchet, P., & Pacton, S. (2006). Implicit learning and statistical learning: One phenomenon, two approaches. *Trends in cognitive sciences*, *10*, 233-238.
- Sakai, K., Kitaguchi, K., & Hikosaka, O. (2003). Chunking during human visuomotor sequence learning. *Experimental brain research*, 152, 229-242.
- Servan-Schreiber, E., & Anderson, J. R. (1990). Learning artificial grammars with competitive chunking. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 16*, 592-608.
- Verstynen, T., Phillips, J., Braun, E., Workman, B., Schunn, C., & Schneider, W. (2012). Dynamic sensorimotor planning during long-term sequence learning: the role of variability, response chunking and planning errors. *PLoS One*, 7, e47336.

# Estimating ACT-R Declarative Memory Parameters Using a Drift Diffusion Model

Gillian Grennan (gillkg@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

Accurately fitting cognitive models to empirical datasets requires a robust parameter estimation process which is often arduous and computationally expensive. A way to mitigate this challenge is to integrate participant-specific and efficient mathematical models such as a drift diffusion model (DDM) into the parameter estimation process of cognitive modeling. In this study, we exhibit a clear mapping of the parameters outputted by DDM onto the declarative memory parameters utilized in the cognitive architecture, ACT-R. We show a fairly consistent recovery of simulated ACT-R parameters using DDM and a successful application in using this method to optimize ACT-R simulated fit to an empirical dataset. Notably, we show that the DDM-derived estimated parameters are individualized to the original participant, providing a unique opportunity for parsing out individual differences in cognitive modeling. This method outlined here allows one to estimate ACT-R parameters without the need to manually build and run an ACT-R model while also allowing for neural contextualization of DDM parameters.

**Keywords:** Drift Diffusion Model, Cognitive Architecture, Computational models, Individual Differences

#### Introduction

A common challenge associated with cognitive modeling is how to accurately capture individual differences within the parameters that comprise these models. Parameter estimation now relies on unfastidious and computationally expensive methods such as manual parameter gridsearches. Incorporating a statistically rigorous and behaviorally-valid computational model such as a drift diffusion model into the parameter estimation process of ACT-R may allow for better empirically-informed ACT-R models. Similarly, although DDM has been widely replicated in behavioral paradigms and the outputted parameters show distinct and replicable behavioral correlates (Ratcliff & Tuerlinckx, 2002; Voss et al., 2004), studies examining the neural substrates of the DDM parameters have large variability in their results (Gupta et al., 2022). Integrating DDM into a well-established cognitive architecture such as ACT-R (Anderson, 2007) would allow DDM parameters to have robust neural correlate interpretations. Further, accurate ACT-R parameter estimation would eliminate the need for the modeler to manually build and run an ACT-R model to use for neural or cognitive interpretation in the context of declarative memory tasks, increasing the accessibility of these methods to a wider array of non-modeler researchers.

## **ACT-R Declarative Memory**

ACT-R is a well-established cognitive architecture that includes a highly reliable model of declarative memory (Anderson et al., 2004; Anderson, 1974; Kotseruba & Tsotsos, 2020; Pavlik & Anderson, 2005). Declarative memories or knowledge within ACT-R are encoded in record-like structures called chunks, representing semantic memories. ACT-R's declarative memory module functions by making less used chunks harder to retrieve over time through their assigned activations. Chunks are selected on the bases of their *activation*, a quantity that reflects the log odds that the chunk will be needed. Specifically, the activation  $A_c$  of a chunk c at time t is computed as:

$$A_c = \sum_i (t_i - t)^{-d} \quad (1)$$

where  $t_i$  represents the time of the *i*-th event in which *c* was encoded or retrieved. Retrieval of information from memory can be viewed as a process of evidence accumulation, where environmental or internal cues contribute evidence to competing chunks within one's memory. These chunks are competing for retrieval and the first chunk to accumulate enough evidence to be chosen, crosses a "decisional threshold" and a response is initiated (Anderson, 2007).

#### **Drift Diffusion Model**

A drift diffusion model (DDM; Ratcliff, 1978; Voss et al., 2013) has been proposed to model a two-alternative forced-choice task and is based on early models of the continuous random walk process (Stone, 1960; Wald & Wolfowitz, 1948). The DDM is based on several basic assumptions: during a binary decision process, information will accumulate at a continuous rate and this accumulation process can be explained using a Weiner diffusion process (Ratcliff & McKoon, 2008; Ratcliff & Tuerlinckx, 2002). Information accumulation is characterized by a constant systemic component with an added component of normally distributed random noise. This assumption of random noise is meant to emulate repeated processing of the same stimulus or same type of stimulus and explains the variance in response times and erroneous response errors observed in empirical reaction/accuracy distributions (Ratcliff & McKoon, 2008; Ratcliff & Tuerlinckx, 2002; Voss et al., 2013). The decision process is terminated as soon as the systemic counter accumulates information to the point of reaching

one of the two decisional thresholds. The basic model can be depicted in Figure 1A.

A drift diffusion model is distinguished by its distinct parameters estimated from empirical decision time distributions. The first parameter, or drift rate (v) is calculated through the average of the rate of evidence accumulation from the start of the decision process (beginning of evidence accumulation) until a decision is made (evidence accumulator reaches either upper or lower decisional threshold). Previous studies have shown that drift rate can be interpreted as a measure of cognitive speed and is affected by value associated with the stimulus as well as the separation between choices (Bond et al., 2018; Ratcliff & Frank, 2012). We are similarly able to estimate the decisional threshold (a). The decisional threshold represents the amount of evidence needed to make a decision. A higher decisional threshold indicates a larger distance between the lower and upper decisional thresholds. Decisional threshold has been shown to highly depend on a speed-accuracy tradeoff and is sensitive to changes in instructions emphasizing speed over accuracy or vice versa (Mulder et al., 2013). We are also able to calculate the decisional starting point, or decisional bias (z). The decision starting point represents the starting bias at the beginning of the decision process and represents the relative distance to the upper/lower decisional threshold. A higher decision starting point would represent bias towards the upper decisional threshold. Finally, we are able to estimate the extradecisional time component  $(t_0)$ which represents the time used to complete all processes not directly related to the decisional process such as stimulus encoding or motor execution of the response.

#### Mapping DDM Parameters onto ACT-R

Recent work has shown that we can treat the ACT-R declarative memory module as an evidence accumulator model, and therefore can map the actual evidence accumulator model (DDM) parameters onto the declarative memory parameters within ACT-R (van der Velde et al., 2021). The total time required to retrieve the winning chunk c with activation  $A_c$  within ACT-R is defined by the equation below. Included in the equation is the latency factor F.

$$RT_c = Fe^{-A_c} + t_{er} \tag{2}$$

Over a trial average, this equation can be rewritten to derive the expected time for retrieval across a series of trials using the average latency factor  $\overline{F}$  and average activation  $A_{c}$ :

$$E(RT_c) = \frac{\bar{F}}{e^{A_c}} + t_{er} \qquad (3)$$

As the DDM assumes evidence accumulation at a constant rate, the expected time for accumulator c to reach the upper decisional threshold a is dependent on the decisional starting point z and drift rate v with a scaling factor  $t_0$ (Bogacz et al., 2006).

$$E(RT_c) = \frac{a-z}{v} + t_0 \tag{4}$$

The DDM is different from other evidence accumulator models in which there are two separate accumulation processes occurring for each choice (or chunk) as DDM incorporates the difference of the two possible decisions into the evidence accumulation process (Bogacz et al., 2006). In DDM, the probability P of accumulator c with drift rate v of reaching the upper decisional threshold a is defined by the equation below.

$$P_c = \frac{1}{1 + e^{-2\nu}}$$
(5)

This equation is reminiscent of the probability of receiving a certain chunk over a competitor in ACT-R: The probability P of retrieving chunk c with activation  $A_c$  over a foil f with activation  $A_f$  can be represented by the equation below.

$$P_{c} = \frac{e^{A_{c}}}{e^{A_{c}} + e^{A_{f}}} = \frac{1}{1 + e^{A_{c} - A_{f}}} \quad (6)$$

Using the above equations, we can then map ACT-R parameters onto those outputted by DDM (Figure 1B).  $\overline{F}$  in ACT-R (latency factor) is related to the relationship between the upper decisional threshold and the decisional starting point or bias in DDM.

$$\overline{F} = a - z \tag{7}$$

Drift rate v within DDM is related to the difference between the activations of the competing chunks within ACT-R. Here, we adapted the equation to reflect the difference of the average activations of competing chunks c and f represented by  $\Delta A$ .

$$\Delta A = -2\nu \qquad (8)$$

Similar to previous work, we see a direct equivalency of the extradecisional component within an evidence accumulator model of DDM and that within ACT-R (van der Velde et al., 2021).



Figure 1: (A) Illustration of the diffusion model with the four main parameters (a, z, v, and  $t_0$ ) with three exemplary trials (in blue). (B) The same model depiction but with the equivalency of ACT-R parameters using equations (7)-(9).

#### **Simulation: Recovering ACT-R Parameters**

#### **Materials and Methods**

Data. The data used in this analysis was simulated using ACT-R with code adapted from van der Velde et al. 2021. ACT-R was used to simulate 25 model participants undergoing a declarative memory retrieval task with two competing chunks, c and f. Time to reach the decision boundary of the winning chunk was recorded in seconds (referred to as response time). A "correct" trial was indicated when chunk c was the first accumulator to reach the decision boundary. Overall DDM fit can be affected by outlier reaction times (Lerche et al., 2017) at lower trial numbers so an IQR outlier correction was applied to simulated data prior to model fitting. Simulations were repeated with a varying number of trials per model participant, ranging from 25 to 5000 to best understand the minimum trial size needed for accurate parameter recovery.

**Model Fitting.** The DDM was fitted individually to each model's simulated response/accuracy distributions using the *ddiffusion* density function within the *rtdists* package in R (R version 3.2.0; *rtdists* 0.8-3). For each model participant, we used DDM to estimate parameters *a*, *z*, *v*, and *t*<sub>0</sub>. We likewise allowed the model to fluctuate on an inter-trial basis by including inter-trial variability parameters that account for changes in *t*<sub>0</sub>, *z*, and *v* from trial-to-trial (variability parameters: *st*<sub>0</sub>, *sz*, *sv*). These parameters have been shown to help with DDM fit to the empirical distribution and improve accuracy of parameters ( $\Delta A, \overline{F}, T_{er}$ ) were recalculated using the equations (7)-(9) previously described.

#### Results

To understand the optimal trial size for consistent ACT-R parameter recovery we attempted the parameter recovery simulation at varying trial sizes from 25 trials per participant to 5000. Across all trial sizes we calculated absolute error and correlations across the recovered parameters:  $\overline{F}$ ,  $T_{er}$  and  $\Delta A$ . Notably, we saw comparable absolute errors and correlations of original vs. recovered parameters at trial sizes of 100 trials per simulated participant or greater (Figure 2;  $T_{er}$  at 100 trials per participant: r = 0.97,  $T_{er}$  at 5000 trials per participant: r = 0.99;  $\overline{F}$  at 100 trials per participant: r = 0.46,  $\overline{F}$  at 5000 trials per participant: r = 0.48).



Figure 2: Absolute error (A) and Pearson correlation values (B) across trial sizes 25-5000 per participant across the three recovered parameters:  $\Delta A$  (shown in blue),  $\overline{F}$  (shown in orange), and  $T_{er}$  (shown in green). Mean absolute error (A) or mean correlation (B) across all parameters is shown in black.

With just 100 trials per participant, the original inputted ACT-R parameters showed a fairly linear and consistent recovery with DDM parameter estimation (Figure 3).



Figure 3: Scatter plot of original (x-axis) versus recovered (y-axis) parameter values for 25 model participants with 100 trials per participant for the three recovered parameters:  $\Delta A$  (left: shown in blue),  $\overline{F}$  (center: shown in orange), and  $T_{er}$  (right: shown in green).

We do, however, see larger variability in the recovery of the difference of activation rates ( $\Delta A$ ) with few outlier participants causing large increases in the error observed in the recovery. This effect did not seem to reduce with increased trials per participant (Figure 4).



Figure 4: Scatter plot of original (x-axis) versus recovered (y-axis) parameter values for 25 model participants with 25-5000 trials per participant for the three recovered parameters:  $\Delta A$  (shown in pink),  $\overline{F}$  (shown in green), and  $T_{er}$  (shown in blue).

#### **Parameter Estimation in an Empirical Dataset**

#### **Materials and Methods**

Data. The data used here come from an experiment carried out by Verstynen (2014) and freely available on OpenNeuro (dataset ds000164). Twenty male and ten female participants performed the color-word Stroop task (Botvinick et al., 2001; Gratton et al., 1992; MacLeod, 1991; Stroop, 1935) which consisted of congruent, incongruent, and neutral stimulus conditions. Participants were presented with word-stimuli and were instructed to respond to the color in which the word was printed and to ignore the meaning of the printed word. In a congruent "GREEN", condition, the words "BLUE", and "YELLOW" were displayed in the colors green, blue, and yellow respectively. The incongruent condition showed words whose meaning was a different color than the ink in which the printed word was displayed (i.e., the displayed word was "GREEN" in blue ink). In neutral conditions, a non-color word was presented in an ink color (i.e., the word "HAT" printed in blue ink). Participants responded by pressing different buttons, with different right-hand fingers, for each color (e.g., red: index; green: middle; and yellow: ring finger). Each participant completed 120 trials (42 congruent, 42 neutral, 36 incongruent). Trial types and stimuli types were pseudorandomized in an event-related fashion. Response time and accuracy were recorded for each trial. The data was collected as part of a larger study and more information of the participants and procedure can be found in Verstynen (2014).

**Model Fitting.** The DDM was fitted to each participant's response-accuracy distribution separately. To optimize computing speed and for added statistical rigor, DDM was fitted using the *Fast-dm-30.2* toolbox (Voss & Voss, 2007). Each participant's parameter optimization was statistically verified using the Kolmogorov-Smirnov method. Similar to the simulation experiments, inter-trial

variability parameters ( $st_0, sz, sv$ ) were allowed to fluctuate across trials during the parameter estimation process to optimize DDM fit. ACT-R parameters ( $\Delta A, \overline{F}, T_{er}$ ) and were again recalculated from the outputted DDM parameters (v, z, a, and  $t_0$ ) using equations (7)-(9). DDM density plots were created using the *ddiffusion* density function within the *rtdists* toolbox in R (R version 3.2.0; *rtdists* 0.8-3).

ACT-R Stroop Task. A simple model of the Stroop task was implemented to test the possibility of translating DDM parameters directly into ACT-R models. This model borrows the central idea of previous models of response interference in the Stroop (Lovett, 2002) and Simon tasks (Stocco et al., 2017) and captures the Stroop effect as interference in the color name retrieval due to competing sources of activation. Specifically, the model responds by initially focusing on the word's color. While attending to the color, the model attempts to retrieve an associated color name. This retrieval process is aided by activation spreading from the attended color to the corresponding name (e.g., from the color green to the word "green"), which confers an additional boost of activation to the correct color name over the equally active names of other colors. Once a color name is retrieved, a production rule performs the corresponding motor response. The simplicity of this model makes the DDM parameters immediately translatable. Specifically, the difference in mean activation between competing chunks  $(\Delta A)$ corresponds to the contribution of spreading activation from the word's color, and the  $T_{er}$  parameter corresponds to the duration of motor execution (the "motor burst time" parameter) once the visual encoding time (fixed and maintained at its default value of 50ms) and the execution time of the necessary productions (three productions for 50ms each) are accounted for.

Note that although  $T_{er}$  by definition represents time components split across both the visual encoding and motor module, functionally it does not make a difference which of these  $T_{er}$  is incorporated into as regardless it will be added on to overall reaction time. We ran individualized ACT-R models with these inputted parameters for each of the participants with the same number of trials as in the empirical study (42 congruent, 42 neutral, and 36 incongruent).

#### Results

We fit DDM to each participant's data individually. Across all participants, we observed a reasonable fit of DDM to the empirical distribution which was further verified through the Kolmogorov-Smirnov test statistic (p = [0.83-0.99] across all stimulus types). This provided reassurance that the outputted DDM parameters were reasonably estimated and could be used for subsequent ACT-R parameter recovery. Excitingly, we were able to estimate reasonable ACT-R parameters:  $\overline{F}$ ,  $T_{er}$ , and difference of activation rates between the competing

chunks ( $\Delta A$ ). Although we observed moderate variability across subjects and condition types,  $\overline{F}$  (across all conditions  $\overline{F} = 0.64 \pm 0.13$ ),  $T_{er}$  (across all conditions  $T_{er}$ = 0.61 ± 0.07),  $\Delta A$  (across all conditions  $\Delta A = 6.13 \pm$ 1.74) and were within typical ranges according to previous ACT-R studies (Anderson et. al, 1998).

We were further interested to see if ACT-R simulated data of a Stroop task that utilizes these estimated parameters would provide a comparable reaction time/accuracy distribution to the empirical data we originally inputted into the DDM. Across the 28 participants we saw relatively linear recovery of mean reaction time and accuracy across participants (Figure 6A). To ensure these parameters were indeed individualized to the participant and not a factor of task, we randomized the estimated parameters across participants and again compared the recovery of mean reaction time/accuracy across participants. As seen in Figure 6B, this recovery is substantially worse if parameters are not matched to the original participant, providing evidence that this parameter estimation method is sensitive to and sustains individual differences in its integration into ACT-R.

A. DDM-Derived Parameters B. Scrambled Parameters



Figure 6: Mean accuracy and reaction time of the original empirical subjects (x-axis) versus the ACT-R simulated data (y-axis) with the DDM-Derived participant-specific ACT-R parameters inputted (A) versus if the DDM-Derived ACT-R parameters are randomized across different subjects (B).

#### Discussion

In this paper, we have presented evidence of an ability to integrate DDM parameters into the ACT-R parameter estimation process. Across trial sizes as low as 100 trials per model participant, we observed a fairly consistent and linear recovery of the extradecisional time component  $T_{er}$ , the latency factor  $\overline{F}$ , and difference of activation rates between the top two competing chunks  $\Delta A$ , within a simulated declarative memory retrieval task. Both  $T_{er}$  and  $\overline{F}$  showed a relatively consistent increase in correlation and decrease in observed absolute error as trial sizes increased from 25-5000 trials per participant. Interestingly, in observing the recovery of  $\Delta A$ , we observed a "zig-zag" pattern in correlation and observed absolute error as trial sizes increased instead of the steady increase in recovery correlation/decrease in absolute error as observed with the other parameters. We expect this is due to the presence of 1-4 simulated participants within each simulation in which the estimated DDM drift rate (v)was very high due to the presence of numerous trials with extremely short simulated reaction times (<200ms). As our simulated reaction time/accuracy distributions were drawn from random distributions, the presence of model participants with trials like this were randomly observed, which caused the odd pattern of recovery (i.e., seemingly better observed absolute error in trial sizes of 50 compared to 100 trials per participant). In use with empirical data and non-simulated participants, this becomes less of an issue as extremely short reaction times are typically removed by way of outlier correction prior to model fitting. However, to not only reduce the presence of these apparent outliers but similarly increase the statistical rigor of the DDM parameter estimation, we plan to integrate an optimizer function into the process of fitting the DDM to the original dataset. From there, one could choose the set of parameters with an optimized fit before mapping to ACT-R parameters. One could similarly utilize existing software such as the Fast-dm-30.2 toolbox (Voss & Voss, 2007) which incorporates optimization methods without added burden to the user.

To further emulate this method's applicability, we utilized this DDM-ACT-R parameter estimation method on an empirical data set of a Stroop task (Verstynen, 2014). We demonstrated that by using DDM-derived parameters, we were able to estimate ACT-R parameters within typical ranges according to prior studies. Most excitingly, when we integrated these DDM-derived parameters into an ACT-R simulated Stroop model, we were able to accurately recreate the reaction time/accuracy distributions observed within the empirical dataset as shown by comparing empirical versus recovered mean reaction time and accuracies. Notably, these parameters seemed to be individualized to the participant, as randomization of these parameters showed a worse recovery of empirical mean reaction time and accuracy across participants. Further comparison experiments are needed to understand whether DDM-ACT-R parameter estimation method is indeed more accurate/individualized compared to common parameter estimation methods such as parameter grid searches or sweeps, although this DDM-ACT-R method has been shown to be quicker and less computationally expensive in this application.

While this method has shown promising results in optimizing incorporating empirical data into a simulated model, the Stroop ACT-R model we used is significantly simplified compared to existing models of this task that have been based on the neurocognitive properties this task elicits (Lovett, 2002; Stocco et al., 2017). In applications confined to a declarative memory task, we are hopeful that this method will be relevant beyond binary decision tasks to multi-alternative decisions, again increasing the usability of drift diffusion models. However, future work utilizing this method outside of the scope of a declarative memory task (i.e., one that relies on procedural complexity) is needed to understand the breadth of its applicability.

Individualized, consistent and accurate estimation of ACT-R parameters with this method, even on simple tasks, would allow us to have a proxy measure for task neural dynamics in datasets that only have behavioral data, greatly reducing the need for expensive and time-consuming fMRI data collection. The integration of DDM into ACT-R can further give neural context to the parameters used in DDM, an application of DDM that has been inconsistent in previous work (Gupta et al., 2022).

In summary, we have exhibited a clear integration of the drift diffusion model into the cognitive architecture of ACT-R. This relationship contributes to a larger effort in optimizing the utilization of empirical data in informing cognitive models as well as in the overall integration of modeling methods.

#### References

- Anderson, J. R. (2007). How Can the Human Mind Occur in the Physical Universe? Oxford University Press.https://doi.org/10.1093/acprof:oso/97801953242 59.001.0001
- 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. https://doi.org/10.1037/0033-295X.111.4.1036
- Anderson, R. C. (1974). Substance Recall of Sentences. *Quarterly Journal of Experimental Psychology*, 26(3), 530–541. https://doi.org/10.1080/14640747408400443
- Bogacz, R., Brown, E., Moehlis, J., Holmes, P., & Cohen, J. D. (2006). The physics of optimal decision making: A formal analysis of models of performance in two-alternative forced-choice tasks. *Psychological Review*, *113*(4), 700–765. https://doi.org/10.1037/0033-295X.113.4.700
- Bond, K., Dunovan, K., & Verstynen, T. (2018). Valueconflict and volatility influence distinct decisionmaking processes. 2018 Conference on Cognitive Computational Neuroscience. 2018 Conference on Cognitive Computational Neuroscience, Philadelphia, Pennsylvania, USA.

https://doi.org/10.32470/CCN.2018.1068-0

Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S., & Cohen, J. D. (2001). Conflict monitoring and cognitive control. *Psychological Review*, *108*(3), 624–652. https://doi.org/10.1037/0033-295X.108.3.624

Gratton, G., Coles, M. G. H., & Donchin, E. (1992). Optimizing the use of information: Strategic control of activation of responses. *Journal of Experimental Psychology: General*, *121*(4), 480–506. https://doi.org/10.1037/0096-3445.121.4.480

Gupta, A., Bansal, R., Alashwal, H., Kacar, A. S., Balci, F., & Moustafa, A. A. (2022). Neural Substrates of the Drift-Diffusion Model in Brain Disorders. *Frontiers in Computational Neuroscience*, *15*, 678232. https://doi.org/10.3389/fncom.2021.678232

- Kotseruba, I., & Tsotsos, J. K. (2020). 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
- Lerche, V., & Voss, A. (2016). Model Complexity in Diffusion Modeling: Benefits of Making the Model More Parsimonious. *Frontiers in Psychology*, 7. https://doi.org/10.3389/fpsyg.2016.01324
- Lerche, V., Voss, A., & Nagler, M. (2017). How many trials are required for parameter estimation in diffusion modeling? A comparison of different optimization criteria. *Behavior Research Methods*, *49*(2), 513–537. https://doi.org/10.3758/s13428-016-0740-2
- Lovett, M. C. (2002). Modeling selective attention: Not just another model of Stroop (NJAMOS). *Cognitive Systems Research*, *3*(1), 67–76. https://doi.org/10.1016/S1389-0417(01)00045-6
- MacLeod, C. M. (1991). Half a century of research on the Stroop effect: An integrative review. *Psychological Bulletin*, 109(2), 163–203.
- https://doi.org/10.1037/0033-2909.109.2.163 Mulder, M. J., Keuken, M. C., van Maanen, L., Boekel, W., Forstmann, B. U., & Wagenmakers, E.-J. (2013). The speed and accuracy of perceptual decisions in a random-tone pitch task. *Attention, Perception, & Psychophysics*, 75(5), 1048–1058. https://doi.org/10.3758/s13414-013-0447-8
- Pavlik, 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. https://doi.org/10.1207/s15516709cog0000 14
- Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, 85(2), 59–108. https://doi.org/10.1037/0033-295X.85.2.59
- Ratcliff, R., & Frank, M. J. (2012). Reinforcement-Based Decision Making in Corticostriatal Circuits: Mutual Constraints by Neurocomputational and Diffusion Models. *Neural Computation*, 24(5), 1186–1229. https://doi.org/10.1162/NECO\_a\_00270
- Ratcliff, R., & McKoon, G. (2008). The Diffusion Decision Model: Theory and Data for Two-Choice Decision Tasks. *Neural Computation*, 20(4), 873–922. https://doi.org/10.1162/neco.2008.12-06-420
- 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. https://doi.org/10.3758/BF03196302
- Stocco, A., Murray, N. L., Yamasaki, B. L., Renno, T. J., Nguyen, J., & Prat, C. S. (2017). Individual differences in the Simon effect are underpinned by differences in the competitive dynamics in the basal ganglia: An experimental verification and a computational model. *Cognition*, 164, 31–45.
  https://dxi.org/10.1016/j.jamasiti.org/2017.02.001

https://doi.org/10.1016/j.cognition.2017.03.001

Stone, M. (1960). Models for choice-reaction time. *Psychometrika*, 25(3), 251–260. https://doi.org/10.1007/BF02289729

Stroop, J. R. (1935). Studies of interference in serial verbal reactions. *Journal of Experimental Psychology*, *18*(6), 643–662. https://doi.org/10.1037/h0054651

van der Velde, M., Sense, F., Borst, J. P., van Maanen, L., & van Rijn, H. (2021). *Capturing dynamic performance in a cognitive model: Estimating ACT-R memory parameters with the linear ballistic accumulator* [Preprint]. PsyArXiv. https://doi.org/10.31234/osf.io/yg7s6

Verstynen, T. D. (2014). The organization and dynamics of corticostriatal pathways link the medial orbitofrontal cortex to future behavioral responses. *Journal of Neurophysiology*, *112*(10), 2457–2469. https://doi.org/10.1152/jn.00221.2014

Voss, A., Nagler, M., & Lerche, V. (2013). Diffusion Models in Experimental Psychology: A Practical Introduction. *Experimental Psychology*, 60(6), 385– 402. https://doi.org/10.1027/1618-3169/a000218

Voss, A., Rothermund, K., & Voss, J. (2004). Interpreting the parameters of the diffusion model: An empirical validation. *Memory & Cognition*, *32*(7), 1206–1220. https://doi.org/10.3758/BF03196893

Voss, A., & Voss, J. (2007). Fast-dm: A free program for efficient diffusion model analysis. *Behavior Research Methods*, 39(4), 767–775. https://doi.org/10.3758/BF03192967

Wald, A., & Wolfowitz, J. (1948). Optimum Character of the Sequential Probability Ratio Test. *The Annals of Mathematical Statistics*, 19(3), 326–339. JSTOR.

# Learning to Expect Change: Volatility During Early Experience Alters Reward Expectations in a Model of Interval Timing

Nora C. Harhen (nharhen@uci.edu)

Aaron M. Bornstein (aaron.bornstein@uci.edu)

Department of Cognitive Sciences, University of California, Irvine, Irvine, CA 92697 USA

#### Abstract

An unpredictable early life environment can have enduring effects on mental health outcomes in adulthood. Despite widespread evidence for this relationship, it remains unclear what core mechanism links the two. Here we propose that early life unpredictability (ELU) shapes the development of temporal sequence representations. Critically, we show that this in turn produces impairments in reward sensitivity and learning, phenotypes that have been associated with anhedonia, a transdiagnostic symptom often observed in individuals with ELU. We formalize this hypothesis using a principled model of interval timing whose representations adjust with experience to support adaptive temporal predictions. The core observation is that initial unpredictability in timing produces broader, more imprecise temporal expectations. As a result, reward anticipation and learning are diminished. When we introduced agents with broader expectations into a stable environment, they showed a greater response to the omission of reward relative to its presence. This bias accords with negative attentional and mnemonic biases associated with anhedonia. In sum, we show that a single mechanism can explain a range of behaviors associated with anhedonia, offering insights into the role of temporal representations in reward learning and in the emergence of phenotypes linked to psychiatric disorders.

**Keywords:** early life unpredictability; reinforcement learning; interval timing; temporal representation

#### Introduction

Across development, brain circuits adapt to meet the demands of the environment. Concretely, sensory receptive fields are tuned to reflect the statistics of the early life environment, determining perceptual discrimination abilities in adulthood. Consistency is crucial to this maturation process. For functional circuits to form, the input statistics must be consistent (Li, Fitzpatrick, & White, 2006). It has recently been proposed that similar processes may occur in reinforcement and memory systems critically involved in associative learning (Birnie et al., 2020). This implies that the consistency or predictability of associations encountered early in life may shape the acquisition of associations later on.

Interactions with caregivers are one contributor to the associative statistics an infant encounters. For example, the infant behaves in some way and, normatively, the caregiver produces a consistent response to this behavior such that the infant can anticipate the response in the future. The timing between behavior and response is encoded and can be represented using a set of temporal receptive fields (TRFs) similar to receptive fields found in sensory areas. Instead of being tuned to visual angle or auditory pitch, these TRFs are sensitive to the time between associated stimuli and its consistency.

Caregivers vary in the valence and predictability of their responses. Most prior work has focused on the effect of valence on later child mental health outcomes. However, recent work has begun to examine how early life unpredictability, or ELU, might also contribute (Baram et al., 2012). Caregiver signals, if unpredictable, can result in anhedonia-like behaviors such as reduced experience of pleasure and motivation (Bolton et al., 2018). Importantly, anhedonia is a transdiagnostic symptom associated with several psychiatric disorders previously shown to be related to ELU (Glynn et al., 2019).

In the current work, we propose that TRFs are tuned to the unpredictability of timing in the environment, and these adaptations produce an anhedonic phenotype. We extend a principled computational model of interval timing (Ludvig, Sutton, & Kehoe, 2008) to examine how enhanced volatility during an early period of plasticity can, with minimal assumptions, affect later predictions of reward during maturity, when adaptation no longer occurs. With this model, we formally demonstrate that early unpredictability in timing and adaptation of temporal receptive fields to this timing can lead to an array of anhedonia-like symptoms. This includes an asymmetric response to reinforcement and omission despite no differences in the overall amount of reinforcement. This reproduces empirical findings that poor mental health outcomes can emerge from unpredictability in early life experience beyond what would be predicted from the overall number of adverse events (Glynn et al., 2019).

#### Methods

### **The Temporal-Difference model**

Temporal-Difference (TD) models aim to accurately estimate the value of states in the world, V, in terms of the future rewards they predict. Time is explicitly represented in these models with a separate V for each time step, t, in a trial.

$$V^* = E[\sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k}]$$
 (1)

where  $r_t$  is the reward received at the current time step, and  $\gamma$  controls how heavily future rewards are discounted. Future rewards are less influential on *V* when  $\gamma$  is low. A TD agent



Figure 1: Two groups of agents, early life unpredictability (ELU) and control, learned to associate a cue and reward across two environments. The cue was partially reinforced in both environments — 75% of the time in the first and 50% in the second. During the first phase, both groups adapted their temporal receptive fields to the statistics of reward timing. The timing of reward delivery varied from trial to trial, differently for each group: The ELU group's timing was sampled from a much wider distribution relative to the controls. However, during the second phase, both groups received rewards at the exact same time on every reinforced trials.

learns *V* by an error driven learning rule. The estimate of *V* at the next time step is updated using the difference,  $\delta_t$ , between the reward that was predicted  $(V_{t-1})$  and what was actually received  $(r_t + \gamma V_t)$ .

$$\delta_t = r_t + \gamma V_t - V_{t-1} \tag{2}$$

#### The Microstimulus model

All TD models explicitly represent time, but do so in various ways. Basic TD models use a complete-serial-compound (CSC) representation in which each time step is treated as independent from one another. The agent is assumed to have perfect knowledge of the time between cue and reward. This representation prohibits temporal generalization, creating issues in environments where the time between cue and reward varies. The microstimulus representation addresses this problem by relaxing its temporal markers (Ludvig et al., 2008). CSC's discrete markers are replaced with less precise microstimuli that allow for uncertainty to be represented. A stimulus, whether it be a neutral, rewarding, or aversive is assumed to leave behind a memory trace that decays with time. The trace is represented by a basis set of overlapping temporal receptive fields — Gaussian distributions whose standard deviations increase with the time after onset of the initial stimulus.

$$f(y,\mu,\sigma) = \frac{1}{\sqrt{2}\pi} e^{\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)}$$
(3)

A time step's value,  $V_t$ , is estimated as the weighted average of the microstimuli.

$$V_t = w_t^T x_t = \sum_{i=1}^n w_t(i) x_t(i)$$
(4)

This value is compared to the reward received. The error term,  $\delta_t$ , adjusts the weights on the microstimuli, consequently updating the predicted value at the next time step.

$$w_{t+1} = w_t + \alpha \delta_t e_t \tag{5}$$

 $\alpha$  is the learning rate controlling the time window over which trial to trial experiences are integrated.  $e_t$  is a vector containing each stimulus's eligibility traces.

$$e_t = \gamma \lambda e_t + x_t \tag{6}$$

Following the stimulus, its eligibility trace decays at a rate determined by  $\gamma$  and  $\lambda$ .  $\gamma$  is a discounting factor as above while  $\lambda$  controls the time window over which a stimulus can induce learning within a trial. For all simulations, we use the parameter settings from Ludvig et al, 2008 —  $\alpha = 0.01$ ,  $\gamma = 0.98$ ,  $\lambda = 0.95$ , n = 50, and  $\sigma = 0.08$ .

#### Simulating development

To model developmental changes in learning, we limit the period over which microstimuli weights can adapt to experience. We treat this as a critical period during which the temporal receptive fields are tuned to support accurate estimation of V. This adaptation process is designed to mimic the observed tuning of sensory receptive fields during analogous sensitive periods of development (Simoncelli & Olshausen, 2001).

We simulated two groups of agents learning cue-reward pairings across two phases (Figure 1). One group of agents, the early life unpredictability or ELU group, experienced a volatile environment in the first phase. Specifically, the delay between cue and reward considerably varied from trial to trial. The other group of agents, the control group, experienced relatively much less variation.

On each of the 1000 simulated trials, a cue was always presented at 100 milliseconds and there was a 75% probability of a reward following it. If a cue was reinforced on a trial, the timing of reward delivery was sampled from a normal distribution with  $\mu$  set to 300 milliseconds for all agents while  $\sigma$  varied. For the ELU group,  $\sigma$  was sampled from a zero-truncated normal distribution with  $\mu_{hyper,elu} = 10$  and  $\sigma_{hyper,elu} = 3$ . The control group experienced much less temporal variability with  $\sigma$  being sampled from a zero-truncated normal distribution with  $\mu_{hyper,control} = 1$ ,  $\sigma_{hyper,control} = 2$ .

In the second phase, the weights could no longer adapt to the new environment. Thus, this phase was post the critical period. Both groups encountered another 1000 trials of learning to pair the same cue to a reward. On each trial, there was now a 50% probability of reward being presented following



Figure 2: **A.** Temporal receptive field (TRF) imprecision was computed by taking a weighted average of the standard deviations of the temporal receptive fields following the critical period phase. The ELU group showed greater average temporal receptive field imprecision, a consequence of their more volatile experience during the critical period. **B.** Both groups' positively weighted temporal receptive fields. Recapitulating the results shown in panel A, the ELU group relied on more broadly tuned, less precise temporal receptive fields relative to the control group.

the cue. As before, the cue arrived at 100 ms. Reward timing was more stable in this environment with reward always arriving at 500 ms.

Here we focus on anhedonia, variously defined as the inability to experience and/or anticipate pleasure, as a symptom associated with many disorders observed to result following ELU. Following previous work, we model anhedonia as a reduced sensitivity to rewards and an impaired ability to learn from reinforcement (Huys, Pizzagalli, Bogdan, & Dayan, 2013). We asked if the simulated agents could exhibit these features of anhedonia from variability in reward timing alone, despite outcome valence being equated across groups.

#### Results

#### **Critical Period**

First, we examined how the initial environment shaped the tuning of temporal receptive fields by comparing the groups' microstimuli weights following the critical period. For each agent, we computed a temporal precision measure by taking a weighted average of the microstimuli's standard deviations. We found that the ELU group relied on more broadly-tuned receptive fields relative to the controls (Figure 2; t(198) = -7.83, p < .0001).

Prior work has demonstrated that early life unpredictability impedes learning from reinforcement (Birn, Roeber, & Pollak, 2017; Dillon et al., 2009). Thus, we examined whether the model could capture this. As a proxy for learning, we use prediction error magnitude. The more an agent has learned to associate a cue and a reward, the smaller their prediction error will be when a cue is reinforced with reward and the greater their prediction error will be when reward is omitted. To compare prediction errors between groups, we computed the median prediction error extremum for each agent. On reinforced trials, the maximum prediction error magnitude following the cue was taken while on omission trials, we took the minimum. We found that the ELU group demonstrated more extreme prediction errors relative to controls on reinforced trials (Figure 3, 4, t(198) = 15.15, p < .0001) but less extreme on omission trials (t(198) = 6.09, p < .0001). These results are consistent with the ELU group showing weaker learning under reinforcement. Critically, this is despite experiencing the same amount of reward on average as the control group (t(198)= 0.67, p = 0.51). This suggests that impaired reward learning, as observed in anhedonia, can emerge from experienced temporal volatility alone during a period of plasticity.

Early life unpredictability has also been shown to impair motivation (Hanson, Williams, Bangasser, & Peña, 2021). This may stem from a reduced expectation of reward. Thus, we compared the groups' expectation of value across time following the cue. The ELU group's value signal peaked early following the cue (mean = 156 ms; sd = 27) and slowly decayed, not reaching its minimum for several 100s of milliseconds following the cue (mean = 500 ms; sd = 1.4). This suggests if the reward is not received immediately, ELU individuals gradually grow less confident it will come at all. Conversely, the control group's signal peaked much later (mean = 265; sd = 19; t(198) = 32.66, p < .0001) but reached its minimum much sooner near the average reward time (Figure 5 mean = 373; sd = 89; t(198) = -15.45, p < .0001). In other words, control individuals increasingly anticipate the reward as its expected arrival time approaches.

#### **Post Critical Period**

During the second phase, the reward timing was consistent for both groups and the weights were no longer allowed to adapt. Under these conditions, the ELU group showed less extreme positive prediction errors relative to controls (Figure 6, t(198) = -14.57, p < .0001) but more extreme negative prediction errors (t(198) = -8.13, p < .0001), the opposite pattern as observed during the critical period.

To ensure our simulated agents' bias did not emerge from aggregating over the data, we computed an asymmetry index for each agent:

$$index = \frac{PE_+ - PE_-}{PE_+ + PE_-} \tag{7}$$

We found that the the ELU group had asymmetry indices that were in aggregate negative (t(199) = -2.87, p = .005)



Figure 3: Critical period results - prediction error. Prediction error,  $\delta$ , across time on trials where the cue was reinforced versus when it was omitted. For the ELU group, the timing of the large prediction error following the cue varies from trial to trial as a result of the reward timing varying. In contrast, the control group consistently experience a large prediction error near 300 ms.



Figure 4: Critical period results - median prediction error extremum. For each trial, the extreme points of the prediction error was taken following the cue. For each agent, the measure was computed by taking the median over the trials' extremums. The ELU group showed larger predictions errors on trials where the cue was reinforced but weaker prediction errors when reward was omitted following the cue. Stars indicate significance of the test reported in the main text as follows: \* p < .05, \*\* p < .01, \*\*\* p < .001.



Figure 5: Critical period results - value. V, at each time step averaged across trials. The ELU group's value decreased following the cue while the control group's increased. Once the typical reward time was reached, the ELUs' value signal continued to steadily drop while the controls' did so quickly.



Figure 6: Post critical period results. **A.** Prediction errors at each time step for reinforced and omitted trials. On trials where the cue was reinforced, the control group showed larger and earlier cue-related and reward-related prediction errors relative to the ELU. On trials where reward was omitted, again, the control group showed larger and earlier cue-related prediction errors. However, for the reward-related prediction errors, the ELU groups' were larger. **B.**Asymmetry index. The ELU group displayed more extreme prediction errors on omission trials relative to reinforced while the control group showed the opposite pattern.

while the control group's were positive (t(199) = 7.00, p < .0001).

#### Discussion

Here, we've proposed a novel computational link between early life unpredictability and the emergence of anhedonia — the optimization of temporal representations to the early life environment. We assume that the volatility of the early life environment adaptively tunes temporal receptive fields in such a way that several behaviors associated with anhedonia — impaired learning from reinforcement reduced anticipation of reward, and a greater response to the omission of events — emerge.

These findings are consistent with behavioral outcomes observed in the laboratory and clinical settings. One representative such set of findings is of an asymmetric attentional bias in anhedonia. If we assume that attention increases with prediction error magnitude, then the ELU group were attentionally biased toward the omission outcome over the reinforced. Additionally, if we treat the omission of reward as a negatively valenced event and the presence of reward as positive, this suggests a negative attentional bias in the ELU group and positive bias in the controls, reproducing empirical findings (Dillon & Pizzagalli, 2018; Frank, 2004). Larger negative prediction errors may not only affect attention in the moment but also shape mood over the longer term (Eldar, Rutledge, Dolan, & Niv, 2016). Recurring negative prediction errors may give rise to the persistent negative mood that characterizes anhedonia (Dillon et al., 2009).

In the current work, we've interpreted the results while treating the outcome paired with the cue as a reward. However, the model is agnostic to whether the associated stimuli are neutral, rewarding, or aversive. Different outcome valences suggest different behavioral phenotypes. If the outcome is aversive, like a shock, rather than a reward, the ELU group's prolonged expectation of an outcome's appearance could produce a sort of "paranoia". The agent generalizes their expectation of the aversive event over a longer time period, producing a continual state of nervousness that aligns with symptoms of anxiety. If the outcome is neutral, impairments in reward learning become more general impairments in relational learning. This may explain memory deficits and alterations in hippocampal structure in ELU individuals (Granger et al., 2021; Molet et al., 2016) and its relationship with anhedonia. Prior work has suggested that anhedonia is characterized not only by the inability to experience pleasure in the moment but also the inability to recall past and anticipate future pleasurable experiences (Dillon & Pizzagalli, 2018).

Here we've only considered the mechanism under Pavlovian learning conditions. However, it suggests differences in ELU individuals' instrumental learning and action selection. The inability to accurately predict the timing of future outcomes diminishes an individual's perceived controllability of the environment, which has been implicated in psychiatric disorders such as anxiety (Bishop & Gagne, 2018).

Hidden-state inference models capture a similar idea as the microstimulus model at a different level of analysis (Starkweather, Babayan, Uchida, & Gershman, 2017). Often, the true state of the world is unknown or hidden and must be inferred from observations. This inference process is in part driven by prediction errors (Rouhani, Norman, Niv, & Bornstein, 2020), and by extension is more difficult in volatile environments. As a result, ELU individuals may infer fewer states in the world (or, analogously, more states in an environment where negative prediction errors predominate) and group their experiences accordingly as a result of this early volatility. We have previously shown that this assumption of reduced sensitivity with a hidden-state inference model can produce reduced exploration in a foraging task (N. C. Harhen & Bornstein, 2021), a behavior found in ELU populations (Lloyd, McKay, & Furl, 2022), and may also explain why individuals who experience early life unpredictability are at higher risk of developing substance use disorders and relapsing following treatment (N. Harhen, Baram, Yassa, & Bornstein, 2021).

Our results highlight the key role time plays in shaping reinforcement learning and consequently its impact on behaviors associated with mental illness. The varied phenotypes that emerge from the same computations is consistent with the idea that the mechanism identified here has implications that extend beyond anhedonia. It suggests a common origin for a number of psychiatric disorders, potentially explaining their high co-morbidity rates. Further empirical research is needed to test the model's behavioral implications for early life unpredictability's impact on interval timing, and interval timing's relationship with psychiatric disorders.

# Acknowledgements

This work was supported by a NARSAD Young Investigator Award from the Brain and Behavior Research Foundation to AMB. NCH was supported by a National Defense Science and Engineering Graduate fellowship. The authors thank Elliot Ludvig for providing the microstimulus model code, and Tallie Z Baram, Michael A Yassa, Steven J Granger, Steven V Mahler, and Sophia C Levis for helpful conversations.

#### References

- Baram, T. Z., Davis, E. P., Obenaus, A., Sandman, C. A., Small, S. L., Solodkin, A., & Stern, H. (2012, September). Fragmentation and unpredictability of early-life experience in mental disorders. *Am. J. Psychiatry*, 169(9), 907–915.
- Birn, R. M., Roeber, B. J., & Pollak, S. D. (2017, December). Early childhood stress exposure, reward pathways, and adult decision making. *Proc. Natl. Acad. Sci. U. S. A.*, 114(51), 13549–13554.
- Birnie, M. T., Kooiker, C. L., Short, A. K., Bolton, J. L., Chen, Y., & Baram, T. Z. (2020, May). Plasticity of the reward circuitry after Early-Life adversity: Mechanisms and significance. *Biol. Psychiatry*, 87(10), 875–884.
- Bishop, S. J., & Gagne, C. (2018, July). Anxiety, depression, and decision making: A computational perspective. *Annu. Rev. Neurosci.*, *41*, 371–388.
- Bolton, J. L., Molet, J., Regev, L., Chen, Y., Rismanchi, N., Haddad, E., ... Baram, T. Z. (2018, January). Anhedonia following Early-Life adversity involves aberrant interaction of reward and anxiety circuits and is reversed by partial silencing of amygdala Corticotropin-Releasing hormone gene. *Biol. Psychiatry*, 83(2), 137–147.
- Dillon, D. G., Holmes, A. J., Birk, J. L., Brooks, N., Lyons-Ruth, K., & Pizzagalli, D. A. (2009, August). Childhood adversity is associated with left basal ganglia dysfunction during reward anticipation in adulthood. *Biol. Psychiatry*, 66(3), 206–213.
- Dillon, D. G., & Pizzagalli, D. A. (2018, March). Mechanisms of memory disruption in depression. *Trends Neurosci.*, 41(3), 137–149.
- Eldar, E., Rutledge, R. B., Dolan, R. J., & Niv, Y. (2016, January). Mood as representation of momentum. *Trends Cogn. Sci.*, 20(1), 15–24.
- Frank, M. J. (2004). Dynamic dopamine modulation of striato-cortical circuits in cognition: Converging neu-

*ropsychological, psychopharmacological and computational studies* Unpublished doctoral dissertation. Ann Arbor, United StatesUniversity of Colorado at Boulder.

- Glynn, L. M., Stern, H. S., Howland, M. A., Risbrough, V. B., Baker, D. G., Nievergelt, C. M., ... Davis, E. P. (2019, April). Measuring novel antecedents of mental illness: the questionnaire of unpredictability in childhood. *Neuropsychopharmacology*, 44(5), 876–882.
- Granger, S. J., Glynn, L. M., Sandman, C. A., Small, S. L., Obenaus, A., Keator, D. B., ... Davis, E. P. (2021, February). Aberrant maturation of the uncinate fasciculus follows exposure to unpredictable patterns of maternal signals. J. Neurosci., 41(6), 1242–1250.
- Hanson, J. L., Williams, A. V., Bangasser, D. A., & Peña, C. J. (2021). Impact of early life stress on reward circuit function and regulation. *Front. Psychiatry*, 12, 1799.
- Harhen, N., Baram, T. Z., Yassa, M. A., & Bornstein, A. M. (2021, May). Formalizing the relationship between early life adversity and addiction vulnerability: The role of memory sampling. *Biol. Psychiatry*, 89(9), S189.
- Harhen, N. C., & Bornstein, A. M. (2021). Structure learning as a mechanism of overharvesting. *Proceedings of the 19th International Conference on Cognitive Modeling*.
- Huys, Q. J., Pizzagalli, D. A., Bogdan, R., & Dayan, P. (2013, June). Mapping anhedonia onto reinforcement learning: a behavioural meta-analysis. *Biol. Mood Anxiety Disord.*, 3(1), 12.
- Li, Y., Fitzpatrick, D., & White, L. E. (2006, May). The development of direction selectivity in ferret visual cortex requires early visual experience. *Nat. Neurosci.*, 9(5), 676– 681.
- Lloyd, A., McKay, R. T., & Furl, N. (2022, January). Individuals with adverse childhood experiences explore less and underweight reward feedback. *Proc. Natl. Acad. Sci.* U. S. A., 119(4).
- Ludvig, E. A., Sutton, R. S., & Kehoe, E. J. (2008, December). Stimulus representation and the timing of reward-prediction errors in models of the dopamine system. *Neural Comput.*, 20(12), 3034–3054.
- Molet, J., Maras, P. M., Kinney-Lang, E., Harris, N. G., Rashid, F., Ivy, A. S., ... Baram, T. Z. (2016, December). MRI uncovers disrupted hippocampal microstructure that underlies memory impairments after early-life adversity. *Hippocampus*, 26(12), 1618–1632.
- Rouhani, N., Norman, K. A., Niv, Y., & Bornstein, A. M. (2020, October). Reward prediction errors create event boundaries in memory. *Cognition*, 203, 104269.
- Simoncelli, E. P., & Olshausen, B. A. (2001). Natural image statistics and neural representation. *Annu. Rev. Neurosci.*, 24, 1193–1216.
- Starkweather, C. K., Babayan, B. M., Uchida, N., & Gershman, S. J. (2017, April). Dopamine reward prediction errors reflect hidden-state inference across time. *Nat. Neurosci.*, 20(4), 581–589.

# Multitasking while Driving: Central Bottleneck or Problem State Interference?

# Moritz Held<sup>1,2</sup> (moritz.held@uol.de), Jochem Rieger<sup>1</sup> (jochem.rieger@uol.de), Jelmer Borst<sup>2</sup> (j.p.borst@rug.nl)

<sup>1</sup>Dept. of Applied Psychology, University of Oldenburg, Germany

<sup>2</sup>Bernoulli Institute, Dept. of Artificial Intelligence, University of Groningen, the Netherlands

**Keywords:** ACT-R; driving; working memory

#### Methods

# Introduction

In the context of driving, man-machine systems have recently been envisioned to adapt to the driver's cognitive state to mitigate accident risk as a result of cognitive failure (Hancock et al., 2013). A crucial step in this direction is to understand how different tasks affect the mental capacity of the drivers. Previous research by Scheunemann et al. (2019) demonstrated that visuospatial demands and working memory not only affect driving performance but show interactions, which complicate accurate predictions of the driver's mental state. Scheunemann et al. (2019) have proposed that the interaction between the two cognitive concepts could be due to a common resource at a taskunspecific level or a task-specific level.

Understanding how these tasks affect cognitive load and where they show interactions while driving requires a comprehensive understanding of the underlying computational mechanisms of the task (Kriegeskorte & Douglas, 2018), which is why we developed two ACT-R models based on the driving model by Salvucci (2006): one implementing a bottleneck merely at the central processing unit, the other additionally implementing a bottleneck at the problem state. We use these models to explain where common resources can cause interactions between different kinds of cognitive load in a simple driving-simulator experiment.

The models used in this study was a modification of the ACT-R driving model by Salvucci (2006), re-implemented in  $Java^{1}$ .

The models performed a highway driving task, while navigating through concurring traffic following the experimental design by Unni et al. (2017). The road layout changed between a three-lane highway with 3.5m lanewidths and a two-lane construction site with 2.5m lanewidths.

At the same time, the models performed a modified n-back task involving speed signs, which occurred every 20s on the right side of the road. Depending on the n-back level (ranging from 0 to 4) the model had to drive according to the speed that was presented n signs back. Thus, the 0-back condition translates to common highway driving.

To interleave both tasks, the models used threaded cognition (Salvucci & Taatgen, 2008) that dictates which task is pursued based on available resources.

# **Central Bottleneck Model**

Based on the work by Salvucci & Beltowska (2008), we did not explicitly model an interaction between the tasks in the central bottleneck model but hypothesized that the model would predict human driving behavior by a contention for the central processing unit of ACT-R. As only one production rule can be initiated at the same time, performing the n-back task simultaneously can cause a delay in the execution of production rules of the driving loop causing fewer steering updates (purple dashed box with diagonal lines in Figure 1).



Figure 1: Demonstration of the two bottlenecks. The figure starts when a new sign appears and the correct speed is being recalled. The driving loop is ongoing, and a new iteration is initiated by attending the near point ("attend-near"). Boxes with diagonal lines signify the delay period due to the specific bottlenecks.

<sup>&</sup>lt;sup>1</sup> https://www.cs.drexel.edu/~salvucci/cog/act-r/

For the driving of the model, part we updated the Salvucci (2006) model to implement a lowcontrol loop and a high-control loop. The high-control loop is identical to Salvucci (2006), which continuously negotiates a new steering angle using a near and far point at the center of the road. However, the low-control loop does not update the steering angle but merely checks if the car is in a safe position on the road. The safety margin is based on the distance to the lane edges and was parameterized for a good model fit. If the car is in an unsafe position, it transitions back to the high-control loop to steer back to a safe position on the lane. The safety margin is identical in both driving conditions. Because the construction site is narrower than the normal highway, the car spends less time in a safe position on the road and enters the high-control loop and consequently updates the steering angle more often.

The n-back task is modeled via a sequential recall. When a speed sign is encountered it is stored in declarative memory together with a unique episodic marker indicating when the speed sign was observed. In addition, the chunk contains a reference to the speed sign encountered directly before. Thus, the memorized list of speed signs can be described as a linked list going backwards in time. As each rehearsal may potentially interfere with driving due to a competition for resources, the number of times the model rehearses has a direct effect on the driving performance and, thus, has been adjusted to fit the model. To follow the correct speed, the target speed is held in a chunk in the problem state. During recall this chunk is updated according to the n-back task.

#### **Problem State Bottleneck Model**

In the problem state bottleneck model, we revised the parameters regarding the control law and implemented a restriction to the start of each iteration of the driving control loop, which starts with the "attend-near" production such that it could only be initiated if the problem state is not busy (green arrow indicates the delay in Figure 1), which it is for 200ms after creating a new chunk in the buffer. This restriction can delay the execution of the driving loop (green dashed box with diagonal lines in Figure 1) and acts as a second bottleneck in the model.

For the revision of the n-back model we categorized the target speed as control information and stored it in the goal buffer chunk of the driving goal. In the recall or rehearsal process, the model goes through the speed signs backwards in time. While doing so, each of the signs is held in the problem state and released when the previous sign is recalled. Thus, the chunks encoding the speed signs are constantly replaced in the recall and rehearsal process. Upon reaching the target sign, the problem state is cleared before a new rehearsal starts. In this time window, the problem state is not occupied.

#### Human data

The experimental data, which was used to validate the model was recorded using the same simulation the model was



driving in. Twenty-five participants completed a block for each pairing of n-back condition and visuospatial condition twice for a total of 20 blocks.

#### Results

As can be seen in Figure 2, the central bottleneck model underestimates the steering reversal rate in total compared to human behavior, which results in a lower number of steering reversals overall. Additionally, the decrease of steering reversal rate (SRR) over n-back level is only marginal in the model and significantly higher in human participants.

In the problem state bottleneck model, we observe a better fit to human data. This is evident for the SRRs across all conditions, but also for the effect of decreasing SRRs as nback difficulty increases.

In addition, the central bottleneck model captures the effect of narrower lane width in the construction condition, resulting in a higher number of steering reversals, which can be seen in human participants. Importantly, the revised model is able to show the same effect of decreasing SRRs while still showing differences in SRRs between n-back levels.

#### Discussion

The ACT-R models are able to show how both tasks compete for available resources on either a task-unspecific level or task-specific level. In the central bottleneck model, the driving behavior is mainly influenced by a contention for the central processing unit simulating a bottleneck at a taskunspecific resource. This model demonstrates that a central bottleneck is insufficient to account for human behavior regarding the influence of the secondary task. The implementation of a bottleneck for the problem state shows that both the driving task and n-back task require this resource indicating a bottleneck at a task-specific resource.

#### References

Hancock, P. A., Jagacinski, R. J., Parasuraman, R., Wickens, C. D., Wilson, G. F., & Kaber, D. B. (2013). Humanautomation interaction research: Past, present, and future. *Ergonomics in Design*, 21(2), 9–14.

- Kriegeskorte, N., & Douglas, P. K. (2018). Cognitive computational neuroscience. Nature Neuroscience, 21(9), 1148–1160. https://doi.org/10.1038/s41593-018-0210-5
- Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Hum. Factors*, 48(2), 362–380.
- Salvucci, D. D., & Beltowska, J. (2008). Effects of memory rehearsal on driver performance: Experiment and theoretical account. *Human Factors*, *50*(5), 834–844.
- Salvucci, D. D., & Taatgen, N. A. (2008). Threaded Cognition: An Integrated Theory of Concurrent Multitasking. *Psychological Review*, 115(1), 101–130.
- Scheunemann, J., Unni, A., Ihme, K., Jipp, M., & Rieger, J. W. (2019). Demonstrating Brain-Level Interactions Between Visuospatial Attentional Demands and Working Memory Load While Driving Using Functional Near-Infrared Spectroscopy. *Front. in Hum. Neurosci.*, 12, 542.
- Unni, A., Ihme, K., Jipp, M., & Rieger, J. W. (2017). Assessing the driver's current level of working memory load with high density functional near-infrared spectroscopy: A realistic driving simulator study. Frontiers in Human Neuroscience, 11, 167. https://doi.org/10.3389/fnhum.2017.00167

# **Obstacles to the Skill-Based Approach: Why is Skill Reuse so Difficult** for Cognitive Architectures?

### Corné Hoekstra (c.hoekstra@rug.nl)

Bernoulli Institute for Mathematics, Computer Science, and Artificial Intelligence University of Groningen, Groningen, the Netherlands

## Sander Martens (s.martens@umcg.nl)

Department of Biomedical Sciences of Cells and Systems University Medical Center Groningen, University of Groningen, Groningen, the Netherlands

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

Bernoulli Institute for Mathematics, Computer Science, and Artificial Intelligence University of Groningen, Groningen, the Netherlands

#### Abstract

Skill reuse is a commonly accepted aspect of human cognition but it has been difficult to translate to cognitive architectures. We developed the skill-based approach which enables modelers to create models composed of skills created for other tasks but it does not (yet) support fully reusable skills. We will discuss three factors that prevent full reusability: inflexible WM, rigid goal selection and all-or-nothing condition checking. The factors are discussed in the context of the architecture PRIMs but they also apply to many other cognitive architectures. Finally, we discuss possible solutions to alleviate these issues.

Keywords: cognitive modeling, PRIMs, ACT-R, skill reuse, generalizability, cognitive architecture

Many tasks share considerable overlap in the cognitive elements required to complete it (Lee & Anderson, 2001). This cognitive overlap is one of the key fundamental principles underneath the attempt of cognitive architectures to arrive at a unified theory of cognition. In cognitive architectures the overlap in cognitive elements is put into practice by defining the blank slate cognitive system (i.e., the architecture) consisting of modules and buffers that underlies all behavior (Anderson et al., 2004). This approach has led to successful modeling of a wide range of tasks and paradigms; however, a crucial additional consequence of the cognitive overlap between tasks has never received much attention. Not only can the same architecture be used to complete many tasks, this architecture can also very often be used in the same way (i.e., with the same procedural knowledge). Incorporating this into cognitive architectures would take into account the fact that huge proportions of our capabilities have been acquired through a long process of development and learning while currently only the innate aspects of cognition are considered. In order to bring this idea into practice, we have developed the skill-based approach to cognitive modeling.

This approach can be valuable for multiple reasons. Firstly, models will mirror human behavior more closely which will improve model fit (Stearns & Laird, 128

2018). Secondly, reusing procedural knowledge is a large contributor to the flexibility people possess in executing various tasks. Incorporating it into cognitive modeling and AI could strongly improve flexibility and robustness (Taatgen, Huss, Dickison, & Anderson, 2008). Finally, the large range of models created in the different fields of cognitive science can be integrated more easily if they all draw from one pool of basic building blocks.

#### PRIMs

We have explored the idea of skill reuse in the cognitive architecture PRIMs (Taatgen, 2013). We will give a short introduction to PRIMs here and in the relevant sections further down the paper. (See Taatgen (2013) for a complete introduction). PRIMs is based on ACT-R and inherits many of its properties. It is a cognitive architecture built up from distinct cognitive modules whose actions are controlled by "productionrules" (operators in PRIMs) and it contains a similarly functioning declarative memory system. An important distinction between the two architectures is that the operators in PRIMs are built up from smaller units than ACT-R's production rules. These smaller units are the primitive information processing elements (PRIMs). PRIMs are considered the basic elements of cognition and are only capable of either moving or comparing pieces of information in the workspace. Although a single PRIM is not very powerful, combinations of PRIMs (i.e., operators) are able to execute complex cognition on the same level as ACT-R. These primitive operations are assumed to be universally applicable to any task and therefore can provide low-level mechanisms of transfer. They are also relatively easy to implement in neural architectures (Stocco, Lebiere, & Anderson, 2010). The central concept of the skillbased approach, a skill, is one level above an operator. A skill is a reusable collection of operators that perform a part of a task. Although a skill is larger than an operator, carrying out a skill still only takes a small amount of time in the order of one second or less.

The low-level transfer combined with the higherlevel concept of a skill make PRIMs well-suited for exploring the skill-based approach although (most of) its principles can be implemented in other cognitive architectures as well.

#### The Skill-Based Approach

The central idea of the skill-based approach is to construct models of tasks in the same way humans would approach a new task. When people are confronted with a new task, they do not need to figure out from scratch how to complete this task but instead can rely on previously learned knowledge which has proven successful (Salvucci, 2013). A good example of this are the experimental tasks typical of cognitive psychology. Participants have usually never encountered these tasks before, yet they are quickly able to figure out what to do. Since they do not have time to learn new procedural knowledge specific to this task, it suggests that they reuse existing procedural knowledge. Concretely, the skill-based approach assumes that learning (almost) any new task merely means composing it from already existing skills.

A fundamental challenge to emulating this humanlike flexible behavior in cognitive models is balancing generalizability with accuracy. Different tasks come with different contexts and the model needs to be general enough to function in all these contexts but also specific enough to produce the same result regardless of that context. The common solution to this challenge is to allow for dynamic variable binding (Greff, van Steenkiste, & Schmidhuber, 2020); that is, allow variables to take on different values depending on the context. Although this solution is commonly adopted across different types of AI, there is no consensus on how it should be implemented (Feldman, 2013). The solution adopted by PRIMs is variable instantiation; a skill is created with general variable names which are only defined (instantiated) when the skill is used in a new context. However, there is no principled way in which this mechanism is implemented in the architecture.

More exact details can be found in our previous publications on the skill-based approach in which we propose the method (Hoekstra, Martens, & Taatgen, 2020) and test the validity of its predictions (Hoekstra, Martens, & Taatgen, 2022), but in short the skill-based approach works as follows. The first step of the skillbased approach is determining which basic skills are responsible for performing the modeled task based on previous literature. This step comes forth out of the fundamental principle of the skill-based approach that every task is a composition of basic processing steps that have been done (many times) before. For example, in the attentional blink (Martens & Wyble, 2010) model we have constructed (Hoekstra et al., 2020), the four basic skills we included were 'visual search', 'consolidation', 'retrieval', and 'response'. Skills that were reused from other models. This first step increases the generalizability of a model because the ubiquity of its basic building blocks allows it to be easily linked to other models and theories. The second 129 system but instead consists of two modules that

step involves creating and testing the validity of the basic skills. In this step, other models which include (some of) the basic skills are built and these models are compared with human data. In our attentional blink model, we completed this step by creating a model of a simple visual discrimination task and two working memory tasks (a simple working memory task and a complex working memory task). This step is necessary to create the basic skills and it provides evidence for the accuracy of these skills. The final step involves adapting the basic skills to the context of the task of interest. In PRIMs, the cognitive architecture we used, this is done by instantiating the skills.

Following this method, we succeeded in constructing a model of the attentional blink (AB) that consisted of elements (skills) that worked in both the original task (e.g., the complex working memory task) as well as the AB task. This shows that it is possible to create cognitive models out of elements created for other tasks and that models can be created by merely assembling already existing procedural knowledge. However, the process of creating these skills was quite laborious and it often required making modifications to the basic skills that seemed too "AB-specific" to be part of general basic skills (Hoekstra et al., 2020). In short, we succeeded in creating a model with reused skills but not with fully reusable skills. That is, we managed to create an AB model out of skills that are also parts of other models (and are therefore *reused*) but these skills cannot be freely reused in every other task that includes the same basic skill (i.e., they are not fully *reusable*). However, this is crucial; making the step from *reused* skills to *reusable* skills would realize the full potential of the skill-based approach. It would standardize the knowledge used in cognitive models as well as increasing the ease with which skill-reuse can be implemented during model building.

# Current paper

In the current paper, we will discuss which factors cause the difficulties in creating fully reusable skills. We will describe three open questions that complicate the implementation of the skill-based approach, specifically in PRIMs but some also apply to ACT-R. Although these open questions demonstrate practical problems in implementing the skill-based approach, they also point to fundamental unanswered questions about how flexibility should be balanced with cognitive plausibility as well as learnability. The questions will be illustrated by challenges we encountered while using the skill-based approach to model the updating tasks described by Miyake and colleagues (Miyake et al., 2000).

# **Inflexible Working Memory**

In PRIMs and ACT-R the main purpose of working memory (WM) is to keep relevant information guickly available and to support the building of new chunks. WM in ACT-R does not consist of one dedicated together function as WM: declarative memory and the problem state (Nijboer, Borst, van Rijn, & Taatgen, 2016). Declarative memory is responsible for storing chunks while the problem state takes care of keeping the chunks immediately available and is capable of creating new chunks.

In PRIMs, WM does consist of a single dedicated module responsible for keeping information readily available and for creating new (long-term) memory chunks. This module is called the imaginal buffer; however, it is often referred to as the WM-buffer and, for clarity, we will follow that convention. The WMbuffer in PRIMs works as any other buffer in the architecture in the sense that it has slots in which information can be placed and retrieved without any penalty. The slots function independently of one another and are numbered starting with one. Information is placed in and withdrawn from WM by a PRIM. For example, placing information presented on the screen in WM can be done by the PRIM  $V1 \rightarrow$ *WM1* and information can be taken out from WM by a PRIM such as  $WM1 \rightarrow AC2$ . Information can also be moved around within WM, for example  $WM4 \rightarrow$ WM1. The use of numbered slots in WM makes it much easier to reuse skills and operators compared to using named slots such as in ACT-R. However, it is not flexible enough to facilitate full reusability because the numbered slots are often still too rigid.

The inflexible working memory causes two main issues. The first is that the slots that will be used by the skills in the separate tasks need to be calibrated to work together. This requires a lot of effort from the modeler and although it is manageable for smaller and homogenous models, it quickly becomes unwieldy when the model involves many skills and different types of tasks. This is not a fundamental limitation, but it does present an obstacle to the adoption of the skillbased approach, especially when skill reuse is only a secondary interest. The second issue is more fundamental. Reusability of skills depends on the availability of the WM slots used in the original task. When these slots are not available in a different task, the skill cannot be reused. For example, the 'read' skill in our updating model stores the newly presented item in WM5 because the first four slots are used to keep track of the previously presented items. This might become problematic if the model would move on to a five-item memory task because the WM5 slot will be used to keep track of the fifth item. This illustrates that WM is not flexible enough unless a skill is designed while keeping every possible combination of tasks in mind and that full *reusability* is not yet possible.

Besides causing practical difficulties in using the skill-based approach, the issues with WM also point to a more fundamental question of how WM should be implemented in a cognitive architecture. The challenge is that WM needs to be extremely flexible on the one hand, but also consistent with the limitations that have been identified in the literature on working memory.

The buffer-based design of PRIMs' WM has the advantage of being relatively flexible. It can be used in many types of tasks and it can store many types of information, additionally it provides a means of keeping information readily available. However, it lacks some plausibility because it assumes perfect (decay-free) storage of its contents which is not fully in line with the WM literature.

The alternative to using a buffer for WM is to store items in declarative memory. This is an attractive option, because it puts no hard limit on the number of items, but it still imposes a soft limit through memory decay. However, using declarative memory as WM also has a strong limitation in the sense that the information is not readily available, and has to be retrieved first. Given that items can only be retrieved one at a time, it is impossible to interrelate two or more items, which is a necessity for almost all tasks.

In conclusion, the practical issues we encountered while exploring the skill-based approach not only point to implementation issues but also to fundamental questions of how flexibility and plausibility should be balanced in WM.

## **Rigid Goal Selection**

The goal module plays a central role in determining which production will fire in both ACT-R and PRIMs. Although the goal buffer plays a similar role in both architectures it does not work in the same way. In ACT-R the goal buffer influences production selection through the goal-state chunk present in the goal buffer and exerts its influence in a very explicit manner. Only production-rules which condition side matches the pattern in the goal-state chunk will be considered for selection. This way, the goal module is largely responsible for guiding the model towards firing the right productions at the right time.

The goal module in PRIMs has the same general role and also is responsible for the broad strokes 'supervision' of the model through a task. However, the goal module in PRIMs executes its role in a different and less explicit way. Operator selection in PRIMs is determined by the activity of the operators in memory. The most active operator gets selected first and its conditions are compared to the current context, if the conditions match the context the operator will fire. If the conditions do not match, the next most active operator will be retrieved and its conditions tested. This process repeats until an operator with matching conditions is found which will then fire. The goal buffer has a large influence on this process by spreading activation to operators that are associated with the current goal. This biases the selection process towards selecting operators that match the goal without guaranteeing that such operators will fire (noise or non-matching conditions can still prevent it). The subtle but forceful influence the PRIMs goal module exerts allows for organized behavior while still allowing for flexibility within a task and, importantly, 130 between tasks. The limitation related to the goal

module is not how the goal module impacts operator selection but instead in how the goal itself is selected.

As is the case with all exchanges of information in PRIMs, goals are also determined by a PRIM. A new goal becomes active by a PRIM updating the value in G1 (the first slot of the goal-buffer). Although it is also possible to create situations in which multiple goals are active, for simplicity sake we will focus on a situation with one active goal. Goals are defined by symbols (similar to ACT-R) and therefore setting 'respond' as the goal can be done by the PRIM respond -> G1, if there is a skill with that same name. There are no rules about when or how the goal-determining PRIM needs to fire, however the architecture is designed in such a way that the most logical place for such a PRIM is in the final operator of a skill. This is very useful for simple models because it allows for an easy to understand (and flexible) way in which the model moves from one goal to the next. However, it becomes limiting in more complex models, especially in tasks in which the order of the goals is not always the same.

Determining the next skill within the previous skill essentially means that the next goal is decided by the previous goal. This severely limits full reusability of a skill because the role of a skill differs depending on the task. In some tasks, a certain skill might only be used at the end of a task (and therefore would not even require a *next-skill* operator) while in a different task the same skill might be a central part of the task and be used multiple times within a single trial. Switching skills gets further complicated by condition checking (which will be discussed in the next section) because different conditions might require the same skill to be performed next and, therefore, require separate operators. Often these limitations lead to a large array of different operators whose only function is switching to the next skill in different situations. For example, the 'update-WM' skill required four different operators only for switching between skills in the three tasks we modeled due to its centrality in those tasks. Extending the 'update-WM' skill to more tasks would only introduce more of such operators even though the basic procedural knowledge of updating WM would remain the same. This puts the cognitive plausibility of this way of switching skills into question, because it implies that every skill includes many operators that are only responsible for switching to the next skill.

This exposes two core limitations that are present in the current conception of PRIMs (and also ACT-R). Firstly, skills take care of two separate aspects of cognition: they perform the cognitive processing steps and are responsible for goal selection. That is, they are responsible for both selecting the goals and ensuring that they are achieved. This makes skill reuse difficult because, as our example shows, the basic procedural knowledge (which takes care of achieving a certain goal) might remain stable in most situations but the goal selection process might be different. Separating goal selection from goal execution will make skill

reuse much easier. The second limitation is related to the type of information on which goal selection is based. Currently, goals are purely selected based on declarative knowledge. At the start of a task, by creating the goal-switching operators a 'plan' for the task is laid out and the model is practically incapable of deviating from this path. This way of goal selection is too rigid and overlooks the fact that people select goals based on a plan combined with their perception of the current situation (Altmann & Trafton, 2002).

Our modeling suggests that goal selection should be separated from execution and be made more flexible. However, this is not an easy task. The basic assumptions of PRIMs do not consider goal selection a special case of cognition and posit that it should be accomplished by a PRIM. Furthermore, increasing the flexibility of goal selection leads to questions of how this flexibility can be balanced with reliability since a more flexible model will also be more unpredictable.

## Condition checking

The final factor limiting the creation of fully reusable skills we will discuss here is related to a fundamental aspect of both ACT-R and PRIMs, namely condition checking. In both architectures, productions consist of a condition side (left-hand side) and an action side (right-hand side). The conditions are compared to the content of the buffers before the action side is executed. In ACT-R, the conditions of all productions are evaluated in parallel and when multiple productions match the current contents of the buffer the production with the highest utility factor will be chosen. In PRIMs, condition checking occurs serially starting with the first condition of the most active operator. When one of the conditions does not match, the next active operator will be tested until a matching operator is found. This takes a certain amount of time at first, but after a while most conditions will be compiled into one execution cycle and the most active matching operator will usually be picked without any time cost (comparable to ACT-R).

Conditions are thought to be a fundamental part of knowledge in both architectures. procedural Therefore, full skill reusability means that both the action as well as the condition side need to be reused. Although the action side usually works in both tasks, the condition side is more problematic. After all, a different task usually means a different context to which the conditions will be matched. This often means that the condition side of an operator needs to be adapted to the new task which hinders reusability. Conditions that are especially challenging are those that are related to specific situations in a certain task. For example, in one of the updating models WM needed to be updated based on information in the visual buffer while in a different model it had to be updated based on information in WM itself. In this situation the action PRIMs (the right-hand side) were identical, but a different operator still needed to be 131 created to accommodate the difference in conditions.

This leads to the question to which extent conditions are reused. The quick learning displayed by humans suggests that some previously learned condition-action associations are retained when a new task is performed, however our modeling implies that this does not apply to all of them. Take for example the operator depicted below.

```
operator respond-value-WM1 {
V1 = *report-instructions
WM1 <> nil
==>
*action -> AC1
WM1 -> AC2
}
```

This operator gives the response (stored in WM1) at the end of a trial by performing an action (e.g., pressing a key on a keyboard). In this case, the second condition can be retained without problem because reporting WM1 would always require WM1 to not be empty. However, the other condition which tests whether the report instructions are currently on-screen should probably not be retained because it depends on the task.

The example suggests that not all conditions are created equal and that some conditions should not be reused. Especially conditions aimed at representing a task-specific situation hinder skill reuse suggesting that conditions might not be the best way to represent task-specific context.

# **Potential solutions**

The three limitations we discussed impede the practical usefulness of the skill-based approach but we believe that they will not present a fundamental roadblock to fully *reusable* skills. The limitations we discussed are largely consequences of the reliance of cognitive models on the input of task-specific details from the modeler. Therefore, these issues might be alleviated by implementing learning mechanisms with which the model can figure out task-specific details independently or by providing more principled ways in which the modeler can specify such details.

The first limitation we discussed involved WM. The key issue here is that the inflexible WM demands a lot of coordination from the modeler because the model is not aware of the identity of the WM contents. A possible way to alleviate this would be to store the tobe remembered value together with its meaning (e.g., store the value "four" together with "currentstimulus"). This cannot be done in the current conception of the WM; however, the DM module does possess the required properties. By storing chunks in the DM (such as depicted below) the model would be aware of the value as well as the identity.

ISA fact SLOT1 binding-fact SLOT2 current-stimulus SLOT3 four In this situation, the current PRIMs imaginal buffer (i.e., the WM buffer) would be used almost exclusively to facilitate the creation of new chunks and as a problem-state (Borst, Taatgen, & van Rijn, 2010). Importantly, in order to keep the high flexibility of a buffer-based WM, these chunks should be accessible without the need of an explicit retrieval request but instead through means of a PRIM. For example, by allowing a PRIM to directly create bindings (e.g., *four -> \*current-stimulus*).

This way of organizing WM provides a better balance of flexibility and plausibility, because chunks are subject to decay and retrieval times, however the information in WM is still easily accessible because it can be directly done by a PRIM. Furthermore, this design of the short-term memory would also provide a mechanism for the variable binding problem discussed earlier in the introduction. The dynamic bindings required to facilitate flexible model behavior could be stored in this same manner. Ideally, the model would create these flexible binding chunks independently (e.g., when 'reading' the instructions) which would tremendously improve skill reusability as well as model autonomy.

The second limitation we discussed involved the manner in which the next skill is selected in PRIMs. This issue boils down to how the next goal is placed in the goal buffer. In the current situation, the previous skill usually places the next skill in the goal buffer but this method creates a large amount of procedural knowledge only aimed at switching between skills.

There is a possible solution that fits the PRIMs philosophy. Instead of having one active skill, two skills can be active: one skill for execution, and one skill for planning. The execution skill carries out the actions required to achieve a particular subtask, and then terminates itself. The planning skill is then responsible for selecting a next skill.

This would be a big improvement over the current situation because it allows for goal switching separate from goal execution based on both a pre-made plan as well as the current context. Additionally, it allows for a flexible representation of task-specific information without the need to include such information in the general skills.

The final limitation we discussed concerned condition-checking. The limitation to skill reusability associated with condition-checking is that every task has a different context which makes it likely that the original conditions will not apply. Additionally, our modeling showed an important distinction between generally applicable conditions and task-specific conditions and raised the question whether conditions are the best way to represent task-specific context.

Testing conditions is one way to establish a mapping between the current state of the cognitive system and the action to be taken, but not the only one. Neural network approaches to modeling operators often use inspiration from the basal ganglia. The basal ganglia are considered to be central to forming 132

context-action mappings and recent modeling efforts have created models capable of creating such mappings. These mappings provided reusable contextaction associations while retaining flexibility by means of small changes to the connection weights in the network (Stewart, Bekolay, & Eliasmith, 2012; Taatgen, 2020).

Such functionality could be incorporated in production-based architectures by specifying (or learning) connections between certain items in the workspace and operators. For example, the first condition of the previously mentioned example could be replaced by specifying a positive connection (through spreading activation) between the reportinstructions and this operator. This would make it more likely that it gets picked when such instructions are on the screen but it does not prevent the operator from firing when they are absent. This functionality is already possible in PRIMs but it might be helpful to explicitly make it part of an operator definition (in addition to conditions) which is not only practical but also highlights that these connections are reused.

## Conclusion

The skill-based approach is a promising addition to the arsenal of a cognitive modeler; however, the previous discussion has shown that there are still some important limitations. The inflexible WM demands a lot of coordination from this modeler, the unnatural goal selection requires a large amount of inefficient procedural knowledge and the all-or-nothing condition checking severely hampers the versatility of operators. Resolving these issues will require some substantial modifications to the cognitive architecture we employed and to production-system architectures in general. We proposed some solutions in this paper which we will explore in a subsequent study.

The current paper resulted from attempting to apply the skill-based approach to a series of basic tasks that make use of skills that are widely used. The difficulties we experienced show that current cognitive architectures do not support the creation of fully *reusable* skills. This does not mean, however, that the skill-based approach is completely ineffective, current architectures do support the use of *reused* skills and capitalizing on this characteristic will already result in more valid and generalizable models.

#### References

- Altmann, E. M., & Trafton, J. G. (2002). Memory for goals: an activation-based model. *Cognitive Science*, 26(1), 39–83.
- 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. https://doi.org/10.1037/0033-295X.111.4.1036
- Borst, J. P., Taatgen, N. A., & van Rijn, H. (2010). The problem state: a cognitive bottleneck in multitasking. *Journal of Experimental Psychology. Learning*, 133

Memory, and Cognition, 36(2), 363–382.

- Feldman, J. (2013). The neural binding problem(s). *Cognitive Neurodynamics*, 7(1), 1–11.
- Greff, K., van Steenkiste, S., & Schmidhuber, J. (2020). On the binding problem in artificial neural networks. *ArXiv Preprint ArXiv:2012.05208*.
- 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. https://doi.org/10.1111/tops.12514
- Hoekstra, C., Martens, S., & Taatgen, N. A. (2022). Testing the skill-based approach: Consolidation strategy impacts attentional blink performance. *Plos One*, *17*(1), e0262350.
- Lee, F. J., & Anderson, J. R. (2001). Does Learning a Complex Task Have to Be Complex?: A Study in Learning Decomposition. *Cognitive Psychology*, *42*(3), 267–316.
- Martens, S., & Wyble, B. (2010). The attentional blink: past, present, and future of a blind spot in perceptual awareness. *Neuroscience and Biobehavioral Reviews*, 34(6), 947–957. https://doi.org/10.1016/j.neubiorev.2009.12.005
- Miyake, A., Friedman, N. P., Emerson, M. J., Witzki, A. H., Howerter, A., & Wager, T. D. (2000). The unity and diversity of executive functions and their contributions to complex "frontal lobe" tasks: A latent variable analysis. *Cognitive Psychology*, *41*(1), 49–100.
- Nijboer, M., Borst, J., van Rijn, H., & Taatgen, N. (2016). Contrasting single and multi-component working-memory systems in dual tasking. *Cognitive Psychology*, *86*, 1–26.
- Salvucci, D. D. (2013). Integration and reuse in cognitive skill acquisition. *Cognitive Science*, *37*(5), 829–860.
- Stearns, B., & Laird, J. (2018). Modeling Instruction Fetch in Procedural Learning. In *Proceedings of the* 16th international conference on cognitive modeling.
- Stewart, T. C., Bekolay, T., & Eliasmith, C. (2012). Learning to select actions with spiking neurons in the Basal Ganglia. *Frontiers in Neuroscience*, *6*, 2.
- 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– 574.
- Taatgen, N. A. (2013). The nature and transfer of cognitive skills. *Psychological Review*, 120(3), 439– 471. https://doi.org/10.1037/a0033138
- Taatgen, N. A. (2020). A spiking neural architecture that learns tasks. In *Proceedings of ICCM 2019 -17th International Conference on Cognitive Modeling* (pp. 253–258).
- Taatgen, N. A., Huss, D., Dickison, D., & Anderson, J. R. (2008). The acquisition of robust and flexible cognitive skills. *Journal of Experimental Psychology. General*, 137(3), 548–565.

# Individual Differences and Levels of Analysis in Computational Models of Coordination

Alexander R. Hough (alexander.hough.1@us.af.mil)

Air Force Research Laboratory Wright-Patterson AFB

#### Ion Juvina (ion.juvina@wright.edu)

Department of Psychology, College of Science and Mathematics, Wright State University

Dayton, OH

#### Abstract

Coordination failure is common and literature suggests individual differences play a role. Individuals are hypothesized to use strategies, form beliefs about others, and have different starting preferences. Extant data analysis and modeling efforts focus on average-level behaviors and often ignore individual differences in coordination strategies and their correspondence to cognitive mechanisms. Here, we leverage computational models to better understand individual differences and underlying cognitive mechanisms. We use experimental data from a coordination game to assess and compare a model from behavioral game theory, an extension of that model, and a recently developed cognitive model. This work presents challenges for modeling coordination dynamics, strategies, and how players form beliefs about others.

**Keywords:** Coordination; Group dynamics; Signaling; Coordination strategies; ACT-R; Cognitive model

#### Introduction

Coordination failure is common (Camerer, 2003; Riechmann & Weimann, 2008; Van Huvck, Battalio, & Beil, 1990, 1991), and could involve either miscoordination (i.e., failure to converge on a choice) or inefficiency (i.e. converging on a suboptimal choice). It is often attributed to the lack of salient focal points leading to efficient outcomes (Mehta, Starmer, & Sugden, 1994) and individual differences in starting strategies (Costa-Gomes, Crawford, & Iriberri, 2009; Van Huyck et al., 1990, 1991), persistence in trying to improve outcomes (Brandts, Cooper, & Weber, 2015), sensitivity to risks (Cachon & Camerer, 1996), and reciprocity (Offerman, 2002). To counteract these issues, individuals can form and update beliefs about what others will do (Camerer, 2003), engage in counterfactual thinking to consider what could have happened (Hough, O'Neill, & Juvina, 2021), and nudge others to make better choices by signaling or leading by example (Brandts et al., 2015; Hough et al., 2021). However, coordination dynamics and individual differences are not well understood. Extant research focuses on average behavior, rather than individual differences for analysis (Bortolotti, Devetag, & Ortmann, 2016; Leng, Friesen, Kalayci, & Man, 2018; Van Huyck et al., 1991, 1990) and modeling (Camerer & Ho, 1999; Costa-Gomes et al., 2009), which can lead to faulty conclusions about strategy use, strategy shifts due to learning, and differences between individuals (Siegler, 1987). To better understand coordination failure and individual differences, we turn to computational models that require detailed

134

specification of mechanisms which serve as testable hypotheses and allow observation of "black box" processes. We use experimental data collected from the minimum effort game (i.e., MEG) (Van Huyck et al., 1990) to assess and compare a model from behavioral game theory, an extension of that model, and a recently developed cognitive model.

#### The MEG

The MEG is a weak link coordination game where each player chooses a level of effort between one and seven, and their payoff is determined by their choice and the group minimum (Table 1). There are seven coordination points

Table 1: MEG payoff matrix.

		Minimum Effort Choice in Group						
		1	2	3	4	5	6	7
ice	1	70						
iou	2	60	80					
t T	3	50	70	90				
ffoi	4	40	60	80	100			
ц	5	30	50	70	90	110		
iye	6	20	40	60	80	100	120	
Plá	7	10	30	50	70	90	110	130

(i.e., Nash equilibria) represented diagonally in Table 1. Van Huyck et al. (1990, 1991) suggested players use risk and payoff dominant strategies. The risk dominant strategy is low risk, low reward, and is individually focused. Choosing one results in the same payoff regardless of other player's choices. The payoff dominant strategy is high risk, high reward, and is more group focused. The highest choice of seven can result in either the highest or the lowest payoff, depending on other's choices. These strategies serve as focal points, and over time, the minimum can become more influential. This was proposed as a simple explanation for the frequently observed negative trend in effort (i.e., efficiency), with the minimum and full (i.e., all player choices) feedback (Camerer, 2003; Leng et al., 2018; Van Huyck et al., 1991, 1990). However, there is evidence that players signal (Hough et al., 2021; Leng et al., 2018), engage in counterfactual thinking (Camerer & Ho, 1999; Hough et al., 2021), and speculation that individuals form beliefs about others "player type" corresponding to their initial preferences and future choices (Camerer, 2003).

MEG experiments (Bortolotti et al., 2016; Leng et al., 2018; Van Huyck et al., 1990, 1991) typically focus on average efficiency (i.e., minimum and average effort), and rarely explore group and individual behavior. However, Leng et al. (2018) identified signaling as alternating between the minimum and higher effort, Bortolotti et al. (2016) identified weak links as an early source of coordination failure, and Hough (2021); Hough et al. (2021) measured coordination by calculating intra-group variance and signaling by calculating each player's distance from the minimum. These contributions have not significantly improved our understanding of coordination dynamics or individual differences. In an attempt to address these issues, we turn to computational models. We discuss the experience-weighted attraction model (i.e., EWA) (Camerer & Ho, 1999) and an extended version to highlight limitations that motivated developing the cognitive model. We then compare model fits to data from a MEG experiment with four-human groups and full feedback (Hough, 2021)

# **Computational Models of the MEG**

# The EWA Model

EWA is based on forming and updating attractions towards all possible choices. It features initial choice attractions from prior experience, updating attractions weighted by recency and experience, and forgone payoffs that could have been earned (i.e., counterfactual thinking). The model has four parameters: 1) forgone payoff weight ( $\delta$ ) for counterfactual thinking, 2) past attraction decay ( $\phi$ ) and 3) experience decay  $(\rho)$  that control the growth rate of choice attractions, forgetting and recency effects, and 4) discrimination sensitivity  $(\lambda)$  to account for individual differences. The functioning of EWA can be described in three equations that calculate: experience, choice attractions, and choice probability. The experience equation calculates current experience (i.e., rounds), N(t), based on previous experience, N(t-1), that is depreciated,  $\rho < 1$ , each time there is a new experience (+1): N(t) = $\rho * N(t-1) + 1$ . The choice attraction equation determines the attraction,  $A^{j}$ , for each choice,  $s^{j}$ , based on adding the depreciated previous attraction,  $\phi * N(t-1) * A^j(t-1)$ , to the current weighted payoff,  $[\delta + (1 - \delta) * I(s^j, s(t)))] * payof f$ , and dividing it by current experience, N(t). The current weighted payoff is equal to the payoff for the actual choice and is weighted by  $\delta$  for forgone choices. Choice probability is determined by the Luce choice rule (Luce, 1956). Choice probability,  $P^{j}(t+1)$ , is based on a logistic transformation by raising the Euler's number, e, to the power of the choice attraction,  $A^{j}(t)$ , multiplied by the sensitivity parameter,  $\lambda$ , and is normalized by dividing it by the sum of all logistically transformed choice probabilities,  $e^{\lambda * A^j(t)} / \sum_{k=1}^m e^{\lambda * A^k(t)}$ .

Figure 1 shows EWA processes relating to the MEG. At the first round, EWA "knows" the payoff matrix, uses pregame experience and choice attractions to determine probabilities that serve as weights for sampling a choice. The model receives the minimum as feedback and calculates actual and weighted forgone payoffs. These payoffs are used to update experience and choice attractions, which are depreciated by experience,  $\rho$ , and attraction decay,  $\phi$ . As the model only receives the minimum, it is not aware of all players choices or signaling efforts. To enable this capability, we extended EWA so it considers each player's choice as a hypothetical minimum. The extended EWA (i.e., EEWA) generates a set of attractions based on the minimum, then additional sets are generated for each hypothetical minimum. All additional sets are forgone choices so their payoffs are weighted by  $\delta$ .



Figure 1: High-level summary of EWA behavior in the MEG.

EWA and EEWA were previously fit to MEG data (Hough, 2021) and starting choice attractions were based on sampling a choice from the first round choice distribution of the human data. We chose to estimate the mean and standard deviation for each parameter to fit average effort and intragroup variance. This introduced variability (i.e., individual differences) as parameter values for each agent were sampled from a normal distribution using estimated values. EWA and EEWA used the same estimated mean and standard deviation (parentheses):  $\delta = .2(.01)$ ,  $\rho = .9(.01)$ ,  $\phi = .21(.17)$ , and  $\lambda = .49(.1)$ . After model fitting, we found several issues. Both had a large portion of agents that never varied choices (35% compared to 1.4% of humans) suggesting an artificial fit. This is likely related to the low value of  $\delta$  (.2), which conflicts with Byrne (2016), who suggest counterfactuals are weighted less than actual outcomes, but have a strong influence on behavior. Similarly, Camerer and Ho (1999) estimated  $\delta$  as .85 for a similar game. A value of .2 makes actual payoffs carry 5x more weight than forgone ones and is likely to encourage repeated choices. The models were also missing features from the literature (e.g, beliefs about other players, initial preferences, strategy use and switching). Lastly, EEWA had a better fit than EWA, but it considers 27 forgone choices which might be unrealistic. These limitations motivated the development of a cognitive model that included the missing features and a higher degree of psychological correspondence. The model, called prediction, strategy, and simulation (i.e., PSS) included: player types, player choice predictions, strategies, and counterfactual thinking.

# The PSS Model

The PSS model was implemented in the ACT-R cognitive architecture (Anderson, 2007), which includes both symbolic and sub-symbolic structures, and modules that represent systems of the mind. The PSS model uses the goal, imaginal, declarative and procedural modules. The imaginal module represents visual short-term or working memory and the goal module controls the model's current focus. The declarative memory module represents facts stored as chunks in long term memory and the availability of chunks is controlled by a sub-symbolic component. The procedural module uses condition-action rules (i.e., productions) to represent knowledge about how to do things. The procedural module's pattern matcher determines whether any production conditions match the current state and, if so, it "fires" and changes the state of the model. The PSS model includes three main features: predictions about other players, strategies, and learning. The instance-based learning framework (Gonzalez, Lerch, & Lebiere, 2003) is used for player choice predictions, however, we use a slightly different approach. Instances (i.e., chunks) are used strictly for player choice predictions, there is no pre-decision consideration of possible decisions, and decisions are a function of productions. Player choice predictions serve as input to strategies, which together produce a decision. After a decision is made, counterfactual thinking takes place and the model simulates unchosen strategies.

Players make choices simultaneously and often react to previous round choices. To capture reactions, instances store choices for two rounds: situation (t - 1) and reaction choice sets (t). To leverage reactions, the model uses last round choices as cues to retrieve an instance(s) with best matching situation choices (t-1) (i.e., target), and the reaction choices (t) are extracted. The blending mechanism (Lebiere, 1999) aggregates accumulated reaction choices to serve as player choice predictions. Instances with a higher likelihood of retrieval, determined by activation, carry more weight. The ACT-R activation equation,  $A_i = B_i + S_i + P_i + \varepsilon_i$ , 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,  $\varepsilon_i$ , for noise in memory. However, PSS uses blending instead of retrieval and only includes partial matching,  $P_i$ , and 4) noise,  $\varepsilon_i$ , terms. The blending mechanism uses an equation,  $V = min\Sigma_i P_i * (1 - sim(V, V_i))^2$ , to produce a value that minimizes the sum of all squared dissimilarities,  $(1 - sim(V, V_i))^2$ , of each chunk, *i*, and weights it by its probability of retrieval,  $P_i = (e^{M_{i/t}})/(\Sigma_i e^{M_{j/t}})$ . The probability of retrieval is a function of the match score for a chunk,  $e^{M_{i/t}}$ , which normalized by the match score of all retrieved chunks,  $\Sigma_j e^{M_{j/t}}$ . If the blended chunk has an activation below the activation threshold (default of 0), it fails and previous round choices serve as player choice predictions.

The PSS model includes four strategies (i.e., productions) that use player choice predictions to make decisions based on the 1) minimum, 2) average, 3) maximum, or 4) one higher than the average (i.e., signaling). After making a decision, the model receives feedback and updates the utility of strategies using payoffs as rewards in the ACT-R utility learning equation:  $U_i(n) = U_i(n-1) + \alpha [R_i(n) - U_i(n-1)]$ . The current utility for each strategy,  $U_i(n)$ , is a function of the: 1) previous utility,  $U_i(n-1)$ , 2) utility learning rate,  $\alpha$ , 3) temporally discounted reward value,  $R_i(n)$ , and 4) a noise component,  $\varepsilon$ . Starting utility influences which strategies are initially selected, and the learning rate influences how quickly utilities change. The PSS model includes risk (i.e., RD) and payoff dominant (i.e., PD) player types (Van Huyck et al., 1990, 1991), which are represented by a pattern of starting utilities. RD players are risk averse and PD are willing to take risk in the pursuit of higher rewards. The four strategies were sorted by risk (i.e., min, ave, max, and signal) and RD players had the highest utility for the min-strategy (i.e., 130), which linearly decreased along the continuum to the signal-strategy (i.e., 70). The PD player type was defined as the opposite of RD. All unchosen strategies simulated during counterfactual thinking receive a fraction of the forgone payoff to correspond with psychology (Byrne, 2016) and game theory literature (Camerer, 2003).

PSS model parameter values were set based on ACT-R defaults, corresponding literature, or MEG structure. There were two architectural parameters for declarative memory: Activation noise and partial matching. Activation noise was set at its default value (i.e., 1). The mismatch penalty parameter (mp) was set to a small value (i.e., 1) so that all instances have influence. Procedural memory included two architectural (i.e., learning rate and noise) and two theory-driven parameters (i.e., starting utilities and counterfactual weight). Utility noise was scaled up from 1 to 7.5 to better correspond to payoff values and utility learning rate was left at the default value of .2. A counterfactual weight parameter, set to .75, discounted forgone payoffs (i.e., 75% of payoff) for strategies during counterfactual thinking. Two player types (i.e., RD and PD) had different starting utility patterns.

The model starts the game by selecting a player type, then predicts other player choices, makes a choice, and processes the results. For the first round, predictions and choices are sampled from the first round choice distribution of the human data. For all subsequent rounds (Figure 2), the model starts each round by attempting to recall and blend past instances with situation choices (t - 1) similar to last round choices stored in the goal buffer. If successful, reaction choices (t)are blended and serve as player choice predictions. If blending fails, last round choices serve as player choice predictions. A new instance is then created in the imaginal buffer to store situation choices (i.e., last round choices), and predictions replace last round choices in the goal buffer. The model



Figure 2: Simplified diagram of the PSS model processes.

selects the strategy with the highest utility, uses player choice predictions as input, and makes a choice. The model is then shown all player choices and its own payoff (i.e., results). At this point, a reward is triggered equal to the earned payoff and the utility of the chosen strategy is updated. In addition, values (i.e., reaction choices, decision, and payoff) are added to complete the instance in the imaginal buffer. The player choice predictions in the goal buffer are used to simulate all unchosen strategies to produce forgone choices (i.e., counterfactual thinking). Actual player choices in the imaginal buffer determine the forgone payoffs weighted by the counterfactual weight parameter. After unchosen strategies are simulated, the model replaces player choice predictions with actual choices in the goal buffer, and the instance is cleared from the imaginal buffer and is added to declarative memory.

# **Model Fit and Findings**

EWA, EEWA, and PSS models simulated 100 groups with four agents playing the MEG and were fit to the average effort and intra-group variance of the human data (18 groups of 4).

The PSS model fit to average effort (Figure 3a), r(38) =.4, RMSE = .27, was better than EWA, r(38) = .52, RMSE =.96, and only slightly better than EEWA, r(38) =.55, RMSE = .31. However, EEWA had the best fit to average intra-group variance (Figure 3b), r(38) = -.06, RMSE = .42, followed by EWA, r(38) = -.10, RMSE = .53, and PSS, r(38) = -.14, *RMSE* = 1.62. EEWA also had the best fit to average payoff (Figure 3c), r(38) = .43, RMSE = 7.19, followed by PSS, r(38) = .34, RMSE = 10.4, and then EWA, r(38) = .64, RMSE = 11.26. EEWA best approximated average behavior, as PSS failed to fit intra-group variance. To better understand the data, we calculated variance and distance from the minimum (i.e., min-dist) for each individual. We found 35% of EWA and EEWA agents had choice variance of 0 and a mode of 0. About 1.4% of humans and 2% of PSS agents had no variance and modes were .27 and .05, respectively. Agent first round choices were based on first round choices of humans, with a mean of 4.38 and variance of 2.77. This lack of choice change likely contributed to EWA and EEWA's fit to both average effort and intra-group variance.



Figure 3: EWA, EEWA, and PSS model fits to the human data for average effort (a), intra-group variance (b), and payoff (c).

Next, we classified players as signalers if min-dist was above 0 for five consecutive rounds, as persistent signaling is more effective (Brandts et al., 2015). About 46% of humans were signalers, compared to 72% EWA, 78% EEWA, and 44% PSS agents. Most PSS agent signalers were PD types (64% compared to 17% for RD). Groups were then categorized based on amount of signalers and compared to assess how signaling influences group behavior. For simplicity, we classified groups as having 0-1 (i.e., -2), 2, or 3-4 (i.e., 2+) signalers. Most human groups were classified as 2 (55.5%), followed by -2 (27.7%), and 2+ (16.6%). PSS had a similar pattern (44%, 34%, and 22%, respectively). For EWA and EEWA, most groups were 2+ (74% and 82%), followed by 2 (19% and 16%), and -2 (7% and 2%).



Figure 4: Average effort for signaler groups with human data (a), and EWA (b), EEWA (c), and PSS (d) model data.

We used linear mixed effects models with round and signaler group as fixed effects, and players nested within groups
and groups as random effects. We report the most relevant interactions for comparison. For effort (Figure 4), human (a) -2 groups had the strongest positive trend and overtook the 2+ group (*Round* \* 2*Group* :  $\beta = -.07, t(1434) = -4.16, p < .001$  and *Round* \* 2+ *Group* :  $\beta = -.06, t(1434) = -2.58, p = .01$ ), EWA (b) 2+ groups had the strongest negative trend (*Round* \* 2 + *Group* :  $\beta = -.03, t(7994) = -3.63, p < .001$ ), and PSS (d) -2 groups had the weakest negative trend and overtook the 2+ group (*Round* \* 2*Group* :  $\beta = -.02, t(7994) = -4.98, p < .001$ , and *Round* \* 2 + *Group* :  $\beta = -.04, t(7994) = -7.33, p < .001$ ). PSS had the most similar patterns for effort across signaler groups.



Figure 5: Average payoff for signaler groups with human data (a), and EWA (b), EEWA (c), and PSS (d) model data.

For payoff (Figure 5), human (a) -2 had the strongest positive trend (*Round* \* 2*Group* :  $\beta = -.56$ , t(1434) = -2.31, p = .02 and *Round* \* 2 + *Group* :  $\beta = -1.33$ , t(1434) = -4.06, p < .001), EWA (b) 2+ groups had the weakest positive trend (*Round* \* 2 + *Group* :  $\beta = -.32$ , t(7994) = -3.38, p < .001), and PSS (d) -2 groups had the weakest positive trend (*Round* \* 2*Group* :  $\beta = .22$ , t(7994) = 4.98, p < .001, and *Round* \* 2 + *Group* :  $\beta = .38$ , t(7994) = 7.33, p < .001). For payoff, EWA and EEWA (no findings) were more visually similar to human data and PSS differed as payoff increased for groups with more signalers, suggesting effective signaling.

The signaler group results showed human groups with fewer signalers had higher effort and payoff, and Figure 6 shows this at group and individual levels. Coordination and response to signals was better with 0 (a) or 1 (b) signalers, and mixed with 2 (c) and 3 signalers (d). These groups suggest player behavior is complex and may involve changes in strategies. To explore underlying mechanisms of choice changes, we present an example group for EEWA and PSS.

In the EEWA group (Figure 7), one agent (P4) never varied its choice, two others made one change midway (P3) or at the last round (P1), and one made more than one change (P2). Effort and resulting low payoffs (b) show the "sticky choice" problem goes beyond no variance players. Choice attractions for agents 1 (c) and 2 (d) show the relationship between choice attractions and choice changes. One choice at-



Figure 6: Player effort for human groups with 0 (a), 1 (b), 2 (c), and 3 (d) signalers.

traction dominates, suggesting the weighted choice sampling was necessary for choice changes.



Figure 7: Agent effort (a) and payoff (b) for one EEWA group, with choice attractions for agents 1 (c) and 2 (d).

In the PSS group (Figure 8), there was more variation in choices and strategy competition. Agent 1 (P1) was a payoff dominant signaler that started at high effort, then ended on the minimum. Agent 3 (P3) was a risk dominant nonsignaler that frequently set the minimum, then increased effort choices. Payoffs (b) show the benefits of these strategy shifts, most notably when agent 3 started choosing higher. Agent 1 (c) had the highest starting utility for signaling, then min- and ave-strategies competed until the min-strategy won. Agent 3 (d) had higher utility for the min-strategy, then competed with the ave-strategy. PSS Model agents demonstrated dynamic behavior by shifting from starting strategies based on group dynamics, counterfactual thinking, and learning.

## Discussion

The complexities of coordination behavior and differences between models were only apparent through analyses at different levels of behavior. Results suggest analyses and modeling excluding individual and group level behavior(s) may



Figure 8: Agent effort (a) and payoff (b) for one PSS group, with strategy utilities for agents 1 (c) and 3 (d).

lead to faulty conclusions about behavior and underlying mechanisms. The PSS model took a step towards addressing these issues. It included features missing from previous models: player types, strategies, and player choice predictions. Although EEWA was the best fit to average behavior, PSS showed greater capability to approximate human behavior at average, group, and individual levels. PSS agents displayed interdependent behavior by switching strategies based on group behavior and learning, and model mechanisms allowed for explanation of each players behavior based on player choice predictions and strategy utility. However, there are several limitations. 1) Choice variation was approximated with arbitrary strategies, agents predicted their own choices to enable repeated choices, and signaling behavior was rigid. 2) The PSS model players were more sensitive to signaling and better able to coordinate, which corresponded with the literature, but not with the human data. 3) The player choice predictions were based on reaction choices and did not include forming beliefs about other's player types suggested in the literature. 4) Model comparisons were based on simulations and complexity was not punished, potentially giving PSS an unfair advantage. Overall, we highlighted issues related to levels of analysis, and the strengths and weaknesses of the PSS model can inform future work to better approximate and explain complex coordination behavior.

## Acknowledgments

This research was supported by the U. S. Air Force Research Laboratory's 711th Human Performance Wing, Cognitive Models Branch. The contents have been reviewed and deemed Distribution A. Approved for public release. Case number: AFRL-2022-1525. 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?* New York, NY: Oxford University Press.

- Bortolotti, S., Devetag, G., & Ortmann, A. (2016). Group incentives or individual incentives? a real-effort weak-link experiment. *Journal of Economic Psychology*, *56*, 60–73.
- Brandts, J., Cooper, D. J., & Weber, R. A. (2015). Legitimacy, communication, and leadership in the turnaround game. *Management Science*, *61*(11), 2627–2645.
- Byrne, R. M. (2016). Counterfactual thought. *Annual review* of psychology, 67, 135–157.
- Cachon, G. P., & Camerer, C. F. (1996). Loss-avoidance and forward induction in experimental coordination games. *The Quarterly Journal of Economics*, *111*(1), 165–194.
- Camerer, C. F. (2003). *Behavioral game theory: Experiments in strategic interaction*. New York: Russell Sage Foundation.
- Camerer, C. F., & Ho, T.-H. (1999). Experience-weighted attraction learning in normal form games. *Econometrica*, 67(4), 827–874.
- Costa-Gomes, M. A., Crawford, V. P., & Iriberri, N. (2009). Comparing models of strategic thinking in van huyck, battalio, and beil's coordination games. *Journal of the European Economic Association*, 7(2-3), 365–376.
- Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instancebased learning in dynamic decision making. *Cognitive Science*, 27(4), 591–635.
- Hough, A. R. (2021). Understanding and improving coordination efficiency in the minimum effort game: Counterfactual-and behavioral-based nudging and cognitive modeling. Unpublished doctoral dissertation, Wright State University.
- Hough, A. R., O'Neill, K., & Juvina, I. (2021). Counterfactual-based nudging and signaling promote more efficient coordination during group tasks. *Comprehensive Results in Social Psychology*, 1–27.
- 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.
- Leng, A., Friesen, L., Kalayci, K., & Man, P. (2018). A minimum effort coordination game experiment in continuous time. *Experimental Economics*, 21(3), 549–572.
- Luce, R. D. (1956). Semiorders and a theory of utility discrimination. *Econometrica, Journal of the Econometric Society*, 178–191.
- Mehta, J., Starmer, C., & Sugden, R. (1994). The nature of salience: An experimental investigation of pure coordination games. *The American Economic Review*, 84(3), 658–673.
- Offerman, T. (2002). Hurting hurts more than helping helps. *European Economic Review*, *46*(8), 1423–1437.
- Riechmann, T., & Weimann, J. (2008). Competition as a coordination device: Experimental evidence from a minimum effort coordination game. *European Journal of Political Economy*, 24(2), 437–454.
- Siegler, R. S. (1987). The perils of averaging data over strate-

gies: An example from children's addition. Journal of Experimental Psychology: General, 116(3), 250.

- Van Huyck, J. B., Battalio, R. C., & Beil, R. O. (1990). Tacit coordination games, strategic uncertainty, and coordination failure. *The American Economic Review*, 80(1), 234–248.
- Van Huyck, J. B., Battalio, R. C., & Beil, R. O. (1991). Strategic uncertainty, equilibrium selection, and coordination failure in average opinion games. *The Quarterly Journal of Economics*, 106(3), 885–910.

## A Quantum Walk Framework for Multialternative Decision Making

Jiaqi Huang (huangajq@iu.edu)<sup>1,2</sup>, Jerome Busemeyer (jbusemey@indiana.edu)<sup>1</sup>

<sup>1</sup> Department of Cognitive Science, Indiana University, 1001 E. 10th Street, Bloomington, IN 47405 USA

<sup>2</sup> Corresponding author.

**Keywords:** Computational modeling; Quantum cognition; Multialternative decision making;

## Introduction

Traditionally, evidence accumulation in multialternative decision making is modeled by the Markov random walk process (**MRW**) (Roe, Busemeyer, & Townsend, 2001; Usher & McClelland, 2001; Bhatia, 2013; Noguchi & Stewart, 2018, 2018). Despite their all-around successes, these MRW models are challenged by recent evidence of Markov violations in evidence accumulation including interference effects of choice on confidence for multistage decision making (Kvam, Pleskac, Yu, & Busemeyer, 2015), interference effects of confidence on confidence (Busemeyer, Kvam, & Pleskac, 2020), and order effects in experimental test of attraction effects (Trueblood & Dasari, 2017).

On the other hand, the quantum walk process (**QW**) (Busemeyer, Wang, & Townsend, 2006; Wang, Solloway, Shiffrin, & Busemeyer, 2014) explain these Markov violations in a natural way. However, QW models have only been applied to binary alternative decision making, and this raises the questions of whether we can extend existing QW models to explain both Markov violations and traditional context effects in multialternative decision making. Our goal here is to present a general framework for this potential extension.

#### Quantum walk model for binary alternative

Quantum walk (**QW**) is the quantum analogy of Markov random walk (**MRW**) which, instead of describing the time evolution of an initial probability distribution, describes that of an initial probability amplitude distribution. The quantum time evolution is governed by Schrödinger's equation:

$$\frac{d}{dt}\psi(x,t) = -i \cdot H \cdot \psi(x,t), \qquad (1)$$

where  $\psi$  is the probability amplitude distribution (quantum wavefunction), and *H* is the Hamiltonian operator in analogous with the Markov transition rate matrix. For discrete-finite-state quantum walk, *H* can be written in the following  $N \times N$  matrix form:

$$\begin{cases} H(i,i) = u(i), & \text{for } 1 \le i \le N \\ H(i+1,i) = H(i,i+1) = \sigma^2, & \text{for } 1 \le i \le N-1, \end{cases}$$
(2)

where u(x) denotes the potential function, and  $\sigma^2$  is the diffusion rate that describes the effect of a constant nonconservative force acting on the system. The solution to Schrödinger's equation gives:

$$\Psi(x,t) = U^t \cdot \Psi(x,0) = e^{-i \cdot H \cdot t} \Psi(x,0), \qquad (3)$$

where  $U = e^{-i \cdot H}$  denotes the quantum unitary operator. In binary alternative decision making problem to which QW is previously applied (Busemeyer et al., 2006),  $\psi(x,t)$  can be viewed as a probability amplitude distribution over the confidence states.  $u(x) = \alpha x + \beta$  is modeled by a linear function with drift parameter  $\alpha$ .

#### Multi-alternative quantum walk framework

The multi-alternative quantum walk model (**MQW**) is inspired by the existing QW model for binary choice decision making and MRW models that use multiple accumulators to explain multialternative decision making. MQW is defined by (1) initial state, (2) Hamiltonian that describes how the initial state evolves (3) stopping conditions.

**Initial state** Suppose there are  $N \ge 3$  alternatives to choose from, we define *N* initial states with  $\psi(x, 0)_m$  being the state for the *m*th alternative. The aggregated initial state is written as a direct sum:

$$\Psi(x,0) = \bigoplus_{m=1}^{N} w_m \cdot \Psi(x,0)_m, \tag{4}$$

where  $w_m$  with  $\sum_{m=1}^{N} |w_m|^2 = 1$  models the attention weights to each alternative. By the definition of direct sum, if each  $\psi(x,0)_m$  is of dimension  $P \times 1$ , then  $\psi(x,0)$  will have dimension  $NP \times 1$ .

**Hamiltionian** To describe the time evolution, for each of the alternatives denoted as  $A_m$ , we define *N* Hamiltonian matrices, where the *q*th of such denoted as  $H_{m,q}$  has dimension  $P \times P$ . This  $H_{m,q}$  represents how evidence accumulation of alternative  $A_m$  influences evidence accumulation of alternative  $A_q$ , and thus can be used to model context effects in multialternative decision making. For example, in the case of similarity effects, accumulating evidence in favor of  $A_m$  inhibits accumulating evidence in favor of the similar alternative  $A_q$ , and thus time evolution described by  $H_{m,q}$  will decrease  $A_q$ 's confidence rating. In cases when  $A_m$  and  $A_q$  are independent,  $U_{m,q} = e^{-i \cdot H_{m,q}}$  will be the identity matrix, and  $H_{m,q}$  will thus be the zero matrix.

According to equation 2, we need to define a potential function  $u_{m,q}(x)$  and a diffusion rate  $\sigma_{m,q}$  for each  $H_{m,q}$ . Similar to Busemeyer et al. (2006), we make  $\sigma_{m,q}$  a free parameter, and  $u_{m,m}(x) = \alpha_{m,m}x + \beta_{m,m}$  a linear potential function with free parameters  $\alpha_{m,q}$  and  $\beta_{m,q}$ . To further constrain the number of parameters, we make  $\sigma_{m,q} = \sigma_q$  for each alternative  $A_q$ , which means that the diffusion effect on  $A_q$  is independent of  $A_m$ . In the most general case, there will be a



Figure 1: Mean confidence ratings as a function of time of an example model with three choices. Mean confidence ratings are computed as the mean of the confidence rating distribution at each time point T. The three choices are assumed to be independent (all  $U_{m,q}$  for  $m \neq q$  are identity matrices). The two stopping conditions are illustrated by the red lines. Mean confidence as a function of time of an example Markov model for choice 1 is also plotted. Compared with the Markov model, the quantum model shows an oscillating mean confidence rating.

total of (2N+1)N parameters to fit, though in many cases the model can be further constrained.

With these  $H_{m,q}$ , we then define the general Hamiltonian H with dimension  $NP \times NP$  for the entire system as

$$H(\varepsilon) = \prod_{i=1}^{N} H_{\varepsilon(i)} = \prod_{i=1}^{N} \bigoplus_{q=1}^{N} H_{\varepsilon(i),q},$$
(5)

where  $\varepsilon$  is a permutation function of the *N* alternatives (eg.  $\varepsilon(1) = 2$  means the alternative  $A_2$  evolves first),  $H_m = \bigoplus_{q=1}^{N} H_{m,q}$  is the  $NP \times NP$  Hamiltonian for alternative  $A_m$  in the direct sum space.

Since the  $H_m$  matrices may not commute with each other when multiplied, the general Hamiltonian matrix H would have been different for different permutations  $\varepsilon$ . Thus, MQW can explain order effects by using different  $\varepsilon$  to model the different orders of presentation of the alternatives. Besides, since MQW directly inherits from QW, it is also capable of explaining interference effects as QW does in the binary choice case.

**Stopping condition** Finally, we need to define two stopping conditions in analogous to that in multi-alternative decision field theory (Roe et al., 2001). The first stopping condition is a common boundary defined by some  $P \times P$  projection matrix for each alternative to compute response time distribution without time pressure, and the second condition models how people choose among the alternatives under time pressure (see Figure 1). To compute the choice probability, we first define random variables  $C_m$  that describes the current

confidence level of the alternative  $A_m$ , and C as the set of all such  $C_m$  for each alternative. The choice probability of  $A_m$  at time T is then computed as

$$P(A_m|T) = P(C_m = max(C))$$
(6)

Conceptually, the above means that the probability of choosing  $A_m$  is the probability that  $A_m$  is the most confident alternative to be chosen at time T.

#### **Future works**

Despite the benefits of MQW framework in predicting jointly Markov violations and context effects, we acknowledge that this framework has not yet been fully adapted to multialternative decision making. Future works are needed to define a model that builds on this framework and connects its parameters with the subjective values of different attributes of the alternatives and expected utilities of the multiple alternatives.

## Acknowledgments

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-20-1-0027.

#### References

- Bhatia, S. (2013). Associations and the accumulation of preference. *Psychological review*, 120(3), 522.
- Busemeyer, J. R., Kvam, P. D., & Pleskac, T. J. (2020). Comparison of markov versus quantum dynamical models of human decision making. *Wiley Interdisciplinary Reviews: Cognitive Science*, 11(4), e1526.
- Busemeyer, J. R., Wang, Z., & Townsend, J. T. (2006). Quantum dynamics of human decision-making. *Journal* of Mathematical Psychology, 50(3), 220–241.
- Kvam, P. D., Pleskac, T. J., Yu, S., & Busemeyer, J. R. (2015). Interference effects of choice on confidence: Quantum characteristics of evidence accumulation. *Proceedings of the National Academy of Sciences*, *112*(34), 10645– 10650.
- Noguchi, T., & Stewart, N. (2018). Multialternative decision by sampling: A model of decision making constrained by process data. *Psychological review*, *125*(4), 512.
- Roe, R. M., Busemeyer, J. R., & Townsend, J. T. (2001). Multialternative decision field theory: A dynamic connectionst model of decision making. *Psychological review*, 108(2), 370.
- Trueblood, J., & Dasari, A. (2017). The impact of presentation order on the attraction effect in decision-making. In *Cogsci.*
- Usher, M., & McClelland, J. L. (2001). The time course of perceptual choice: the leaky, competing accumulator model. *Psychological review*, 108(3), 550.
- Wang, Z., Solloway, T., Shiffrin, R. M., & Busemeyer, J. R. (2014). Context effects produced by question orders reveal quantum nature of human judgments. *Proceedings of the National Academy of Sciences*, 111(26), 9431–9436.

## Using GOMS to Model Individual Differences in a Competence Assessment Task

Hadeel Ismail (hi71@sussex.ac.uk)

Department of Informatics, University of Sussex Brighton, BN1 9QJ, UK

Peter C-H. Cheng (p.c.h.cheng@sussex.ac.uk)

Department of Informatics, University of Sussex Brighton, BN1 9QJ, UK

#### Abstract

This study aims at modelling individual differences using GOMS. In an attempt to evaluate a competence assessment task in natural language, results revealed limitations of a previous GOMS model that was used to design the task (Ismail & Cheng, 2021). The task, Chunk Assessment by Stimulus Matching (CASM), exploits measurements of chunk signals to assess competence in the English language. It was tested with 34 speakers of English as a second language. Results were compared against the initial GOMS models. The models' predictions were partially supported, showing substantial performance differences between the levels of expertise. Contrary to expectations, major differences were found amongst those at the same level of expertise. A refinement of the models was built to coherently capture differences between and within levels of competence.

**Keywords:** chunking; GOMS; individual differences; language competence; model evaluation; pause analysis

#### Introduction

In cognitive science, pauses in recall and copying tasks have long been associated with mental processes. Some studies have specifically examined how pauses might reflect aspects of an expert's memory and inspired others to examine how these temporal measures might distinguish between experts and novices in specific domains. The classic studies performed by Chase and Simon (1973) involved memory and perceptual tasks for replicating item positions on a chessboard. Their findings reveal that expert's ability to remember far more positions than novices with close to perfect replications in memory tasks, and returning less to view the stimulus during copying tasks. Their observed physical actions are typically explained by the chunking theory (Cowan, 2001; Gobet et al., 2001; Miller, 1956). In short, the theory clarifies that during the process of perceiving domain-specific information, the amount held in working memory (WM) is dependent on the individual's representation of related information in their long-term memory. The more knowledgeable a person is, the richer their representation, which in turn assists in perceiving large chunks of meaningful information. Therefore, an expert's knowledge overcomes the limitations of WM providing them with the advantage of having larger chunks that encode more units of information than a novice.

Chunking theory informed the studies of Cheng and colleagues. They observed individuals' hand transcriptions of mathematical equations (Cheng, 2014; Cheng & Rojas-Anaya, 2007), English sentences (Zulkifli, 2013), and



Figure 1: CASM task (Ismail & Cheng, 2021) – text in red are explanatory labels about the interface.

programming scripts (Albehaijan & Cheng, 2019) using behavioral measures that included pauses, writing durations, and the number of times a stimulus is viewed. These microbehavioral measures, captured at a millisecond timescale, show some potential as metrics for assessing competency.

Rather than logging pen strokes, a recent paper proposed the *Competence Assessment by Stimulus Matching technique*, CASM, that utilizes the mouse device in word matching tasks (See Fig.1), in order to assess competence in the English language (Ismail & Cheng, 2021). The researchers used GOMS cognitive task analysis (Card, Moran, & Newell, 1983) to help in designing the tasks for CASM. In particular, GOMS was used to design tasks that would promote expert's use of the chunks, available from their superior knowledge, in order to maximize their performance compared to novices and thereby provide strong measures of competence. From the GOMS models of alternative tasks designs the ones with the largest theoretical differentiation across level of competence were adopted.

The empirical evaluation, to be summarized in the third section, shows substantial performance differences between the levels of expertise. Critically, it also revealed a limitation of the original GOMS models. Major differences were found amongst those at the same level of competence. The aim here is to build models that more coherently capture differences between and *within* levels of competence. In a sense, this study challenges the claim that GOMS does not take into account individual differences (Olson & Olson, 1990).

One motivation for this work is to continue developing CASM by controlling for sources of individual differences unrelated to competence, in order to improve the quality of

	Stimuli presentation		
Stimulus-response	Constant	Voluntary	
matching	display, CD	display, VD	
Word to word, WW	CDWW	VDWW	
Part-word to word, PW	CDPW	VDPW	

Table 1: The four designs of the CASM task

the CASM measures. We are following Gong & Kieras's (1994) general advice to include GOMS modelling as part of our iterative cycle for system development. So, the refinement of the models to encompass a range of individual differences are examined in the fourth section.

## **Design of the CASM Task**

CASM attempts to assess an individual competence in terms of the chunks they possess. The basic idea is to log an individual's' interaction as they decide if given stimuli words correctly match corresponding words in the response group (Fig.1). Individuals must quickly and accurately compare and click their responses. The task is designed in a way that encourages the use of chunking, this includes a separation between the stimuli and the responses and the non-alignment of the two which motivates participants' use of a strategy involving recognizing and remembering the words. It is assumed that depending on an individual's level of English competence, the number of words held in WM would be manifested in their behavior. An expert's prior knowledge would provide them with the advantage of quickly recognizing words and capturing multiple words into their WM, whereas a novice's limited knowledge would constrain their chunking thus forcing them to refer back to the stimuli more times than an expert. Such differences are reflected in the length of pauses preceding their clicks as well as their pause patterns.

Since the design space of the CASM task was large, GOMS analysis was applied to find tasks that maximize the differentiation between experts and novices (Ismail & Cheng, 2021). Two manipulations of stimuli visibility and two types of stimuli to response matching were proposed (see, Table 1). The model predicts that across all four tasks, expert's pauses would be shorter than a novice, with higher differentiation in *part-word to word* (PW) tasks compared to *word to word* (WW) tasks of the same presentation condition (Table 2, top).

In the *constant display* condition (CD) the stimulus and the response items remained visible throughout the trials. The duration of pauses between clicks reflects differences in chunking and hence is a potential measure of competence. In the *voluntary display* condition (VD), the stimulus and the response items were not simultaneously displayed. On loading the screen, the response items (bottom) were made visible with the stimulus (top) remaining concealed under an interactive grey box. A hover of the mouse pointer over the box will reveal the stimulus and mask the response items. Upon hovering away to mark their responses, the stimulus and the response will revert to their initial visibility states. Participants were allowed to hover over the stimulus as many times as they needed. With the VD condition, measures of chunking

Table 2: Compares the original model's median of pauses
prediction (Ismail & Cheng, 2021) to the observed group
median of pauses for the five least and most competent

-	Model predictions (msc)			
Task type	Novice	Expert	Difference	
CDWW	2275	1070	1205	
VDWW	3205	1070	2135	
CDPW	3175	1120	2055	
VDPW	4950	1120	3830	
	Expe	rimental re	sults (msc)	
Task type	Low	High	Difference	
CDWW	2971	1725	1246	
VDWW	3946	1987	1959	
CDPW	4632	2210	2422	
VDPW	5430	2240	3190	

include the number of hovers made to view the stimulus and the duration of time spent clicking between views.

These display conditions were combined with two matching conditions: word to word (WW) or part-word to word (PW) matching tasks. The WW condition consisted of matching whole words in the stimulus with whole words in the response (Fig.1). Since novices might not be familiar with many of the words presented, their basic strategy in matching WW is expected to consist of decomposing a word into parts that are separate chunks, thus filling their WM capacity more quickly than for experts. In PW condition, each word in the stimulus was broken up and presented as a string of syllables with equal spacing between them and the following word's syllables. For example, a stimulus containing the words "indict meringue aardvark" would be presented as "in dict me ringue aard vark" and these, as with the WW condition, were matched with complete words in the response. In this PW task it is assumed that an expert, at a slower pace than in WW, would still be able to recognize and chunk whole words. However, the PW task might encourage a novice to chunk one syllable at a time, thus switching many more times between the stimulus and the response prior to making a matching decision and clicking.

## **Empirical evaluation of CASM tasks**

To assess the model predictions, an empirical evaluation of CASM tasks was carried out.

#### Method

The experiment is a within-subject design. It was approved by the University of Sussex Science School's ethics committee.

**Participants.** The participants were 34 adults, eight males, and twenty-six females, whose ages ranged between 18 to 54 years old. They were recruited on the basis that they spoke Arabic as their first language and English as a second, but with varying degrees of competence in English.

**Materials**. The experiment involved three stages. The first was a questionnaire that gathered participants' background, general ability, and confidence in using the English language.

The second was a generic vocabulary size test that assessed their overall level of competence. Scores gathered from the first two stages determined the participant's overall level of English language competence. The final stage was the CASM task which included the four conditions in Table 1. Each condition started with a set of instructions followed by three practice sessions and then twelve trials. Each trial consisted of eight words. The level of trial difficulty was determined by the frequency of the target words and their number of syllables. There were four types of word frequency, ranging from high to low, and three syllabic levels that included two, three, and four-syllable words. The order of the conditions received by the participants was counterbalanced, and the trials within were presented in random order.

**Procedure.** All of the materials were delivered online and were run on their personal computers using their own mouse. They had the option to complete all stages in one or up to three sittings. However, once a stage has been launched, it must be entirely completed without any interruptions. Specific instructions were given at the start of the CASM task which included matching the words as quickly and as accurately as possible and refraining from removing their hands from the mouse unless instructed otherwise.

In order to examine expert vs novice performance, participants were rank ordered according to their independent measure of competence. A systematic check was applied showing the top and bottom five individuals being reasonably consistent and thus were chosen to represent the extremes.

## **Experimental Results**

To compare the performance of the five highest and lowest competent individuals (HC & LC), a group median was calculated using the mean pause of each participant. Each participant's mean pause was calculated from the median pause for each of their trials.

Across the four tasks substantial differences in the pauses exist (Table 2, bottom). This confirms the original model's prediction of pause lengths decreasing with increasing competence (Table 2, top). Moreover, PW tasks seem to have a higher differentiation effect compared to WW tasks of the same display conditions, in line with the predictions of the model (4<sup>th</sup> column in Table 2). Finally, the difference in pauses between HC and LC participants were comparable to the predictions of the model with an absolute difference of 20% or less (4<sup>th</sup> column in Table 2). According to HCI heuristics, an engineering model is acceptable if it reaches a level of accuracy of at least 80% (John & Kieras, 1994). Contrary to expectations, the absolute pause times were underestimated for both HC and LC individuals, with a level of accuracy as low as 51% (2<sup>rd</sup> & 3<sup>rd</sup> column in Table 2).

The divergence between human performance times and that predicted by the model indicate processes that the original model failed to foresee. This is likely due to variations in participants' strategies. Information concerning their patterns of clicks and hovers in the VD condition allowed us to carry a detailed examination of their strategies. The results show that there are intra-participant strategy differences, Fig. 2. The figure displays a selection of participants from both groups in the VDWW and VDPW conditions. Clear differences in terms of the number of hovers and pattern of clicks exist both at the level between and within groups.

In tasks involving WW matching, the basic assumption made by the original GOMS model is that novices would follow a *single-view-single-pick* strategy (Fig. 2, P48-A) which involves chunking one word during one hover/view of the stimuli, making a comparison and then clicking an answer (Ismail & Cheng, 2021). However, a *multi-view-single-pick* strategy was sometimes applied, where a single click is preceded by several hovers (e.g., Fig. 2, P48-B). One explanation for such behaviour is that LC individuals are uncertain of the chunked item in memory and go back to the stimuli for further verification prior to giving an answer.

In terms of PW matching, the original GOMS model assumed that since words were presented as parts, then novices might find it more convenient to follow a *multi-view-singlepick* strategy by chunking one syllable at a time, comparing each part of a word separately until reaching a decision (Ismail & Cheng, 2021). This would imply that in VDPW, the number of hovers made prior to clicking an answer would equal to the number of parts the word is divided into. The experimental results pertaining to the least competent



Figure 2: Various strategies applied by the participants across and within groups

individuals reveal far more complex strategies than what was originally assumed and they are:

- Single-view-single-pick strategy; indicating their ability to group the parts of one word in one view of the stimuli (Fig. 2, P43-C).
- 2- *Multi-view-single-pick* strategy; conforming to the model's overall prediction but differing in terms of the number of hovers made (Fig. 2, P31-D, E, F). The number of hovers might either be less than, more than, or equal to the number of word parts. Such behavior may be explained in terms of one or a combination of the following:
  - a. Inability to group all parts of a word in one view.
  - b. Difficulty in locating word boundaries impacting their word recognition process
  - c. Uncertainty of the word chunked in WM.

Experts were assumed to follow a *single-view-multi-pick* strategy by consistently loading large chunks of words into WM (e.g., P39 in Fig. 2). However, consistency varied, sometimes high competent (HC) participants would engage in a *single-view-single-pick* strategy, similar to that of a novice (Fig, 2, P50-G). Other times, they would apply an alternating strategy (Fig. 2, P38-H). Such variations could imply a lack of motivation in maximizing their use of WM.

There were instances where HC and LC participants would both engage in a *multi-view-multi-pick recoding* strategy, a purely strategic tactic that does not reflect competence (e.g., green highlights in Fig. 2). The initial GOMS models assumed that, across both extremes, once a word in WM is compared to the response item, the process is immediately followed by a mouse click (Ismail & Cheng, 2021). However, by applying a recoding strategy, participants might generate a list of decision codes in their memory by hovering over the stimuli, chunking a word or so, hovering away to reveal the responses, making a comparison, encoding their decision and then proceeding to process the next word in the stimuli without clicking an answer pertaining the first word(s). This would continue for a few times until enough codes have been loaded into WM, only then would they proceed to click multiple answers at once.

Although the experimental results support the predictions of the initial model (Ismail & Cheng, 2021) in terms of overall pause differences between the experts and the novices. Findings reveal that the strategies applied are far more complex than the previous model, thus allowing for individual differences within groups to arise.

## **GOMS Models**

Based on the HC and LC individuals' performances, two models were generated that cohesively account for the different individual strategies that exist between and within the expert and novice groups (Fig. 3 and Fig. 4). The models represent the processes when working under the CD condition in PW and WW matching tasks. Overall, the models are divided into two parts, everything prior to the process "move eye to response area" concerns chunking processes, and everything there after deals with comparing, matching and mouse moving processes. The new models are more complex than the original attempt as they encompass individual differences at all levels. The green dashed lines in the figures point to WW processes, the purple dashed lines are associated with PW, while the black solid lines are those shared by both tasks. It is worth noting that the overall construct of the models under the CD condition is similar to the VD condition with the exception of having a hover over/away action whenever alternating views between the stimuli and the responses. The models explain the chunking process in terms of nested loops. The novice and the expert models differ in the number of loops and the type of processes contained within each loop (see the orange brackets in Fig. 3 and Fig. 4). This explains the inter-participant differences. Moreover, not all loops are experienced by all members of the same group, which explains for the intra-participant differences.

The first loops in both models concern the chunking process. In WW tasks novices break each word into its parts, individually processing them until a whole word is recognized and captured in WM (Fig. 3, NLP1(WW)). Experts have the ability to immediately recognize a word and capture it in memory, thus looping around ELP1(WW) (Fig. 4) as many times to generate a chunk of words. In PW tasks, the presentation of the words slows down the recognition process. Experts now experience two loops when chunking. The ELP1A(PW), shows how an expert must process enough syllables until a word is recognized. The second loop ELP1B(PW) explains the forming of a chunk of words. Novices on the other hand seem to experience much more difficulty, as their NLP1(PW) loop is more complex by including "the boundary confusion" decision process. Since the stimuli presents the words with equal spacing between all syllables, novices might find it difficult to distinguish word boundaries. With this added level of complexity, novices might not be able to chunk a whole word in one view causing them to return to the stimuli as many times as needed until a whole word is successfully recognized. Moreover, novices might be uncertain during the process of comparing the chunked item with the response word, and may wish to verify their answer prior to clicking. This in turn introduces the NLP2(Both) loop, that gives another explanation for their returns to the stimuli.

These differences across groups, in both WW and PW, show how novices experience increased cognitive effort in processing the presented words limiting their chunking ability, and causing them to perform multiple returns to the stimuli, therefore experiencing many long pauses between clicks. In contrast, experts have the opportunity to chunk more than one word per view, thus demonstrating shorter pauses between clicks (Table 2).

Within these two groups individual differences were found, which could be explained by the number of times individuals choose to go through the loops depicted in the models. For instance, in WW matching tasks, if a word is very familiar to a novice, then they might apply a *single-view-single-pick* strategy therefore bypassing the NLP2 loop in Fig.3. Otherwise, an unfamiliar word would produce a *multi-view-single-*



Figure 3: Novice Model

*pick* strategy by entering the NLP2 loop as many times as needed until certainty is attained.

In PW matching tasks, the variations in the number of stimuli views, as seen in Fig 2. P31(D,E,F) are mainly explained by two loops; NLP1(PW) and NLP2 in Fig. 3. If a novice finds it difficult to locate word boundaries, then a process of chunking one or more syllables without reaching a complete word might be applied. This means that at any point in time they might choose to opt out of NLP1(PW) and proceed to compare the chunked parts via NLP3, then loop back to the stimuli via NLP4 and continue on in this process until the whole word is compared. Another explanation, as shown earlier, might be due to uncertainty and thus entering the NLP2 loop. The NLP1(PW) and NLP2 might be experienced as many times as needed in a manner that includes either one or both of them until a response is provided. This is then reflected in their number of views and pause lengths.



Figure 4: Expert Model

According to Fig. 2, experts mainly varied amongst each other in the number of words chunked into their WM. This is due to their preference of opting out of ELP1(WW) and ELP1B(PW) in Fig. 4 at any point in time without fully loading their WM. This might be due to the nature of the task which did not put a premium on loading WM to capacity as much as possible

Finally, the recoding strategy observed in the performances in both groups can be explained by the "code" decision process (see the blue colored flow of processes in Fig. 3& 4). If the participant finishes comparing the memorized item to the response, they might choose to recode their decisions rather than clicking answer, thus viewing the stimuli multiple times, comparing and then generating a list of codes. Once enough codes have been produced, they would enter the NLP5 (Fig. 3) or ELP3 (Fig. 4) loop by simply retrieving one decision code at a time and clicking their options.

The models produce a range of pause durations based on the type and number of loops encountered. To evaluate the



Figure 5: Model evaluation

predictions (Fig. 5), an expert's max and min pauses were calculated based on chunk size (ELP1(WW) & ELP1B(PW) in Fig. 4). By observing the HC participant's performances, their chunk size ranged between one to three words per view, and thus were chosen to represent the expert's boundaries. Consisted with the LC's performance, a novice's chunk size was limited to one word, however the number of times entering the confusion or uncertainty loops are what determined the novice's model limits (NLP1(PW) & NLP2(both) in Fig. 3). Therefore, the min value was set at no entry to those loops, while the max was based on encountering each loop once for every word processed. In Fig. 5, the solid lines indicate to the expert and HC data, while the dotted represent the novice and LC. Moreover, the black lines represent the original model, the dark point to the min and max boundaries of the new model, and the light lines represent the experimental data.

Results show that the original model was always at the lower bound of the range of participants, while the modified model encompasses much more of the range, excluding some of the LC's min values and a few of the HC's max values.

#### Discussion

We are developing the Competence Assessment by Stimulus Matching (CASM) technique for the assessment of competence in natural language that exploits measurements of chunk signals. A summary of an empirical evaluation of the CASM was presented and compared against the initial GOMS models (Ismail & Cheng, 2021) used to design CASM tasks. The models' predictions were partially supported. Overall, high competent individuals experienced shorter pause durations prior to clicking answers and made a smaller number of stimuli views compared to less experienced counterparts. This likely reflects the different chunk structures between the two groups, conforming to the chunking theory (Cowan, 2001; Gobet et al., 2001; Miller, 1956). Moreover, the results are in line with previous studies that used hand transcription tasks to measure competence in various domains (Albehaijan & Cheng, 2019; Cheng & Rojas-Anaya, 2007; Zulkifli, 2013).

However, the experiment revealed major intra-participant differences otherwise not captured by the initial GOMS model. To address the limitation, we examined the participants' strategies when interacting with the VD tasks. Three main observations were made.

First, low competent individuals differed amongst each other in their number of views. This might have been caused by either uncertainty of items held in WM, or inability to group the syllables into a word causing participants to loop the associated processes in the model (Fig. 3) as many times as needed until a word is recognized. Both cases are explained by an absence of chunks pertaining these words in long term memory. The weaker the chunk, the higher the chance of these loops occurring, causing an increased number of views.

Second, experts' ability to load a large number of words into WM, reflects the existence of those chunks in their longterm memory. However, some participants in this group did not fully utilize their WM, by limiting their chunking following a *single-view-single-pick* strategy. This is caused by not performing enough loops in the initial processes pertaining to word recognition and chunking (Fig. 4) The main aim of CASM is for experts and novices to be loading WM to the same extent with numbers of chunks, so that they are comparable in that regard, but what differs between them is the size of the chunks, which will be larger for the experts than the novices, hence a better performance.

Third, participants across groups applied a recoding strategy, that reflects nothing of their language ability. Following this strategy, the information contained within their chunks is a code of their potential responses rather than the words themselves. Such could assist the participants in managing their working load, as they drop information pertaining the words early on and retain a much easier to memorize code.

The evaluation results show that the new model, though not producing perfect matches, out-performs the original one. As for the out-of-range values, it was observed that the low competent individuals, in many instances, were employing the recoding strategy, which was not modeled in Fig. 5. However, there is no observed explanation for the high competent, but we hope to find out in subsequent experiments.

The revision of the GOMS models provided for a better understanding of the sources that caused the inter and intraparticipant differences. This is helpful for the future refinement of the CASM task in at least two ways. First, to increase the demands of the task to encourage individuals to load up their WM, hence use their chunking ability more. Second, to eliminate the possibility of individuals applying a recoding strategy. We are modifying the design CASM tasks.

From a wider perspective, this study took an incremental step towards using GOMS to develop a model that includes various individual differences, which challenges the claim made by Olson and Olson (1990). Therefore, a particular contribution of this work is the demonstration how, in one way, GOMS models may address individual differences.

## References

Albehaijan, N., & Cheng, P. C.-H. (2019). *Measuring* programming competence by assessing chunk structures in a code transcription task.

- Card, S., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction*: Crc Press.
- Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive psychology*, 4(1), 55-81.
- Cheng, P. C.-H. (2014). Copying equations to assess mathematical competence: An evaluation of pause measures using graphical protocol analysis. *Proc. of the 36th Annual Conf. of the Cognitive Science Society*, 319-324.
- Cheng, P. C.-H., & Rojas-Anaya, H. (2007). *Measuring mathematic formula writing competence: An application of graphical protocol analysis.* Paper presented at the Proceedings of the Annual Meeting of the Cognitive Science Society.
- Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behav Brain Sci*, 24(1), 87-114.
- Gobet, F., Lane, P. C., Croker, S., Cheng, P. C., Jones, G., Oliver, I., & Pine, J. M. (2001). Chunking mechanisms in human learning. *Trends in cognitive sciences*, 5(6), 236-243.
- Ismail, H. B., & Cheng, P. C.-H. (2021). Competence assessment by stimulus matching: an application of GOMS to assess chunks in memory. Paper presented at the proceedings of the 19th ICCM Conference.
- John, B. E., & Kieras, D. E. (1994). *The GOMS family of analysis techniques: Tools for design and evaluation*. Retrieved from
- Miller, G. A. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *Psychological review*, 101(2), 343-352.
- Olson, J. R., & Olson, G. M. (1990). The Growth of Cognitive Modeling in Human-Computer Interaction Since GOMS. *Human-computer interaction*, 5(2-3), 221-265.
- Zulkifli, M. (2013). Applying pause analysis to explore cognitive processes in the copying of sentences by second language users. In: ProQuest Dissertations Publishing.

# **Cognitive and Motivational Effects in Peer-Assisted Learning**

Ion Juvina<sup>1</sup> (ion.juvina@wright.edu), Jarean Carson<sup>1</sup> (carson.76@wright.edu), Preston Menke<sup>1</sup> (menke.21@wright.edu), & Peter Crowe<sup>1,2</sup> (peter.crowe14@gmail.com)

> <sup>1</sup>Wright State University, Dayton, OH 45435 USA <sup>2</sup>Bellese Technologies, Owings Mills, MD 21117 USA

#### Abstract

Peer-assisted learning has the potential to improve learning in academic settings and beyond. However, the cognitive and motivational effects of learning through interaction with other learners are not fully understood. Here we present an empirical study in which we compare a peer-assisted learning condition with two individual learning conditions. The empirical findings suggest that both positive and negative peer effects may be occurring. A computational cognitive model developed in the ACT-R cognitive architecture is presented and used to explain some of the mechanisms of peer-assisted learning.

Keywords: peer-assisted learning; cognitive modeling

## **Introduction and Background**

Active learning pedagogies characterized by interaction among learners have recently demonstrated some potential to improve learning outcomes. For example, Ulrich, Brewer, Steele-Johnson, Juvina, Peyton, & Hammond (2017) found that implementing team-based learning (TBL) and other active learning pedagogies at a Midwestern university raised the scores on national standardized tests from below to above national averages. While evidence like this is encouraging, it often comes from field studies or classroom quasi-experiments that are notoriously difficult to interpret and replicate. Therefore, there is a need for controlled (and realistic) laboratory experiments in this area.

Humans have an unmatched ability to acquire new knowledge. Some of this learning occurs through interaction with other learners (Rendell, Fogarty, Hoppitt, Morgan, Webster, & Laland, 2011). Under the assumptions that knowledge is unevenly distributed in the population of learners and different learners have different learning experiences, interaction among learners provides opportunities for exchanging knowledge, filling the knowledge gaps in learners' minds, and even creating positive feedback loops that increase the amount of shared knowledge - an effect known in economics as knowledge spillover (Phelps, Yang, & Steensma, 2010).

Besides knowledge, engagement and motivation to study can be increased by interaction among learners, through mechanisms such as *social facilitation* (Guerin & Innes, 1984; Zajonc & Sales, 1965) and *positive peer pressure* (Smith & Fowler, 1984). The perceived presence of peers can increase affective arousal (Geen & Gange, 1977), induce a sense of responsibility for learning (Koles, Stolfi, Borges, Nelson, & Parmelee, 2010), or *nudge* individuals toward higher levels of effort (Hough, O'Neill, & Juvina, 2021; Horton & Zeckhauser, 2016). However, learning from other learners may have negative consequences as well. For instance, individual learners may have incomplete, erroneous, or biased knowledge, which may be compounded by interaction among learners. To mitigate this risk, learners need to learn not only the instructional content but also who to trust among their peers (Collins & Juvina, 2021), so they can filter the information they receive from their peers, that is, learn from trusted peers and ignore or discard information from untrusted peers (Collins, Juvina, & Gluck, 2016). This learning about other learners adds cognitive load to the existing load of learning a particular material.

The presence of peers and peer interaction may add *ambient noise* that may further increase the attentional and cognitive load of peer-assisted learning. For example, Hoxby (2000) showed that boys and girls learn more when there is a larger share of girls among the students in a classroom, an effect attributed to the tendency of girls to be less disruptive to classroom learning activities. More generally, in work environments that require a high level of concentration, participants report higher levels of distraction and stress in open-plan offices as compared to cell offices (Seddigh, Berntson, Bodin Danielson, & Westerlund, 2014).

From a motivational perspective, learners may become too reliant on other learners and less inclined to exert sufficient effort individually to develop and maintain their knowledge base, an effect known in psychology as *social loafing* (Karau & Williams, 1993). Social loafing is a general finding across many types of tasks and subject populations; it occurs even in interventions designed to eliminate or minimize the effect (Karau & Williams, 1995).

An additional risk of learning in the presence of others is *negative peer pressure*. In environments where there are rewards for learning that arise from how one ranks among their peers, learners are made worse off by the studying efforts of their peers, and thus they tend to discourage and punish their peers' learning efforts (Bishop, 2003, 2006).

The work reported in this paper aims to uncover some of the cognitive and motivational mechanisms that underlie peer-assisted learning through a combination of empirical experimentation and computational cognitive modeling. Do learners take advantage of their peers' knowledge to increase or consolidate their own knowledge? Are they more or less willing to exert learning efforts when they are placed in a peer-interaction condition? Does their learning suffer from increased cognitive load or interference from their peers' incorrect knowledge? These are the main research questions addressed here. The first part of this paper presents an in-depth analysis of data from an empirical study that contrasted a peer-assisted learning condition to two control conditions: an individualactive condition and an individual-passive condition. The second part presents a computational cognitive model that explains some of the effects presented in the first part. An extended version of this paper that includes more details on both the empirical study and the cognitive model will be submitted for publication to a journal.

## **Empirical Study**

We describe here a secondary data analysis of a pooled dataset from two studies that were analyzed and reported separately in a master's thesis (Crowe, 2020)<sup>1</sup>. Their differences consisted of minor interface improvements and gamification features to improve task engagement and realism. The differences between the two studies were not consequential to the main results of the two studies, which justifies pooling their data into a common dataset. The analysis reported here and the cognitive modeling efforts go significantly beyond the analyses presented in the master's thesis.

## Method

We set out to design a study that would be well anchored in the peer-assisted learning theory, achieve good experimental control of potential confounders, and be realistic enough to generalize beyond lab settings. Given that peer-assisted learning is an umbrella concept that applies to a variety of approaches (Olaussen, Reddy, Irvine, & Williams, 2016), it is important to acknowledge here that we restricted this research to a scope that could be realistically managed within a lab study: four learners, a simple associative learning task, and a restricted protocol of interaction among learners. A novel game paradigm, the PAL game, was developed and used to administer stimuli, support interaction among learners, and collect responses.

The PAL Game PAL stands for both peer-assisted learning and paired-associate learning. The PAL game added a simple form of interaction among learners to the classical learning task paired-associate (Anderson, 1981). Participants studied 60 arbitrary word<sup>2</sup>-number pairs (a.k.a., paired associates) and were subsequently tested for accuracy of recall. The game alternated between home-time and school-time sessions. During "home time", participants were given the options to study the word-number associations, play relaxation games (solitaire, chess, or minesweeper), use their phones (e.g., to do web browsing), or do nothing. "School time" consisted exclusively of studying the paired-associate learning task. A final session tested retention of all 60 word-number pairs presented over the course of the study. Participants performed the PAL game in groups of four, with each participant represented on the screen as a labeled rectangular box. Participants were physically separated in individual booths. Each member of the group viewed the first part of the word-number pair (i.e., the word) and was prompted to enter the second part (i.e., the number). After all 4 members gave individual answers to the same stimulus, they were given the opportunity to view any of their peers' answers by moving their mouse over their peers' answer boxes. To collect data on viewing behavior, the PAL software displayed participant answers and recorded viewing time only when another participant hovered over their answer box with the mouse. Upon viewing a peer's answer, a participant could decide to take it by clicking on that peer's answer box. This selection counted as a participant's second answer. If a participant did not take any of their peer's answers, their second answer was taken to be the same as their first answer. A group answer was computed as the mode of the players' second answers, with ties resolved by random selection. The game interface presented first, second, and group answers as well as their respective accuracies. The correct answer was shown to the participants at the end of each trial.

Participants and design A sample of 271 (195 female) volunteers (average age = 19, SD = 3) was recruited from the population of undergraduate students in Psychology at a medium size Midwestern university through Sona Systems (https://www.sona-systems.com/) in exchange for course credits. Three between-subjects experimental groups were formed: (1) the peer-assisted learning (PAL) group, (2) the individual active learning (IAL) group, and (3) the individual passive learning (IPL) group. The PAL group was further divided in subgroups of four participants (i.e., peers). The participants were pseudo-randomly allocated to the three experimental groups, according to the following protocol. Four slots were posted for a given time for volunteers to sign up. If all four slots were filled and four participants showed up for the experiment, they were all assigned as a subgroup to the PAL experimental group. If less than four participants showed up for the experiment, they were randomly assigned to either the IAL or the IPL group. The participants who did not complete the experiment (15) were excluded from analysis. Of the 256 participants who were retained, 136 participants (i.e., 34 groups of 4 participants) were assigned to the peer-assisted learning condition, 63 participants were assigned to the individual active learning condition, and 57 participants were assigned to the individual passive learning condition.

**Procedure** After reading and signing the informed consent, each participant was seated in an individual booth in front of a computer and prompted to read the instructions. Participants did not have verbal or visual contact with other participants during the course of the study. The only interaction afforded to participants in the peer-assisted learning condition was computer-mediated interaction during school time (i.e., they were shown information about

<sup>&</sup>lt;sup>1</sup> See data sets, model code, and other supplementary material at https://science-math.wright.edu/lab/astecca-laboratory/software.

<sup>&</sup>lt;sup>2</sup> Four-letter words with low meaningfulness, imagery, and concreteness were selected from Paivio, Yuille, and Madigan (1968).

each other's choices). The reasons for simplifying peer interaction were precision of measurement and experimental control.

The experiment was divided into six sessions, each composed of home time and school time, ending with a final seventh session that tested retention of all 60 stimuli (i.e., word-number pairs, trials) presented over the course of the study. In sessions 1 through 6 there were 20 stimuli administered per session. Sessions 2 to 5 included 10 stimuli from the previous session and 10 new stimuli. Session 6 included 10 stimuli from session 5 and 10 stimuli from session 1. Thus, each word-number pair was presented two times in the school-time learning sessions and one more time in the testing session. The number of additional presentations of the stimuli in home-time learning sessions was a function of how much time each participant decided to allocate to studying in home time.

Participants were allowed short breaks between sessions. State and trait trust scales, described in the next section, were administered as follows: the trait trust scale was administered before session 1 and after session 7, and the state trust scale was administered after sessions 2, 4, and 6.

In the PAL condition participants performed the PAL game in groups of four. At the start of each trial, the four participants in a group were presented with the same target word and given 5 seconds to respond with the corresponding number. Then each member of the group was given the opportunity to selectively view any of their peers' answers by moving their mouse over their peers' answer boxes. Next, participants gave a second answer, either retaining their initial answer or choosing (with a mouse click) an initial answer given by one of their peers. Finally, all participants received feedback (i.e., correct or incorrect) about their second answer. The PAL software also provided participants with data on who their peers. Each trial lasted approximately 15 seconds.

In the individual active learning (IAL) and individual passive learning (IPL) conditions, participants performed the pair associate learning task individually, without the aid of peers. In the IAL condition, participants were presented with a target word, given a period of time to respond, and then received feedback on their response (correct or incorrect). In the IPL condition, participants were presented with the target word followed directly by the correct paired number, without being given the option to respond.

All conditions experienced the same duration of school time, approximately 5 minutes per session, and the same duration of home time, approximately 3 minutes per session.

**Measures** The following measures were recorded and calculated: the accuracy of the participant's first answer before seeing peers' answers, the accuracy of the participant's second answer, which could be chosen from peers' first answers, the accuracy of test (session 7) answers,

a 24-item measure of a participant's trait trust<sup>3</sup> (i.e., general willingness to trust others), a 14-item measure of state trust<sup>4</sup>, the peer's answer inspection and selection behavior, and the amount of time participants studied during home time.

**Hypotheses** We expect that learners in the PAL group will be able to identify the correct answer among their peers' first answers and take it as their second answer. During the learning sessions (1 through 6), this ability will be reflected in *hypothesis H1* stating that second answer accuracy will be higher than first answer accuracy. This may happen because the learners will learn from feedback not only the correct answers but also who can be trusted among their peers to give correct answers. *Hypothesis H2* states that there is a significant positive correlation between self reported trust in a peer and the peer's first answer accuracy.

Next, we hypothesize that identifying the correct answer through peer interaction may lead to consolidation of the learners' knowledge that will last beyond the learning sessions and should be detectable in the test session. Thus, *hypothesis H3* states that the PAL experimental group will perform better at test (session 7) than IAL and IPL groups.

As reviewed in the background section, there are reasons to expect that peer interaction may have negative effects on learning. Along these lines, peer interaction may trigger a social loafing effect, that is, learners may become less willing to exert learning efforts when they are placed in a peer-interaction condition. *Hypothesis H4* states that home study time will be lower in the PAL condition as compared to the other two conditions. Furthermore, learning in the PAL condition may suffer from interference from their peers' incorrect knowledge (*Hypothesis H5*) or increased cognitive load (*Hypothesis H6*).

## **Results and Discussion of the Empirical Study**

Figure 1 below shows that second answer accuracy is much higher than first answer accuracy, supporting H1. However, H3 was not supported, as test accuracy in the PAL condition was not higher than in the IPL condition and it was actually lower than in the IAL condition. H1 suggests that a knowledge spillover effect occurred in the learning sessions. To test this peer effect more directly in the PAL condition. we computed the correlation between learner accuracy in session n and maximum peer accuracy in session n-1 and found that a 1-unit increase in peer accuracy causes a quarter-unit (0.25) increase in learner accuracy (Y = 0.34 +0.25\*X, Adj. $R^2 = 0.04$ , p < 0.001). Thus, interacting with a knowledgeable peer in the previous session causes improved accuracy in the current session, and vice versa. Even though the effect size is small (r = 0.20), this indicates a significant knowledge spillover effect.

<sup>&</sup>lt;sup>3</sup> This scale included a selection of items from Rotter (1967), Yamagishi (1986), and Collins, Juvina, and Gluck (2016).

<sup>&</sup>lt;sup>4</sup> This scale measured the trust in peers (in the PAL condition) and in the computer's feedback.



Figure 1: Accuracy by condition and session in school time. The dark solid line is the PAL condition, the red dashed line is the IAL condition, and the green dot is IPL condition. The PAL data are broken down into first answer accuracy, second answer accuracy, and group answer accuracy. The group answer was computed as the mode of

individual second answers.

To further understand the knowledge spillover effect, we look at whether learners have any control over their peers' influences in the PAL condition. Figure 2 shows the frequency of taking a peer's answer, the accuracy of that answer, and the accuracy of the learner's own answer. Taking a peer's answer (black solid line) occurs quite frequently (about 50% of the time), even though it slightly decreases with learning across sessions. Taking a peer's answer generally occurs when learner accuracy is low (red dashed line), though increasing. In general, learners become increasingly able to recognize accurate responses in their peers or/and trust them to give accurate responses and take them (green dotted line). We take this as additional evidence in favor of the hypothesis (H2) that learners in the PAL condition learn whom they can trust among their peers to give correct answers. However, sometimes learners take inaccurate answers from their peers, as indicated by the accuracy of the taken answer starting low in session 1 (about 40%) and not reaching the ceiling by session 6 (about 80%). Thus, learners were exposed to both correct answers and errors in their peers, which might explain why the knowledge spillover effect did not transfer to the test session, contrary to H3. The results of testing hypotheses H4 through H6 may shed light on why H3 was not supported.

Next, we turn our attention to how long the participants studied at home in each condition, which addresses H4. Figure 3 shows that participants in the PAL condition did not study less at home. Thus, social loafing cannot explain their relatively poor performance at test. In fact, they studied significantly MORE than the other conditions. Home time practice was correlated with test performance and the magnitude of that correlation was higher in the PAL condition, r(134) = 0.68. Thus, the tests for H1 and H4 are consistent with a composite positive peer effect acting via two channels: knowledge (H1) and motivation (H4).

Taking a peer's answer: frequency and accuracy



Figure 2: Frequency and accuracy of taking a peer's answer in the PAL condition by session in school time.



Figure 3: Time spent studying at home per condition.

A possible reason for the finding that the positive peer effect did not lead to better test performance is exposure to peer errors (H5). We have seen in Figure 2 above that exposure to error did occur, even though with less frequency as learning progressed across sessions. Further support for this hypothesis comes from peer inspection data. We used a mouse tracking procedure to record which of their peers' responses learners looked at. We found that, in the PAL condition, learners were exposed to roughly as many incorrect responses as correct ones. Even though learners became better at selecting the correct answers during the learning sessions, the incorrect answers might have persisted in memory and interfered with the retrieval of correct responses at test. Thus, the positive peer effect might have been offset by a negative interference effect. In the computational modeling section below, we will investigate in more depth how exposure to errors can cause a negative peer effect that can lead to relatively lower performance at test as compared to what would otherwise be expected based on the positive peer effect of knowledge spillover and increased motivation to study.

Lastly, the presence of peers can increase learners' cognitive load (H6), which can further impair learning. We cannot test this hypothesis directly, as we did not administer a measure of cognitive load. However, suggestive evidence in favor of H6 was found by analyzing the number of nonanswers (NAs) during the learning sessions (1 through 6) in the PAL and IAL conditions (recall that participants did not answer in the IPL condition, they only passively observed the stimuli). The number of NAs varied widely between the two conditions: ~5% in the PAL condition and 0.3% in the IAL condition. One possible explanation for this discrepancy is that the cognitive load was much higher in the PAL condition than in IAL condition. The number of NAs predicted poor test performance, r(197) = -0.35, p < -0.350.001, suggesting that higher cognitive load explains part of the poorer performance in the PAL condition. When number of NAs was included as a covariate, the difference between the two conditions at test (session 7) became nonsignificant.

## **Computational Cognitive Modeling**

We are now turning to using post-hoc computational cognitive modeling to explore mechanisms that might explain some of the empirical findings presented above. We focus here on modeling cognitive processes and behavior of the participants in the PAL condition.

## **Model Description**

The model was developed in the ACT-R cognitive architecture (Anderson, 2007). A basic ACT-R model that performs the paired-associates task is available in the ACT-R tutorial<sup>5</sup>. This model performs the task well and fits the human data from a study using 20 paired associates in 8 trials (Anderson, 1981). We extended this model to perform the peer-assisted paired-associates task that human participants performed in the PAL condition of the empirical study presented above<sup>6</sup>.

Just as human participants learned in groups of four, four instances of the model were created that were able to perform the task individually and interact with each other: the models first gave their own answer then chose either their own first answer or a peer's answer as their second answer. To give a first answer (i.e., a number associated with a presented word), the model first tries to retrieve an associate (i.e., word-number pair) from memory. If retrieval fails, the model randomly picks an integer between 0 and 9. When retrieval succeeds, the model takes the number from the retrieved associate and gives it as its first answer. When the model receives feedback, it updates its associate with the correct answer (if necessary) and stores it in memory. As the model encounters repetitions of associates (in home time and school time) the activation of the correct associates increases. This mechanism accounts for the observed increase of first answer accuracy across sessions.

To model individual differences in memory between the four instances of the model, we varied the activation decay, activation noise, and retrieval threshold parameters of the ACT-R architecture, assumed to reflect variability in memory encoding, retention, and retrieval between individuals. Therefore, each instance of the model had a different level of first answer accuracy.

After giving a first answer, the model "views" all first answers (including its own) and chooses one as its second answer. This choice is guided by ACT-R's utility learning mechanism. The model has a rule for each of the four peers that looks at the first answer of that peer and takes it as its own second answer. The four rules compete with each other and the one with the highest utility is selected. When the model is given feedback, if its second answer was correct, a positive reward value is assigned to the selected rule; if its second answer was incorrect, a negative reward value is assigned to the selected rule. This mechanism explains how the model gradually learns which peer is more like to respond accurately and picks their answer, which results in the observed effect of second answer accuracy being higher than first answer accuracy.

However, utility learning is slow and noisy (as governed by the ACT-R parameters learning rate and utility noise), which may lead to selection of incorrect answers. A model's second answer is saved in memory even if it is incorrect, affecting its future first answer (including test) accuracy. This mechanism accounts for the hypothesized mixture of positive and negative peer effects and the observed accuracy of human participants at test in the PAL condition.

## **Model Simulation Results and Discussion**

The model was run for 100 repetitions. Figure 4 shows the model fit to the human data. For first answer accuracy, the correlation was 0.978 with a mean deviation of 0.024. Just as with the human data, the model's second answer accuracy was higher than first answer accuracy, though the fit was not as good. The correlation was 0.683 with a mean deviation of 0.135.

<sup>&</sup>lt;sup>5</sup> Available at http://act-r.psy.cmu.edu/software/

<sup>&</sup>lt;sup>6</sup> The model code can be downloaded from https://sciencemath.wright.edu/lab/astecca-laboratory/software



Figure 4: Comparison of model data (red lines) to human data (black lines) for first answer accuracy (solid lines) and second answer accuracy (dashed lines).

The difference between first and second answer accuracy varies between the four instances of the model or players (not shown here). As expected, player 1, who has the lowest decay rate, activation noise, and retrieval threshold, does not usually benefit from taking another player's answer, whereas players 2, 3, and 4 benefit progressively more.

Overall, the model accuracy at test was facilitated by repetition, exposure to correct responses from peers, and feedback, while being hindered by forgetting (i.e., activation decay in ACT-R) and exposure to incorrect peer responses.

#### **General Discussion**

To summarize, we found positive peer effects acting through both the knowledge channel (i.e., knowledge spillover among peers) and the motivation channel (i.e., increased willingness to practice in the PAL condition). These positive peer effects were offset by negative peer effects acting through the knowledge channel (i.e., exposure to incorrect responses from peers) and the attentional/cognitive channel (i.e., increased cognitive load in the PAL condition).

An ACT-R model using basic architectural mechanisms like base-level learning and utility learning accounted for some of the observed effects. Further modeling work is needed to account for the observed motivational and cognitive load effects of interaction among learners.

#### References

- Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University Press.
- Anderson, J.R. (1981). Interference: The relationship between response latency and response accuracy. *Journal*

of Experimental Psychology: Human Learning and Memory, 7, 326-343.

- Bishop, J.H. (2003). An Economic Theory of Nerd and Slacker Harassment and it's Role in Enforcing Social Norms in Schools?" Center for Advanced Human Resources Discussion Paper 03-06, Cornell University.
- Bishop, J. (2006). Drinking from the fountain of knowledge: Student incentive to study and learn–externalities, information problems and peer pressure. *Handbook of the Economics of Education*, 2, 909-944.
- Collins, M.G. & Juvina, I. (2021). Trust miscalibration is sometimes necessary: An empirical study and a computational model. *Front. Psychol.* 12:690089. doi: 10.3389/fpsyg.2021.690089.
- Collins, M.G., Juvina, I., & Gluck, K. (2016). Cognitive model of trust dynamics predicts outcomes within and between two games of strategic interaction. *Frontiers in Psychology, section Cognitive Science*, 7.
- Crowe, P. (2020). Examining the Role of Trust in Peer-Assisted Learning [Master's thesis, Wright State University]. OhioLINK Electronic Theses and Dissertations Center. http://rave.ohiolink.edu/etdc/view?acc\_num=wright15900 67466554322.
- Geen, R. G., & Gange, J. J. (1977). Drive theory of social facilitation: Twelve years of theory and research. *Psychological Bulletin*, *84*(6), 1267–1288.
- Guerin, B., & Innes, J. M. (1984). Explanations of social facilitation: A review. *Current Psychological Research & Reviews*, 3(2), 32–52.
- Horton, J.J. & Zeckhauser, R.J. (2016). The Causes of Peer Effects in Production: Evidence from a Series of Field Experiments. NBER Working Paper No. 22386.
- Hough, A. R., O'Neill, K., & Juvina, I. (2021). Counterfactual-based nudging and signaling promote more efficient coordination during group tasks. *Comprehensive Results in Social Psychology*, 1-27.
- Hoxby, C. (2000). Peer Effects in the Classroom: Learning from Gender and Race Variation. *National Bureau of Economic Research, Working Paper* 7867, 1-62.
- Karau, S. J., & Williams, K. D. (1993). Social loafing: A meta-analytic review and theoretical integration. Journal of Personality and Social Psychology, 65, 681-706.
- Karau, S.J. & Williams, K.D. (1995). Social loafing: Research findings, implications, and future directions. *Current Directions in Psychological Science* 4(5): 134-140.
- Koles, P. G., Stolfi, A., Borges, N. J., Nelson, S., & Parmelee, D. X. (2010). The impact of team-based learning on medical students' academic performance. *Academic Medicine*, 85(11), 1739-1745.
- Olaussen, A., Reddy, P., Irvine, S., & Williams, B. (2016). Peer-assisted learning: time for nomenclature clarification. *Medical education online*, *21*, 30974.
- Paivio, A., Yuille, J. C., & Madigan, S. A. (1968). Concreteness, imagery, and meaningfulness values for

925 nouns. Journal of experimental psychology, 76(12), 1-25.

- Phelps, C.C., Yang, H., & Steensma, K. (2010). Learning from what others have learned from you: The effects of knowledge spillovers on originating firms. *Academy of Management Journal*, Academy of Management, 2010, 53 (2), pp.371-389.
- Rendell, L., Fogarty, L., Hoppitt, W. J., Morgan, T. J., Webster, M. M., & Laland, K. N. (2011). Cognitive culture: theoretical and empirical insights into social learning strategies. *Trends in cognitive sciences*, 15(2), 68-76.
- Rotter, J. B. (1967). A new scale for the measurement of interpersonal trust. *Journal of personality*, 35(4), 651-665.
- Seddigh, A., Berntson, E., Bodin Danielson, C., & Westerlund, H. (2014). Concentration requirements modify the effect of office type on indicators of health and performance. *Journal of Environmental Psychology*, *38*:167-174.
- Smith, L. C., & Fowler, S. A. (1984). Positive peer pressure: The effects of peer monitoring on children's disruptive behavior. *Journal of Applied Behavior Analysis*, 17(2), 213–227.
- Ulrich, D. L., Brewer, T., Steele-Johnson, D., Juvina, I., Peyton, E., & Hammond, C. (2017). Team-based learning's effects on standardized test scores and student reactions. *Journal on Excellence in College Teaching*, 28(2), 133-165.
- Yamagishi, T. (1986). The provision of a sanctioning system as a public good. *Journal of Personality and social Psychology*, 51(1), 110.
- Zajonc, R.B. & Sales, S.M. (1965). Social facilitation of dominant and subordinate responses. *Journal of Experimental Social Psychology 1224*(34):160–168.

## Understanding Adversarial Decisions for Different Probing-Action Costs in a Deception Game via Cognitive Modeling

Harsh Katakwar (katakwarharsh@gmail.com)

Applied Cognitive Science Laboratory, Indian Institute of Technology Mandi Mandi, India

## Palvi Aggarwal (paggarwal@utep.edu)

Department of Computer Science, The University of Texas at El Paso El Paso, TX, USA

## Varun Dutt (varun@iitmandi.ac.in)

Applied Cognitive Science Laboratory, Indian Institute of Technology Mandi

Mandi, India

#### Abstract

In the cyber world, deception through honeypots has been prominent in response to modern cyberattacks. Prior cybersecurity research has investigated the effect of probing action costs on adversarial decisions in a deception game. However, little is known about the cognitive mechanisms that affect the influence of probing action costs on adversarial decisions. The main objective of this research is to see how an instance-based learning (IBL) model incorporating recency, frequency, and cognitive noise could predict adversarial decisions with different probing action costs. The experimental study had three different probing action costs in the deception game: increasing cost probe (N = 40), no-cost probe (N = 40), and constant cost probe (N = 40). Across the three conditions, the cost for probing the honeypot webserver was varied; however, the cost for probing the regular webserver was kept the same. The results revealed that the cost of probing had no effect on probe and attack actions and that there was a significant interaction between different cost conditions and regular webserver probe actions over the trials. The human decisions obtained in the above experiment were used to calibrate an IBL model. As a baseline, an IBL model with ACT-R default parameters was built. In comparison to the IBL model with ACT-R default parameters, the results showed that the IBL model with calibrated parameters explained adversary decisions more precisely. Results from the model showed higher cognitive noise for cost-associated conditions compared to that of no-cost condition. We highlight the main implications of this research for the community.

**Keywords:** deception, adversary, honeypots, attacker, Instance-based Learning Theory (IBLT), cognitive modeling, probing cost.

## Introduction

Cyberattacks are deliberate attempts by the adversary to intrude into computer systems. Among the various cyberattacks, ransomware attacks increased by 105% in 2021 (Taylor, 2022). Furthermore, attackers have employed phishing as the most common method of luring the public by making lucrative false promises (Taylor, 2022). This rapid increase in attacks drives the scientific community to find adaptable solutions for building secure cyberspace.

Some security solutions, including intrusion detection systems (IDSs), filtering strategies, firewalls, etc. are

available to assist in deterring cyberattacks (Aggarwal & Dutt, 2020; Aggarwal et al., 2022; Rowe & Custy, 2007; Scarfone & Mell, 2007; Shang, 2018). When an IDS detects any unusual behaviour, it shoots off a warning (Aggarwal & Dutt, 2020; Scarfone & Mell, 2007). IDSs are robust; however, they can also incur financial losses by generating false warnings (Shang, 2018). Filtering solutions assist in the removal of undesired content while maintaining secure access. This method could lead to bounded non-rational network agents coming to a consensus (Shang, 2018). In general, such an agreement could aid in the detection of intrusions before they become a cybersecurity risk (Shang, 2018). Overall, these available solutions may not be able to assist in combating emerging cyberattacks.

Cyber deception has been a successful method of thwarting cyber-attacks (Rowe & Custy, 2007). In fact, it has been able to reduce the overall cost of data breaches by 30% (BusinessWire, 2021). The main aim of cyber deception is to take human aspects into account in cyber situations while also improving security tools to reduce cyber-attacks (Rowe & Custy, 2007). Cyber deception has been employed via honeypots, which pretend to be real webservers (Almeshekah & Spafford, 2016). This method has been found to be beneficial in monitoring and mitigating cyberattacks. Deception in cybersecurity has been explored using mathematical and canonical games (Carroll & Grosu, 2009; Garg & Grosu, 2007; Kiekintveld et al., 2015). Kiekintveld et al. (2015) examined how a game-theoretical technique could be applied to manipulate information in adversarial environments. Similarly, Garg and Grosu (2007) proposed a mathematical framework for a security game involving deception. Carroll and Grosu (2009) described the interaction between an adversary and a defender as a signalling game.

Recent behavioral cybersecurity research has focused more on technological aspects that influence adversarial decisions in cybersecurity. Some of them include network topology, timing and amount of deception, network size, honeypot proportions, probing action costs, the complexity of cyberattacks, etc. (Aggarwal et al., 2017; Katakwar et al., 2020). Aggarwal et al. (2017) evaluated the impact of timing and amount of deception on adversarial decisions and revealed that late deception increased the proportion of honeypot attacks when compared to early deception. Similarly, Katakwar et al. (2020) investigated the effect of various network sizes on adversarial decisions in cyberspace. In addition, these researchers have also built computational cognitive models that helped them understand the various cognitive elements that play a vital role in influencing adversarial decisions in cyber scenarios (Katakwar et al., in press).

Recently, Katakwar et al. (2022) have evaluated the effect of probing action costs on adversarial decisions in a deception-based security game experimentally. They found that cost of probing had no effect on probe and attack actions and that there was a significant interaction between different cost conditions and regular webserver probe actions over the trials. However, they did not look into different cognitive parameters that drive the adversarial decisions in complex cyber circumstances. Building cognitive models based on Instance-based Learning Theory (IBLT) is one approach to comprehending cognitive factors in dynamic situations (Dutt & Gonzalez, 2012; Gonzalez et al., 2003; Gonzalez & Dutt, 2011). Previously, IBLT-based cognitive models were able to explain how adversaries made decisions in different cyber scenarios (Aggarwal et al., 2017; Dutt et al., 2013). Hence, in this research, we address the research gap by building cognitive models based on IBLT that could account for adversarial decisions in cyber situations with different probing costs.

In what follows, we first briefly discuss the working of the Deception Game (DG). Next, we describe the findings of Katakwar et al. (2022). Thereafter, we detail the background of IBLT, and thereafter we present the results and conclusions of the developed cognitive models.

## **Deception Game**

DG is a sequential, single-player, incomplete information game in which an adversary and a network compete against each other (Aggarwal et al., 2016a, 2016b; Garg & Grosu, 2007). The game is formally defined as DG (n, k,  $\gamma$ ), where *n* denotes the total number of webservers, *k* denotes the number of honeypots, and  $\gamma$  denotes the number of probes after which the adversary makes his final decision to attack the network. The DG has two types of webservers: regular and honeypot. Regular webservers are the real webservers that contain valuable information, whereas honeypots are fake servers that pretend to be real in order to trap opponents and extract meaningful information.

The game is played over multiple rounds. There are two phases in each round of the game: probe stage and attack stage. An adversary could probe webservers several times during the probe stage. Probing implies clicking on the button in the game's UI that represents a webserver. For each probe, the adversary receives a response from the system indicating whether the system is a regular (real) webserver or a honeypot (fake) webserver. Depending on whether or not the deception is present, this feedback may or may not be correct. As a result, the adversary may not be able to learn across multiple rounds in this scenario. Furthermore, the game dynamics may closely resemble those in the real world, in which adversaries may have limited knowledge of the infrastructure they are attempting to attack. Overall, the goal of deception is to deceive the opponent into believing misleading information about the state of the servers. If deception is present in a round, the network response is the total opposite of the webservers' actual state. If there is no deception in a round, the network's response will be identical to the true state of webservers. The adversary also has the option of not probing any webservers during the probe stage. Deception and unreliability in the feedback of the probe stage may increase not-attack activities, as the adversary will likely avoid regular/honeypot attack actions due to the probe stage's response.

We had three different variants of DG in this experiment: increasing-cost, no-cost, and constant-cost. In the increasingcost condition, the cost of probing the honeypot webserver grew linearly as the round progressed. If the adversary probed the honeypot webserver for the  $i^{th}$  time in a given round, the adversary received -5\*i points. In the no-cost condition, there were no penalties for probing the honeypot webservers across all rounds of the DG. In the constant-cost condition, the cost of probing the honeypot webservers was kept constant over the rounds. As a result, the attacker received -5 points for each probe of the honeypot webserver. Across all the conditions, there were constant cost to probe the regular webserver in DG.

## Experiment

## **Experiment Design**

Katakwar et al. (2022) randomly allocated participants to one of three between-subjects conditions: no-cost probe (40 participants), constant-cost probe (40 participants), and increasing-cost probe (40 participants). There were four webservers in the network under all conditions, two of which were regular webservers and the other two were honeypots. In addition, there were 29 trials, 14 of which were nondeception rounds, and the rest were deception rounds. The participants were informed about the presence of deception in a DG, but they did not know which round belonged to the deception/non-deception condition. Also, the deception and non-deception rounds in DG did not form a particular sequence or pattern that participants could predict. Across the conditions, the adversary probes multiple times before moving to the attack stage, where he/she makes the decision to attack one of these webservers present in the network. For all the conditions, there were six dependent variables, three for the probe decisions and three for the attack decisions. In addition, we grouped the 29 trials into blocks of 5 trials each to see the effect of varied cost conditions on probe and attack decisions over the trials. As a result, the 29 trials were divided into 6 blocks, with the first block including 5 trials and the last block containing 4 trials. After that, for each block, the proportions of regular webserver probe/attack, honeypot webserver probe/attack, and no webserver probe/attack were determined.

#### **Participants**

Katakwar et al. (2022) recruited 120 participants anonymously recruited from the crowd-sourcing platform called Amazon Mechanical Turk (Mason & Suri, 2012). Sixty-six percent of participants were male, whereas the remaining thirty-four percent were female. More than ninetyfour percent of the participants had a college degree. Seventyfour percent of the participants were from the fields of Science, Technology, Engineering, and Management (STEM) background. Once the study was over, participants were thanked and compensated INR 50 (USD 0.72) for their participation in the study. In addition, the top-three scorers were randomly chosen for the lucky draw contest, with one of them winning a gift card.

#### Procedure

Participants in the study were provided information about their roles and goals in the DG. Participants were also given information about their tasks and the associated payoffs. Over the course of numerous rounds of DG, participants were asked to maximize their payoff. The presence of deception and non-deception rounds in DG was communicated to participants by text instructions, but they were unaware of which rounds involved deception or non-deception. In addition, the configuration of regular and honeypot webservers was randomized in each round so that the percentage of regular and honeypot webservers remained consistent with the conditions. There were two phases to each round of DG: probe and attack. During the probe phase, the adversary may or may not probe a few webservers present in the network. Similarly, during the attack phase, the adversary had the option of attacking one of the webservers or none of them. Participants were thanked and compensated for their participation once the study was completed.

#### Results

# Influence of different probe costs on adversarial decisions during probe and attack stages

Katakwar et al. (2022) investigated the impact of the different probing action costs on adversarial decisions during the probe stage. They found that proportion of different probe decisions were insignificant across different cost conditions. The proportion of regular webserver probe decisions in the increasing-cost condition, no-cost condition, and constant-cost condition were 0.44, 0.47, and 0.45, respectively (*F* (2, 117) = 0.919, p = .402,  $\eta^2 = 0.015$ ). Similarly, the proportion of honeypot webserver probe decisions in increasing-cost condition, no-cost condition, and constant-cost condition, no-cost condition were 0.43, 0.47, and 0.43, respectively (*F* (2, 117) = 1.454, p = .238,  $\eta^2 = 0.020$ ). The proportion of no webserver probe decisions in increasing-cost condition, no-cost condition, and constant-cost condition, and 0.43, respectively (*F* (2, 117) = 1.454, p = .238,  $\eta^2 = 0.020$ ). The proportion of no webserver probe decisions in increasing-cost condition, no-cost condition, and constant-cost condition, and constant-cost condition webserver probe decisions in increasing-cost condition.

constant-cost condition were 0.13, 0.06, and 0.12, respectively (*F* (2, 117) = 1.359, p = .261,  $\eta^2 = 0.024$ ).

Similarly, they also investigated the effect of the different probing action costs on adversarial decisions during the attack stage. The proportion of different attack decisions were insignificant across different cost conditions. The proportion of regular webserver attack decisions in increasing-cost condition, no-cost condition, and constant-cost condition were 0.42, 0.45, and 0.42, respectively ( $F(2, 117) = 0.606, p = .547, \eta^2 = 0.010$ ). The proportion of honeypot webserver attack decisions in increasing-cost condition, and constant-cost condition, no-cost condition were 0.40, 0.44, and 0.43, respectively ( $F(2, 117) = 1.454, p = .238, \eta^2 = 0.024$ ). The proportion of no webserver attack decisions in increasing-cost condition, and constant-cost condition were 0.18, 0.11, and 0.14, respectively ( $F(2, 117) = 1.359, p = .261, \eta^2 = 0.023$ ).

## Influence of different cost conditions over the trials on adversarial decisions during probe stage

Katakwar et al. (2022) investigated the effect of probe decisions over the trials as a within-subject factor and different probing cost conditions as a between-subject factor. Figure 1 shows the proportion of regular probes over blocks of trials in different cost conditions. As shown in Figure 1, they also found that there was a significant interaction between different cost conditions and blocks (F (10, 585) = 2.052, p < .05,  $\eta^2 = 0.034$ ). Also, averaged over all conditions, the proportions of regular probe decisions over the blocks were significant and decreasing (F (5, 585) = 2.529, p < .05,  $\eta^2 = 0.021$ ).



Figure 1. Proportion of regular webserver probes over the blocks of trials across different cost conditions.

However, the proportion of honeypot webserver probes were not significant over blocks ( $F(5, 585) = 1.662, p = .142, \eta^2 =$ 0.014). Also, the interaction between honeypot webserver probes and different cost conditions was not significant (F(10, 585) = 1.667,  $p = .085, \eta^2 = 0.028$ ). Similarly, the proportion of no webserver probe decisions were not significant over blocks ( $F(5, 585) = 1.348, p = .243, \eta^2 =$ 0.011). Also, the interaction between different cost conditions and the no webserver probe decisions were found to be insignificant ( $F(10, 585) = 1.171, p = .307, \eta^2 = 0.020$ ).

## Influence of different cost conditions over the trials on adversarial decisions during attack stage

Katakwar et al. (2022) investigated the effect of different cost conditions over the trials on adversarial decisions in the attack stage. They found that there was not any significant interaction between different cost conditions and the following proportions of attack decisions over blocks: regular webserver attack ( $F(10, 585) = 0.579, p = .832, \eta^2 = 0.010$ ), honeypot webserver attack ( $F(10, 585) = 0.664, p = .758, \eta^2 = 0.011$ ) and no webserver attack ( $F(10, 585) = 0.664, p = .758, \eta^2 = 0.011$ ) and no webserver attack ( $F(10, 585) = 1.422, p = .166, \eta^2 = 0.024$ ). Also, the proportion of decisions over blocks was not significant for these decisions: regular webserver attack ( $F(5, 585) = 0.111, p = .990, \eta^2 = 0.001$ ), honeypot webserver attack ( $F(5, 585) = 0.936, p = .457, \eta^2 = 0.008$ ), and no webserver attack ( $F(5, 585) = 1.854, p = .100, \eta^2 = 0.016$ ).

#### **IBL Model**

IBLT is a decision-making theory for complicated circumstances based on experience (Dutt & Gonzalez, 2012; Gonzalez et al., 2003; Gonzalez & Dutt, 2011). Prior research in computational modeling using cognitive theories such as IBLT has shown to be effective in forecasting human behaviour in complex situations. The instances are built in the memory for each occurrence of an outcome on choice options in an IBL model. In the model, an instance has the triplet frame situation-decision-utility. The circumstance in the instance represents the current situation, the decision represents the decision made in the current situation (option of one of the alternatives), and utility represents the outcome achieved from the decision made in the current situation. When a decision must be made, the instances of each alternative are retrieved from memory. These occurrences are then blended together for each choice. The activation of occurrences, as well as their likelihood of being recalled from memory, are used thereafter for calculating the blended value of an option.

$$V_{j,t} = \sum_{i=1}^{n} p_{i,j,t} x_{i,j,t}$$

where  $p_{i,j,t}$  is the likelihood of recalling an instance *i* for an option *j* in the *t*<sup>th</sup> trial of the experiment, and  $x_{i,j,t}$  is the utility value of an instance *i* for an option *j* in the trial *t*. In each trial, the model chooses the option with the highest blended value. The blended value for each option is generated using the above equation, which is the summation of all observed outcomes weighted by the retrieval probability. The retrieval probability of the instances is described as follows:

$$p_{i,j} = \frac{e^{\frac{A_{i,j,t}}{\tau}}}{\sum_{i=1}^{n} e^{\frac{A_{i,j,t}}{\tau}}}$$

where  $A_{i,j,t}$  is the activation value of an instance *i* corresponding to the memory choice *j*;  $\tau$  is the random noise parameter, which is specified as  $\tau = \sigma * 2$ ; and  $\sigma$  is the free cognitive noise parameter to represent the uncertainty of recalling prior experiences from the memory. In a given trial, the activation value of an instance is determined by the

frequency with which its outcome happens and the time difference between the current time and the previous time when the instance's outcome occurred in the task. The activation value of a given instance i is defined for each trial t as follows:

$$A_{i} = \ln \left( \sum_{t_{p,i} \in \{1, \dots, t-1\}} (t - t_{p,i})^{-d} \right) + \sigma * \ln(\frac{1 - \gamma_{i,t}}{\gamma_{i,t}})$$

where, *d* and  $\sigma$  are the hyperparameters known as memory decay and cognitive noise respectively; *t* is the current trial;  $t_{p,i}$  are the prior trials in which outcome with instance *i* occurred in the task; and  $\gamma_{i,t}$  is the random number chosen from the uniform distribution between 0 and 1. So, the frequency of occurrence of outcomes in the task and the recency of those outcome observations increase the activation of an instance *c* orresponding to the observed outcome. The decay parameter *d* takes into consideration reliance on current information. The greater the reliance on recency and the faster memory decay, the higher the value of the *d* parameter. The  $\sigma$  parameter compensates for variation in instance activation from sample to sample. The greater the value, the higher variability in instance activations and trial-to-trial decisions.

#### **Parameter Calibration**

We built two different variants of the IBL model. The first variant of the IBL model had calibrated parameters of d and  $\sigma$ , which was referred to as IBL-calibrated model. However, the second variant of the model had default ACT-R parameters of d and  $\sigma$  as 0.50 and 0.25 respectively, referred as IBL-ACT-R model. Using experimental data of different cost conditions, we found the optimal values of d and  $\sigma$  for IBL-calibrated model. For both the variants of IBL-based model, 120 model agents were used across different trials. Across the 29 trials, we tried to minimize the average of Mean Squared Deviations (MSD) on the proportion of attack and not-attack decisions made by humans and models.

$$MSD = \frac{1}{29} \sum_{t=1}^{29} (model_t - human_t)$$

where, *t* depicts trial from 1 to 29; *model*<sub>t</sub> and *human*<sub>t</sub> refers to the attack decisions in the trial *t* from model and human participants, respectively. So, if the MSD value is minimal, the model's fit to human data is better. To maximize the values of *d* and  $\sigma$  parameters for both model participants, the Genetic Algorithm (GA), an optimization algorithm, was utilized. In the genetic algorithm, the utility value for the regular webserver, honeypot webserver, and no probe/attack varied from -100 to 100, whereas the *d* and  $\sigma$  parameters varied from 0 to 10.

The IBL-ACT-R model is based upon ACT-R framework, a cognitive theory that has been used to explain a variety of cognitive science findings (Anderson et al., 1997). ACT-R is a cognitive architecture designed to account for the various complex operations of the human mind. In the IBL-ACT-R model, we have d and  $\sigma$  parameters, which were set based on the ACT-R default values of 0.50 and 0.25, respectively. Smaller values of d suggest that information is less reliant on frequency and recency, and smaller values of  $\sigma$  indicate that trial-to-trial decisions are less variable. We compared the performance of IBL-ACT-R and IBL-calibrated models.

## **Model Results**

Table 1 shows the values of model parameters and MSD between human and model for different conditions of both models. The d and  $\sigma$  are the free parameters of the models where d parameter denotes the memory decay and  $\sigma$  denotes the variability in trial-to-trial decisions. In the IBL-calibrated model, d value was smaller for cost-associated conditions i.e., constant cost (d = 1.21) and increasing cost (d = 1.56) and higher for no-cost condition (d = 8.50). Similarly,  $\sigma$  value was higher for the cost-associated conditions i.e., constant cost ( $\sigma$ = 8.89) and increasing cost ( $\sigma$  = 7.67), and lower for no cost ( $\sigma = 0.56$ ). The MSD value for the attack and not attack actions of the IBL-ACT-R model across all the conditions were higher compared to the total MSD value of the calibrated model. Figure 2 shows the proportion of different attack and not attack decisions over the blocks of trials in increasing cost conditions in human data, IBL-calibrated model, and IBL-ACT-R model. Figure 3 shows the proportion of different attack and not attack decisions over the blocks of trials in constant cost condition in human data, IBL-calibrated model, and IBL-ACT-R model. Figure 4 shows the proportion of different attack and not attack decisions over the trials in no cost condition in human data, IBL-calibrated model, and IBL-ACT-R model.



Figure 2. Proportion of different attack and not attack decisions over the blocks of trials in increasing cost conditions in human, IBL-calibrated model, and IBL-ACT-R model.



Figure 3. Proportion of different attack and not attack decisions over the blocks of trials in constant cost condition in human data, IBL-calibrated model, and IBL-ACT-R model.

Condition	Model d σ		Utility value for different actions			MSD for attack and not-attack actions		Average	
				Regular webserver	Honeypot webserver	No action	Attack action	No attack action	MSD
Increasing	IBL-Calibrated Model	1.56	8.89	39.66	-23.96	10.95	0.008	0.008	0.008
Cost	IBL-ACT-R Model	0.50	0.25	39.66	-23.96	10.95	0.016	0.016	0.016
No Cost	IBL-Calibrated Model	8.50	0.56	61.62	-49.08	41.28	0.006	0.006	0.006
INO COSE	IBL-ACT-R Model	0.50	0.25	61.62	-49.08	41.28	0.015	0.015	0.015
Constant Cost	IBL-Calibrated Model	1.21	7.67	34.45	-46.15	4.46	0.002	0.002	0.002
	IBL-ACT-R Model	0.50	0.25	34.45	-46.15	4.46	0.003	0.003	0.003

Table 1. Model parameters, MSD across different conditions for the IBL-calibrated model and the IBL-ACT-R model, and utility value of the pre-populated instances of regular webserver, honeypot webserver, and no actions



Figure 4. Proportion of different attack and not attack decisions over the trials in no cost condition in human data, IBL-calibrated model, and IBL-ACT-R model.

## Discussion

Deception using honeypot has been demonstrated to be an important approach for combating modern cyber-attacks (Almeshekah & Spafford, 2016). Researchers in the field of adversarial cybersecurity have created and deployed canonical games to investigate the effectiveness of deception in various cybersecurity scenarios (Aggarwal et al., 2016a; 2016b). In addition, researchers have examined the many human factors that influence the adversary's decision in deception-based security games (Aggarwal et al., 2016a; Katakwar et al., 2020). Recently, Katakwar et al. (2022) has evaluated the effects of probing action costs in a deception-based game. However, they did not try to understand different cognitive factors involved in adversarial decisions in this cyber situation.

The findings of Katakwar et al. (2022) revealed that the varying costs of probing actions had no effect on adversarial decisions made during the attack phase in DG. However, there was a significant effect of regular probe decisions over the blocks of trials in DG. The results also indicated that both constant-cost and increasing-cost conditions, the proportion of regular probing decisions followed a consistent pattern over rounds. Furthermore, the proportion of regular probe decisions decreased across the blocks of trials. According to IBL theory, humans choose the alternatives that maximize their overall values. When there is a cost connected for probing honeypot webserver, the adversary suffers negative consequences. This negative experience reduces the combined value of the honeypot probe/attack decision. In contrast, the attacker suffers no negative consequences when probing/attacking a webserver in the no-cost probe. As a result, we see a significant effect of different cost conditions on regular probe decisions in DG over the trials. Also, there was no influence of different cost conditions on the adversarial decision-making during the attack phase. As the attack phase followed the probe phase and the cost was associated with probing. Thus, the proportion of actions during the attack phase across different cost conditions were similar.

The cognitive models' results revealed that the no-cost condition had a higher memory decay value (d = 8.50) than the cost-associated conditions. As in the no-cost condition, the adversaries had no negative experience, which made them more reliant on the DG's feedback. As a result, the memory decay value for the no cost condition is much higher than that for the cost-associated situations. Furthermore, the model revealed a high cognitive noise value for cost-associated conditions ( $\sigma = 8.89$  for constant cost and  $\sigma = 7.67$  for increasing cost). One explanation for this result is that increasing the cost of probing the honeypot webserver increases the adversary's negative experience. This negative experience along with the presence of deception baffled the adversary, prompting the adversary to probe fewer regular webservers.

We also found pre-populated utility values for regular webserver action, honeypot webserver action, and no webserver action for the various cost conditions via calibration. The pre-populated utility value for regular webserver action and no action for the no-cost condition was quite high in comparison to cost-associated conditions. Furthermore, the pre-populated utility value for honeypot webserver action for cost-associated conditions was negative as compared to the no-cost condition. The reasons behind both outcomes can be understood with the aid of IBLT. In the no-cost condition, the adversary does not receive any negative feedback, making instances of gains more active than instances of losses. As a result, adversaries have a positive opinion about the honeypot webserver. Furthermore, in the no-cost condition, the adversary only received positive rewards for probing/attacking webservers, resulting in a positive opinion about webservers. Thus, the utility values for regular webserver action and no webserver action in the nocost condition were higher than in cost-associated conditions. However, in cost-associated conditions, as the adversaries have some negative experiences, this leads to a negative perception of honeypot webservers among the adversaries.

One drawback of this study is that the results are based on a lab-based study. As a result, some of the findings might not be applicable in the real-world settings. In addition, the adversaries in this investigation were unaware of deception rounds and the actual identities of webservers, which could have influenced their decisions during the probe and attack stages. One practical implication of this research in the realworld is that the cognitive models derived from this research could be used to build decision support system for organizations, which may assist inexperienced defenders and analysts to make decisions in cyber environments. Also, the models can be utilized for performing penetration testing in different cyber settings to determine exploitable vulnerabilities.

In the future, we intend to investigate how various deception and non-deception patterns might be used to deceive the enemy from the genuine target in a cyber environment. Furthermore, because of the complicated cyber environment, it is quite expected that adversaries will exhibit various cognitive biases; hence, we plan to investigate the presence of cognitive biases in cyber settings. These are some ideas that we intend to study in our future research.

#### References

- Aggarwal, P., & Dutt, V. (2020). The role of information about opponent's actions and intrusion-detection alerts on cyber decisions in cyber security games. *Cyber Security: A Peer-Reviewed Journal*, 3(4), 363–378.
- Aggarwal, P., Gonzalez, C., & Dutt, V. (2016a). Cybersecurity: Role of deception in cyber-attack detection. *Advances in Intelligent Systems and Computing*, 501, 85–96. <u>https://doi.org/10.1007/978-3-319-41932-9\_8</u>
- Aggarwal, P., Gonzalez, C., & Dutt, V. (2016b). Looking from the hacker's perspective: Role of deceptive strategies in cyber security. 2016 International Conference on Cyber Situational Awareness, Data Analytics and Assessment, CyberSA 2016. https://doi.org/10.1109/CYBERSA.2016.7503288
- Aggarwal, P., Gonzalez, C., & Dutt, V. (2017). Modeling the effects of amount and timing of deception in simulated network scenarios. 2017 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (CyberSA),1–7,

https://doi.org/10.1109/CyberSA.2017.8073405

- Aggarwal, P., Moisan, F., Gonzalez, C., & Dutt, V. (2022). Learning About the Effects of Alert Uncertainty in Attack and Defend Decisions via Cognitive Modeling. *Human Factors*, 64(2), 343–358. https://doi.org/10.1177/0018720820945425
- Almeshekah, M. H., & Spafford, E. H. (2016). Cyber security deception. *Cyber Deception: Building the Scientific Foundation*, 23–50. <u>https://doi.org/10.1007/978-3-319-32699-3\_2</u>
- Anderson, J. R., Matessa, M., & Lebiere, C. (1997). ACT-R: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction*, *12*(4), 439–462. https://doi.org/10.1207/s15327051hci1204\_5
- Carroll, T. E., & Grosu, D. (2009). A game theoretic investigation of deception in network security. *Proceedings - International Conference on Computer Communications and Networks, ICCCN.* <u>https://doi.org/10.1109/ICCCN.2009.5235344</u>
- BusinessWire (2021). Cyber Deception Reduces Data Breach Costs by Over 51% and SOC Inefficiencies by 32%. https://www.businesswire.com/news/home/2020091400516 5/en/Cyber-Deception-Reduces-Data-Breach-Costs-by-Over-51-and-SOC-Inefficiencies-by-32
- Dutt, V., & Gonzalez, C. (2012). Making Instance-based Learning Theory usable and understandable: The Instance-based Learning Tool. *Computers in Human Behavior*, 28(4), 1227– 1240. <u>https://doi.org/10.1016/j.chb.2012.02.006</u>
- Dutt, V., Ahn, Y. S., & Gonzalez, C. (2013). Cyber situation awareness: Modeling detection of cyber attacks with instance-based learning theory. *Human Factors*, *55*(3), 605– 618. <u>https://doi.org/10.1177/0018720812464045</u>

- Garg, N., & Grosu, D. (2007). Deception in honeynets: A game-theoretic analysis. *Proceedings of the 2007 IEEE Workshop on Information Assurance, IAW*, 107–113. https://doi.org/10.1109/IAW.2007.381921
- Gonzalez, C., & Dutt, V. (2011). Instance-Based Learning: Integrating Sampling and Repeated Decisions From Experience. *Psychological Review*, *118*(4), 523–551. https://doi.org/10.1037/a0024558
- Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science*, 27(4), 591–635. <u>https://doi.org/10.1016/S0364-0213(03)00031-4</u>
- Katakwar, H., Aggarwal, P., Maqbool, Z., Dutt, V. (2020). Influence of network size on adversarial decisions in a deception game involving honeypots. *Frontiers in Psychology*, 11, 2385.
- Katakwar, H., Aggarwal, P., Maqbool, Z., & Dutt, V. (2022). Influence of Probing Action Costs on Adversarial Decision-Making in a Deception Game. In S. Fong, N. Dey, & A. Joshi (Eds.), *ICT Analysis and Applications* (pp. 649–658). Springer Singapore.
- Katakwar, H., Uttrani, S., Aggarwal, P., & Dutt, V. (in press). Modeling the effects of network size in a deception game involving honeypots. In A. A. Moustafa (Ed.), *Cybersecurity and Cognitive Science*. Elsevier.
- Kiekintveld, C., Lisý, V., & Píbil, R. (2015). Game-theoretic foundations for the strategic use of honeypots in network security. *Advances in Information Security*, 56, 81–101. <u>https://doi.org/10.1007/978-3-319-14039-1\_5</u>
- Mason, W., & Suri, S. (2012). Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods*, 44(1), 1–23. https://doi.org/10.3758/S13428-011-0124-6
- Rowe, N. C., & Custy, E. J. (2007). Deception in cyber attacks. *Cyber Warfare and Cyber Terrorism*, 91–96. https://doi.org/10.4018/978-1-59140-991-5.CH012
- Scarfone, K., & Mell, P. (2007). Guide to intrusion detection and prevention systems (idps). *NIST Special Publication*, 800(2007), 94.
- Shang, Y. (2018). Hybrid consensus for averager-copiervoter networks with non-rational agents. *Chaos, Solitons and Fractals, 110,* 244–251.
- Taylor, A. (2022). Ransomware cyberattacks surged in 2021 according to a new report. Fortune. https://fortune.com/2022/02/17/ransomware-attackssurge-2021-report/

# Reverse-Engineering of Boolean Concepts: A Benchmark Analysis

Felix Kettner (felix.kettner@hsw.tu-chemnitz.de) Elisa-Maria Heinrich (elisa-maria.heinrich@hsw.tu-chemnitz.de) Daniel Brand (daniel.brand@cognition.uni-freiburg.de) Marco Ragni (marco.ragni@hsw.tu-chemnitz.de) Department Behavioural and Social Sciences, Technische Universität Chemnitz,

Straße der Nationen 62, 09111 Chemnitz

#### Abstract

For a long time the human capability to form hypotheses from observations has been in the focus of research in psychology and cognitive science. An interesting case is to form hypotheses about the underlying mechanisms of technical systems. This process is called reverse-engineering, i.e., to identify how a system works. Research so far has focused on identifying general principles of the underlying reasoning process and lead to the development of at least three general approaches. This paper investigates the predictive power of existing models for each individual reasoner for the first time, i.e., can the individual reasoner reverse engineer the Boolean Concepts from observations. Towards this goal, we (i) defined a modeling task on the individual level, (ii) adapt or re-implement existing models for Boolean Concept learning to make predictions on the individual level, (iii) identify base-line models and additional strategies, and (iv) evaluate the models. By focusing on the individual level, we uncover limitations of current state of the art and discuss possible solutions.

**Keywords:** Boolean concepts; mental models; algebraic complexity; minimal description; reverse-engineering; benchmark

#### Introduction

Imagine a living room with a single lightsource in the middle of the room, and several doors with lightswitches next to each door. The basic assumption is that every single switch is included into the circuit and therefore has an influence on the condition of the light. Given this, and the fact that every switch can have two different states, i.e., on and off, there are several combinations of these states which will result in a shining lightbulb, and the remaining possible combinations will turn the light off. This concept can be reduced and depicted as shown in Figure 1 by utilising just a representation of the switches and the lightsource. If you had the task to figure out and describe the valid combinations of switches to light the bulb, how would you proceed? Presumably, you would try different combinations and finally come up with an corresponding answer. Such an answer could look like "Switch a has to be turned on and switch b has to be turned off to turn on the light.". By answering in such a way we intuitively tend to use so called "Boolean concepts" to develop an idea of the underlying electric circuit. Boolean refers to the fact that a variable, in the example above the single lightswitches and the lightbulb, can only have two different states: on or off. In logical circuits they are represented by true and false. Furthermore Boolean operators are a way to combine variables or states with other ones in a logical way to describe conditions for a certain target state.

Two basic Boolean operators are *AND* and *OR* which are used to combine variables just like in the given example above: "*Switch a* has to be on *AND switch b* has to be off to turn on the light.". Another operator is *NOT*, which reverses the state of a variable (e.g., the state "off" could also be desribed as "*NOT* on"). Instead of writing *AND*, *OR* and *NOT* in Boolean algebra the symbols  $\land$ ,  $\lor$  and  $\neg$ , respectively, are used. For simplicity, we also refer to the switches only with *a*, *b* and *c*, respectively. In our example this would lead to the expression  $a \land \neg b$  to describe when the light turns on.

Although the basic elements of Boolean concepts are quite simple, the combination of several variables can become very complex and therefore hard to comprehend for humans. The effect of increasing complexity leading to more difficulties for humans to understand such expressions is known as the Shephard trend based on work of Shepard, Hovland, and Jenkins (1961) and confirmed by various other authors (e.g., Smith, Minda, & Washburn, 2004; Love, 2002; Feldman, 2000). Since the inception of this trend a lot of attempts have been made to find a suitable measurement for the complexity of Boolean Concepts to predict human performance in this field accordingly. Some of the most prominent theories are Minimal Descriptions (Feldman, 2000), Algebraic Complexity (Feldman, 2006) and Mental Models by Goodwin and Johnson-Laird (2011).



Figure 1: Example of three switches and a lightbulb

While those previous approaches focused on modeling the statistical aggregate of all participant's responses, the focus of this paper is to identify the reasoning difficulty for each individual, i.e., when does the task become too difficult to solve correctly? Hence, we implemented the mentioned theories with a mechanism to adapt to an individual reasoner, compared their accuracy when accounting for the correctness of individual participants and investigated possible extensions.

## **Boolean concepts**

We briefly introduce some necessary background on Boolean concepts. The first step in understanding Boolean concepts is to grasp Boolean variables. Boolean concepts are build upon variables, which can only have the two distinct states of *true* or *false* respectively, when talking about circuits, *on* and *off*. Based on this we can already depict a simple circuit as shown in Figure 2 where the state of the switch equals the state of the whole system, i.e., when the switch is *on*, the light will be *on*.



Figure 2: Depiction of a simple circuit where the state of the switch equals the state of the light.

But when adding more switches to the system we need operators to describe in which way these switches depend on each other and impact the state of the whole system. Figure 3 gives an example of two possible configurations of a circuit with two switches which now leads to the basic Boolean operations: The conjunction (with the operator AND;  $\land$ ) and the disjunction (with the operator OR;  $\lor$ ).



Figure 3: Depiction of circuits.

Conjunctions are evaluated to be fulfilled, hence true, if all the combined single statements are fulfilled. A disjunction is fulfilled if at least one of the combined statements is fulfilled. Therefore, refering to Figure 3a, the circuit shows the conjunction concept where both switches (i.e.,  $a \wedge b$ ) have to be on in order to turn the light on. The circuit in Figure 3b shows the disjunction concept where it is sufficient that solely one switch is on in order to turn the light on, but still both switches on will also lead to a shining lightbulb. The third basic operation of Boolean concepts is the negation, which serves to reverse the state of a variable or statement (*NOT*;  $\neg$ ).

One peculiarity about Boolean concepts is, that although the basics are quite simple, the combination of several variables can easily get very complex. With three variables already eight combinations are available as shown in Figure 4 with the Boolean concept  $(a \lor c) \land \neg b$  used as an example.



Figure 4: All possible combinations for a Boolean concept with three variables representated as switches. The instances for the concept  $(a \lor c) \land \neg b$  are highlighted with an active lightbulb.

#### Approaches for Estimating Difficulty

In this section we introduce the approaches that we use in our analysis as an estimate for the difficulty of Boolean concepts (operationalized by the correctness when solved by participants). For the scope of this paper, we selected only approaches where either a full implementation was available or the respective difficulty estimates were reported by Goodwin and Johnson-Laird (2011). This ensures that the results are comparable and no discrepancies due to a different understanding of the approaches occur.

## **Minimal Description**

For each Boolean expression exists a minimal description length. For example,  $(a \land b) \lor (a \land \neg b) \lor (\neg a \land b)$  can be shortened to the minimal description  $a \lor b$  which can not be shortened any further. Due to the fact that deriving such minimal descriptions from complex Boolean expressions is not computationally tractable, Feldman (2000) used a set of heuristics to find the corresponding minimal descriptions for the Boolean concepts used in the given dataset. The minimal description value then equals the amount of used variables in the respective minimal description as shown in the examples in Table 1. Based on the Shepard trend (Shepard et al., 1961)

Table 1: Examples for minimal description values

Minimal description	Value
$a \wedge \neg b$	2
$(a \land \neg b) \lor (a \land c)$	4
$\neg (((a \land \neg b) \land c) \lor ((\neg a \land b) \land \neg c)))$	6

and Feldman (2000) this classification of Boolean concepts should be able to predict their difficulty.

## **Algebraic Complexity**

The approach of algebraic complexity by Feldman (2006) is based on a decomposition of Boolean expressions into underlying regularities instead of using the minimal description length. Therefore Boolean Concepts are decomposed to their most basic levels, which are single variables on the one hand and the concepts that are combining two variables on the other hand. These atomic elements are the building blocks of each Boolean expression. By analysing the complexity of combinations of those elements within a Boolean expression Feldman (2006) calculates the Algebraic Complexity value. Following Feldman (2006), this principle should perform better in predicting the difficulty of Boolean concepts than minimal description length due to the assumption, that humans are trying to identify statistical regularities in data sets. The corresponding values of Algebraic Complexity for Boolean concepts presented in this paper are taken from Goodwin and Johnson-Laird (2011) who calculated them based on a Matlab Suite provided by Jacob Feldman.

## **Principles of Reverse Engineering**

Regarding the task of reverse engineering of Boolean concepts Lee and Johnson-Laird (2013) postulates three principles related to difficulty. Those are the principle of variable components, the principle of positive outputs and the principle of dependence. Whereas the number of variable components is not applicable for this paper because all tasks of the experiment had the same amount of variables and thus can not be used to determine differences in difficulty, the other two principles appear to be more promising.

**Principle of Dependence** The principle of dependence takes the interdependency of the different variables into account. It states that "the greater the dependence of components on one another in determining the performance of the system, the harder the system should be to reverse engineer." (Lee & Johnson-Laird, 2013).

With respect to the lightswitch scenario this implies that if every single switch by its own is able to turn the light on and off, the respective components are considered independent. An example is a simple circuit with two switches connected as a disjunction as shown in Figure 3b. There, each switch can change the state of the light regardless of the state of the other switch. In contrast, in Figure 3a is an example for dependent components representing a simple conjunction combining two switches. There, each switch can only have an effect on the light if the other switch is in a certain state. Additionally, there is also the case of partial dependency, e.g.,  $a \wedge (b \lor c)$ . In this case switch a is not able to turn the light on on its own because either switch b or switch c or both have to be on too, but a is capable of turning the light off independently from the state of the other switches.

**Principle of Positive Outputs** The third principle postulated by Lee and Johnson-Laird (2013), the principle of positive outputs, is based on the number of instances, i.e., different variable combinations that fulfil a given Boolean concept. The given example in Figure 4 has the Boolean concept of  $(a \lor c) \land \neg b$  which is fulfilled by the three instances  $a \neg b c$ ,  $a \neg b \neg c$  and  $\neg a \neg b c$  to turn on the light. Consequently the difficulty measure for this concept would be three.

## **Mental Models**

The principle of positive outputs also is the foundation of the Mental Models approach (MM), which can be seen as an extension of the instances approach. It introduces a simplification of the instances to estimate the difficulty (Goodwin & Johnson-Laird, 2011). This idea is founded on the tendency of humans to eliminate unnecessary variables in their mental representations of Boolean concepts. To this end, the total number of instances is reduced by systematically eliminating irrelevant variables in order to merge two instances. The resulting simplified set of instances is considered to be an estimate of the mental models that participants have of the task.

Referring to the example in Figure 4, the three instances for the concept  $(a \lor c) \land \neg b$  can be simplified (see Table 2). The only difference between the first two instances  $a \neg b c$  and  $a \neg b \neg c$  is the third variable c. Obviously if  $a \neg b$  is given, the state of the third variable c is not important because it can be true or false but the light will still shine. Therefore, these two instances are simplified to  $a \neg b$ . However, the third instance can not be simplified any further, leading to a representation with two mental models. The difficulty is then estimated based on the number of mental models (e.g., 2 for the previous example).

#### **Evaluation Data**

The analysis of the present paper is based on the results of an experiment by Goodwin and Johnson-Laird (2011). For the research they used a modified experimental design which is based on the switch-task from Johnson-Laird (1983). The setup consists of three independent switches, similar to Figure 4, that control the light. They used nine concepts concerning

Table 2: Instances for an examplary Boolean concept and their corresponding Mental Models.

Boolean concept	Instances	Mental Models
$(a \lor c) \land \neg b$	$a \neg b c$ $a \neg b \neg c$ $\neg a \neg b c$	$a \neg b \\ \neg a \neg b c$

three binary variables (switch on or off). These selected concepts were from a set of 250 possible concepts from Feldman (2003). Goodwin and Johnson-Laird (2011) chose the taken concepts observing their different complexities. In total, 28 students (12 male, 16 female) participated in the experiment. They were asked to describe the conditions in which the light turns on as a result of the positions of the three independent switches. At the beginning of every task, the switches were all turned off and the participants were presented with test trials to figure out which combinations turned the light on. To change the configuration of the switches they had to press a numbered button which was corresponding to the switch numbers. To see whether the light turned on or not the participants had to submit the configuration. Once they could describe the conditions in which the light turned on, they were able to press the "submit" button and proceed. They had to describe the conditions in their own words on a sheet of paper. During the experiment, participants were not allowed to take notes. If they were insecure about how to answer, they were instructed to describe as clearly as possible. Otherwise the response format was up to the participants.

The descriptions provided as responses by the participants considerably varied, but Goodwin and Johnson-Laird (2011) explained that assessing their accuracy (i.e., the correctness of the description) was straightforward. Two independent editors came to almost the same accuracies when interpreting the participants' descriptions.

While we are mostly relying on the original dataset, we augmented it by also annotating the direction of a description: Goodwin and Johnson-Laird (2011) found that, when describing the Boolean concepts, participants might switch from describing cases where the light would be turned on in the following to describing when the light would be turned off. In the following, we will refer to this as the *direction* of the description. Furthermore, the set of instances that cause the light to be turned on is referred to as the onset, while the offset denotes the set of instances causing the light to be turned off. According to Goodwin and Johnson-Laird (2011), participants might switch the direction in order to make the task easier, i.e., if the onset contains too many elements, a switch to the offset might occur. In the experiment by Goodwin and Johnson-Laird (2011) the change of direction was not explored any further. Still the given answers were considered correct when correctly relying on the offset instead of the onset.

## Method

How good are the performances of the described models on an individual level? Compared to the experiment from Goodwin and Johnson-Laird (2011) the focus of the present paper was to find out how the previous presented models perform on an individual level. The following sections describe precise the analyses and results from the new analyses.

Goodwin and Johnson-Laird (2011) analyzed the previously described accounts for difficulty (Mental Models, Minimal Description Length and Algebraic Complexity) with respect to their ability to account for the difficulty of a concept. They assessed the capabilities of the approaches by comparing the correlations between the estimated difficulty of an approach with the average correctness achieved by participants. However, it remains unclear how the results would translate to an individual level, which will be investigated in the present paper. To this end, we implemented each of the presented approaches as an individualized model.

To facilitate this, we use the CCOBRA-framework<sup>1</sup> to ensure a modeling evaluation standard as proposed by Riesterer, Brand, and Ragni (2020b) with a focus on the models' capabilities to account for individual reasoning behavior. We relied on a coverage task, in which a model is presented with the complete set of information available for a specific individual reasoner, including the responses to all tasks (Riesterer, Brand, & Ragni, 2020a). This allows the model to fit to each reasoner, before it is then queried to replicate the responses for the tasks. To this end, it is important to note that this approach is not useful for testing data-driven models that can store the presented information, but, for cognitive models, provides insights into the model's ability to represent the reasoners response behavior in its parameter space. While it is an optimistic estimate of a models predictive capabilities, the correlation-based evaluations are also performed on the complete information. Therefore, we chose it as it can be seen as an extension of the correlation-based analysis to the individual level.

Each of our models consists of the core mechanism to estimate task difficulty (e.g., Mental Models) and a threshold that is used to decide at which point the difficulty is assumed to be too high for a specific participant (i.e., the difficulty at which the participant started to give incorrect answers). When fitted to an individual participant, the optimal value for the threshold was selected based on the accuracy to replicate the participant's correct responses and errors across all tasks.

Regarding the different approaches, we relied on fixed values for the tasks reported by Goodwin and Johnson-Laird (2011) for the *Minimal Description Length*, the *Algebraic Complexity* and the *Principle of Dependence* (referred to as *Dependency*).

The *Principle of Positive Outputs* was incorporated into a model (referred to as *Instances model*) that directly uses the number of instances as an estimate for the difficulty.

<sup>&</sup>lt;sup>1</sup>https://github.com/CognitiveComputationLab/ccobra



Figure 5: Detailed coverage performances of the models. The box-plots show the prediction accuracies of the implemented models with boxes ranging between the quartiles, the middle line indicating the median performance. The dots within each box-plot depict the single individuals. Triangles denote the mean accuracies.

Finally, the model based on the *Mental Models* approach (MM), relies on the *Instances model* to determine the initial set of instances. In order to rule out potential errors when implementing the simplification, our model then internally uses the original LISP model by Goodwin and Johnson-Laird (2011) for reducing the number of instances.

Two additional models were added as reference points for the performance: First the *Random* model, which determines the estimated correctness randomly (based on a uniform distribution) and can therefore be considered as a lower bound that any model should be able to surpass. Second, another *Baseline* model was included that assumes a perfect correctness by each participants. As the tasks are mostly solved correctly by the participants, it represents an aggregate model that does not consider individual differences. Therefore, it should be surpassed by any model that incorporates mechanisms to adapt to individuals.

## **Directions (Onset & Offset)**

The previously introduced *direction* can serve as an extension for the *Instances model*, and therefore, also of the *Mental Models*. While MM focuses on the instances within the onset to determine the difficulty, the minimization process itself is agnostic of the direction. In a similar fashion, the *Instances model* could also rely on the number of instances in the offset instead of relying on the onset. In order to assess the effect of the direction, we used extended versions of the *Instances model* and the MM that rely on the onset or offset depending on the direction that the respective participant used for the given task. In the case that both directions were present in a participant's response, we used the onset as a default. The approach should be able to enhance the predictive capability of the Instances model and the MM that MM by taking into account

that the difficulty decreases in certain cases if the offset of instances is considered instead of the onset.

#### Results

Figure 5 shows the accuracy achieved by the models when replicating the participants' correctness. As expected, the Random model has the lowest performance with a mean and median accuracy of .5. While there are no differences in the median accuracy for all other models (*median* = .875), they differ with respect to their mean performance. All individualized models surpass the *Baseline* (accuracy = .825), which indicates that they can, at least to a small degree, reflect the individual correctness via the threshold. Overall, the approach based on Mental Models outperformed the other models, with a mean accuracy of .876, with the next best being the Instances model and the Algebraic Complexity with an accuracy of .85. While the Minimal Description model comes close (accuracy = .845), the Dependency model (accuracy = 0.83)barely surpasses the performance of the Baseline model. The additional information provided by including the direction could be used by both, the Instances model and MM. The improvement by the MM was higher (from accuracy = .866to accuracy = .876) compared to the Instance model (from accuracy = .85 to accuracy = .855), but no substantial improvement was apparent. This, however, is more of a general problem: the peformance differences between the models were too small for any meaningful quantitative statement, as neither model was able to significantly outperform the baseline (Mann-Whitney-U between *MM* + *dir* and *Baseline*; U = 306, p = .15). This is likely due to the high ratio of correct responses and the low number of tasks available in the dataset, which leaves only very limited options for the models to set themselves apart from the others.

The amount of participants that achieved a perfect accuracy (n = 8), which could easily be replicated by all models, further reduced an already small dataset. To this end, even participants with only one mistake (n = 9) still do not allow for substantial differences in model performance. However, when considering the individual datapoints, it is possible to see that the individualization did in fact work. When comparing the lower quartile boundaries, it becomes apparent that the models show in fact differences for the individuals that did not always solve the tasks correctly.

## Discussion

In the present article, several estimates for difficulty in Boolean concept tasks were evaluated. In contrast to Goodwin and Johnson-Laird (2011), our analysis was not performed on the basis of correlations between the estimate and the ground truth. Instead, we extended the different estimates to models that should account for the difficulty of a task with respect to an individual participant by introducing an additional threshold representing the maximum difficulty the participant could handle. We evaluated the models on the dataset from Goodwin and Johnson-Laird (2011). While the general trend found by our analysis was in line with the findings by Goodwin and Johnson-Laird (2011), the differences between the models were not significant. Especially when compared to a baseline model, that always assumes that participants solve a task correctly, a fundamental flaw of the dataset when used for model evaluation became apparent. A substantial amount of the participants (8 out of 28) solved every task correctly, with most other participants making only one or two mistakes. This meant that the models had only very limited possibilites to show any differences, which showed in the lack of any significant difference in terms of their performance.

However, some tendencies could be found nevertheless: When focusing on the lower quartiles, the models start to show differences, with the Mental Models having the edge. Furthermore, the inclusion of the direction, which was already expected to have an influence by Goodwin and Johnson-Laird (2011), did in fact allow the models to improve. MM was able to benefit more than the Instance model, which corroborates the assumption of MM that a simplification is in fact performed by reasoners.

From a more general perspective, the present analysis showed the importance of model evaluation on different settings, especially with a focus on individual participants. The different approaches differed substantially based on correlations alone, but did not translate to a more simulationoriented setting, where a precise response to a task should match the response of a specific participant. To this end, the proposed evaluation with a well-defined setting and implemented, individualized models can serve as a first step.

Cognitive modeling should strive for the creation of models that are able to account for the human behavior, with as little interpretation and preprocessing of the recorded behavior as possible. The foundation to this also lies in a suitable data foundation, as model evaluation requires the ability to distinguish between different models. To this end, a suitable dataset should not only consist of a big corpus of participants, but should above all offer a large variety of tasks, which allows to find meaningful patterns in participants' responses. If the selected tasks are too easy or too difficult, evaluation will be impeded by ceiling/floor effects. In the setting of Boolean concepts, an extension of the tasks to tasks with more variables would also be important to add another dimension in which models and theories can differ. Furthermore, with a solid data foundation, the task can be extended from estimating the correctness into the task of predicting the precise description of the concept provided by a participant (in a standardized simplified way, e.g., by translating the description to a Boolean concept in a preprocessing step). Solving such a task, even in a simplified version, would require models to show a much deeper understanding of the reasoning processes that underlie solving the tasks.

## References

- Feldman, J. (2000). Minimization of boolean complexity in human concept learning. *Nature*, 407(6804), 630–633.
- Feldman, J. (2003). A catalog of boolean concepts. *Journal* of Mathematical Psychology, 47(1), 75–89.
- Feldman, J. (2006). An algebra of human concept learning. *Journal of mathematical psychology*, *50*(4), 339–368.
- Goodwin, G. P., & Johnson-Laird, P. (2011). Mental models of boolean concepts. *Cognitive psychology*, 63(1), 34–59.
- Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and conscious-ness* (No. 6). Harvard University Press.
- Lee, N. L., & Johnson-Laird, P. (2013). A theory of reverse engineering and its application to boolean systems. *Journal of Cognitive Psychology*, 25(4), 365–389.
- Love, B. C. (2002). Comparing supervised and unsupervised category learning. *Psychonomic bulletin & review*, 9(4), 829–835.
- 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). Predictive modeling of individual human cognition: Upper bounds and a new perspective on performance. *Topics in Cognitive Science*, *12*(3), 960–974. doi: 10.1111/tops.12501
- Shepard, R. N., Hovland, C. I., & Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological monographs: General and applied*, *75*(13), 1.
- Smith, J. D., Minda, J. P., & Washburn, D. A. (2004). Category learning in rhesus monkeys: a study of the shepard, hovland, and jenkins (1961) tasks. *Journal of Experimental Psychology: General*, *133*(3), 398.

## A Computational Cognitive Theory of Temporal Reasoning

## Sangeet Khemlani

sunny.khemlani@nrl.navy.mil Navy Center for Applied Research in Artificial Intelligence US Naval Research Laboratory, Washington, DC 20375 USA

#### Abstract

I describe a novel model-based theory of how individuals reason deductively about temporal relations. It posits that temporal assertions refer to mental models -- iconic representations of possibilities -- of events. In line with recent accounts of spatial reasoning, the theory posits that individuals tend to build a single preferred model of a temporal description. The more models necessary to yield a correct answer, the harder that problem is. The theory is implemented in a computer program, mReasoner, which draws temporal deductions by building models. It varies three parameters governing separate factors in the process: the size of a model, the typicality of its contents, and the propensity to search for alternative models. Two experiments corroborate the predictions of the theory and its computational implementation. I conclude by discussing temporal and relational inference more broadly.

**Keywords:** temporal reasoning; events; mental models; reasoning; simulation

## Introduction

People make temporal inferences when they schedule future events, reconcile past experiences, and attempt to understand ongoing scenarios. For instance, consider this description:

1. The car hit a pothole during the road trip. The car broke down after the road trip. Does it follow that the car hit a pothole before it broke down?

The words *during* and *after* are temporal relations – they describe how events and outcomes relate to one another – and reasoners have no difficulty inferring the correct answer ("yes") from the premises in (1). Indeed, English and other natural languages encode tense and aspect into every utterance, and so they provide abundant cues for drawing temporal conclusions. But some inferences are systematically easier than others. Contrast the example above with this problem (adapted from Schaeken et al., 1996):

2. The car hit a pothole before the road trip. The car's radio broke before the road trip. The car's windshield cracked when it hit a pothole. The car's headlights fused while on the road trip. Does it follow that the car's windshield cracked before its radio broke?

The correct answer – "no" – seems more difficult to infer compared to (1). Why? Many factors distinguish (2) from (1): it has more premises, and describes more events; it uses more temporal relations – *before*, *when*, and *while*; and its correct

answer is negative instead of affirmative. Yet these factors don't provide an adequate explanation of the mental representations and processes humans use to reason about time. And no computational cognitive theory exists that's robust enough to simulate why (1) is an easy inference to make; why (2) is more difficult (though cf. the computer model described in Schaeken et al., 1996); and how people generate rational responses to even difficult temporal reasoning problems.

In what follows, I briefly summarize previous computational treatments of temporal deduction, and show how they are psychologically implausible. Next, I synthesize a theory of temporal reasoning based on how humans simulate the passage of time. I argue that to reason about time, humans can construct a mental timeline of events, i.e., an event model. These event models are easy to process when humans build and maintain just one in memory, but difficult to process when they need to maintain multiple event models. I describe a computational implementation of the theory and the predictions it makes, as well as a series of studies designed to test those predictions. And I show how the computational implementation fits data from those studies, and how it can model additional forms of temporal inference. I conclude by contrasting the theory with alternative proposals.

#### The logic of temporal reasoning

Systems of symbolic logic are designed to generate the correct answers to problems such as (1) and (2). Temporal logics, such as Prior's (1967) tense logic and Allen's (1983) interval calculus, treat each premise as a formula describing a temporal relation between events, and can be written as, e.g., during  $(\mathbf{X}, \mathbf{Y})$  where **x** can stand in place for any proposition, such as hitAPothole(*car*). Many systems of temporal logic in AI (e.g., Allen, 1991; Freksa, 1992; Øhrstrøm & Hasle, 1995; see also Fischer, Gabbay, & Vila, 2005; Goranko, Montanari, & Sciavicco, 2004 for reviews) posit primitive temporal relations that do not map into simple everyday English (Knauff, 1999) or other natural language expressions, and likewise, many temporal relations in natural language are flexible in ways AI systems cannot characterize. For instance, AI systems often neglect Reichenbach's (1947) distinction between different points of reference in natural language, and so they are insensitive to the distinction between, e.g., "I had done it" (past perfect tense) versus "I did it" (past tense). For many AI applications, these distinctions are irrelevant - but they ensure that such systems cannot interface with the full range of natural language capabilities (see, e.g., Khemlani & Johnson-Laird, 2019).

Event calculi (see, e.g., Kowalski & Sergot, 1986) may be more psychologically plausible, because they describe inference rules and axioms between two or more temporal relations, so they abide by the constraints of logic-based cognitive accounts of reasoning (e.g., Rips, 1994; see also Bringsjord & Govindarajulu, 2020). But, a limitation common to all temporal logics and event calculi are that they describe only valid deductions: they have no capacity of explaining what happens when reasoners err. And so they cannot explain why (1) is easy and why (2) is hard. For that, we turn to a psychological theory of temporal reasoning.

## Mental models of events

The theory of temporal deduction I present is based on the tenets of mental model theory – the "model" theory for short (Johnson-Laird, 2006). The theory states that when people reason, they use language observation, and imagination to construct and mentally manipulate possibilities. The theory is based on several fundamental principles:

- Mental models are iconic representations of possibilities. That is, the structure of a mental model corresponds to the structure of what it represents as far as possible (Peirce, 1931-1958, Vol. 4). Models of temporal relations can use space to represent time by constructing mental timelines in which tokens represent events (Schaeken et al., 1996), or they can represent sequences of events as they unfold in time (Khemlani et al., 2013).
- Models represent durations as discrete episodes. Reasoners encode durations and intervals by representing episodes that mark the starts and ends of events (Khemlani et al., 2015a). By default, people do not maintain representations of metric time. To comprehend specific intervals, as in, *the meeting lasted 2 hours*, individuals tag events with ancillary information, and then reason arithmetically.
- The principle of emergent consequences. Logical relations are emergent consequences of iconic structure of models and so no special logical rules, operations on formulas, or syntactic transformations are necessary for individuals to reason logically (Goodwin & Johnson-Laird, 2005).
- Inferences are easier with one model; multiple models yield errors. Human reasoning is based on two interacting sets of processes: one system produces rapid, intuitive inferences by building a single model. Hence, people are faster and make fewer errors when considering descriptions that yield only one model. When descriptions yield multiple models, i.e., when an initial model doesn't suffice, reasoners are more prone to errors (Khemlani & Johnson-Laird, 2017) and they take longer (Schaeken & Johnson-Laird, 2000).

The model theory posits that to simulate relations between events, people have two options: first, they can simulate a series of events in the same order as they would unfold. For example, to represent an individual's meals over the course of a day, you might simulate the individual eating breakfast, then lunch, then dinner, focusing only on each single meal at a time. By doing so, reasoners build kinematic mental models, i.e., they use time to represent time (Khemlani et al., 2013). Kinematic models may be particularly useful when following complex narratives, e.g., during discourse comprehension (Cain & Oakhill, 1999; Garnham, 2013; Graesser, Millis, & Zwaan, 1997; Zwaan & Rapp, 2006), though they can obscure the temporal relations between simultaneous events and events with durations. To reason directly about such relations, people can use space to represent time (Schaeken et al., 1996, 2000), e.g., they can construct a mental model for (1) in a way that can be depicted in the following diagram:

```
[ road-trip ] broke-down
hit-pothole
```

The diagram represents the events iconically, i.e., with words that stand in place of mental simulations of the event itself. Its spatial layout presents the events in chronological order, from earliest to latest (see Kelly & Khemlani, 2020, under review). And it uses markers to designate the initiation ([) and conclusion (]) of a durative event, namely to depict that hitting a pothole occurred in the time between when the road trip started and ended (Kelly, Khemlani, & Johnson-Laird, 2020). The logical consequences emerge from the model's structure – by scanning it, reasoners can draw many different valid conclusions, e.g.,

the road trip happened before the car broke down; the car hit a pothole before the breakdown; the road trip ended after the car hit a pot-hole; the road trip started before the breakdown;

and so on. Hence, the model serves as a compact, efficient representation to facilitate reasoning.

Models predict difficulty, because inferences that require multiple models place a higher demand on working memory resources. So, what makes (2) difficult is not just that it has more premises, or more relations. Rather, it's difficult because the description yields multiple models. This model satisfies the premises:

hit-pothole	radio-broke	Γ	road-trip	]
windshield			fused-headlight	

but so does this one:

radio-broke	hit-pothole	[ road-trip	]
	windshield	fused-headlight	

Hence, the conclusion in (2) doesn't follow necessarily. To get the correct answer, reasoners must either initially build the second model above, or else keep both models in mind and compare the two. The theory accordingly predicts that all other things being equal, inferences that demand more models should be more difficult – they should produce more errors. I turn next to describe a computational implementation of the theory.

## Temporal reasoning in mReasoner

mReasoner is a computational cognitive reasoning engine that implements the core tenets of the model theory (Khemlani & Johnson-Laird, 2022). The system is equipped with a small grammar that parses and builds iconic mental models for assertions concerning quantity (e.g., "Most of the potholes are large"), causality (e.g., "Hitting the pothole caused the breakdown"), and sentential inference (e.g., "The windshield cracked or else the headlights fused"), and it mimics patterns of human reasoning in all these domains (Briggs & Khemlani, 2019; Khemlani et al., 2015b, 2018). Updates to its components permitted it to reason about temporal relations. Figure 1 depicts a schematic of the system and shows how it draws the correct conclusion for (1). I review each updated component and their functionality in turn.

#### **Building integrated models**

The first component parses premises from natural language into *intensions*, which serve as blueprints for building models. Intensions provide a modal semantics for the meaning of an assertion. The system parses a variety of different temporal assertions, e.g., those describing connectives such as *before, after, while,* and *during*. The intensions of each assertion specify how to construct an initial model, as well as serve as a guide to the space of possible revisions on the model (see Khemlani & Johnson-Laird, 2022). The semantics is as follows:

A happened before B.	$\rightarrow$	A < B
A happened after B.	$\rightarrow$	A > B
A happened while B.	$\rightarrow$	A = B
A happened during B.	$\rightarrow$	$A \subseteq E$

mReasoner builds temporal models by integrating multiple temporal intensions, e.g., it builds an initial model of the first assertion in (1), and then updates that model with information about the second assertion, yielding an integrated model of the two relations. One subtlety of this procedure is that it is sensitive to the order in which it processes premises in that, by default, the system treats events in premises as punctate – but when necessary, it converts a punctate event into a durative one. This example illustrates the phenomenon:

The meeting happened before the conference. The sale happened during the conference.

As in all temporal assertions, the events (*the meeting, the conference*) can be treated as single points or multiple points on a timeline. The system starts by building a model of the first premise:

```
meeting conference
```

(such that *the sale* is contained within an interval), mReasoner breaks the punctate event into two markers because the second premise treats *the conference* as durative by explicitly represent its start and end, e.g.,

"The car hit a pothole during the road trip." "The car broke down after the road trip." "Does it follow that: the car hit a pothole before it broke down?"



mReasoner output

Figure 1. Four components of the mReasoner computational cognitive model that generate conclusions given temporal premises. The system *parses* premises into intensions; *builds* an initial model from those intensions; *scans* the model to locate events in a given premise; and *validates* a relation between those located events. If its deliberative system is engaged, mReasoner can engage a search for counterexamples to decide whether its initial conclusion necessarily follows, and it can modify the conclusion if necessary (see Khemlani & Johnson-Laird, 2022).

Another subtlety of the model-building component concerns how to construct indeterminate descriptions. Consider this set of premises:

The ceremony happened before the storm. The newscast happened before the storm.

By default, mReasoner constructs and reasons with a single model at a time. But the description above is consistent with several different models, e.g., one in which the sale happens before the meeting, another in which the sale happens after the meeting, a third in which the meeting happens during the sale, and so forth. To build an initial model from indeterminate descriptions, mReasoner adopts heuristic strategies for constructing models initially developed for a theory of spatial reasoning (see Ragni & Knauff, 2013, p. 567). That is, by default mReasoner inserts new events at the first available location:

```
\texttt{ceremony storm} \rightarrow \texttt{newscast} \texttt{ ceremony storm}
```

but it can also insert events so that they occur in a way that "spreads apart" existing events in the model, as in:

ceremony storm  $\rightarrow$  ceremony **newscast** storm

These two strategies are governed by a probabilistic typicality parameter that ranges from 0 to 1, and controls the probability of engaging in the latter strategy (see Johnson-Laird et al., 2015; Khemlani & Johnson-Laird, 2022 for additional information on this parameter). In this way, the system mimics the variation in humans' construction of temporal models.

# Scanning models, drawing conclusions, and searching for counterexamples

To draw conclusions, mReasoner scans an integrated model with respect to a given temporal conclusion. For example, it scans an integrated model of (2) above for the two events specified in the conclusion (i.e., the windshield cracking and the reading breaking). If the events are represented in the model, the system generates an intension that describes their temporal relation, and converts that intension back into natural language – and if that relation happens to match the prompt (*the windshield cracked before the radio broke*) then the system responds affirmatively. In all other cases, including those in which it cannot locate an event in the model, the system responds negatively.

As previous investigations of temporal reasoning reveal, humans are able to reason about extraordinarily complex temporal descriptions. Hence, descriptions that concern multiple mental models may pose difficulties for reasoners, but many reasoners are skilled in their ability to overcome such difficulties. A viable theory of temporal reasoning must explain, not just why some problems are more difficult, but how certain individuals manage to provide correct responses despite such difficulties. The model theory proposes - and mReasoner implements - the idea that rational responses often depend on the interrogation of initial responses: people recognize, for instance, that descriptions are consistent with multiple models, and so they attempt to build those models. Their attempts may result in a model in which the premises are true but their initial conclusion is false - i.e., a "counterexample". The model theory further proposes that the search for counterexamples is not a sampling procedure (pace Phillips, Morris, & Cushman, 2019). Instead, the theory posits that reasoners make incremental changes to the events represented in their initial model. Evidence supporting such a procedure comes from the fact that difficult problems are easier when they require counterexamples that have a smaller "edit distance" to the initial model (Ragni, Khemlani, & Johnson-Laird, 2014).

For temporal reasoning, counterexample search depends on some combination of the following 5 strategies: i) shifting an event earlier in time; ii) shifting an event later in time; iii) converting a punctate event into a durative one; iv) converting a durative event into a punctate one; v) expanding a durative event, i.e., shifting a token representing its start to an earlier time and a token representing its end to a later time. The system attempts each strategy in turn in a recursive fashion, and stops when it discovers a counterexample. But, its ability to search for counterexamples in the first place is not turned on by default. It is governed by a *search* parameter (see Khemlani & Johnson-Laird, 2022) that controls the probability of engaging in a search for counterexamples.

These augmentations to the mReasoner computational model provide it with the means to mimick human temporal reasoning. The next section describes experiments designed to test the theory's prediction that one-model problems are easier than multiple-model problems; and the section that follows describes mReasoner's fit to the resulting data.

## **Experiments 1 and 2**

We conducted two experiments to test the computational model described in the previous section. Each experiment presented participants with the same 8 reasoning problems, though the contents of the premises were randomized. Here is an example problem:

The suspect set up surveillance before he closed his bank account. The suspect destroyed the laptop after he closed his bank account. The suspect hired the lawyer while he set up surveillance.

The model theory predicts that the problem should be easy, since the premises in Experiment 1 are consistent with only one model, this one:

surveillance closed-account destroyed-laptop
hired-lawyer

Half of the problems were consistent with one model, and the other half were consistent with multiple models. In all other respects, namely, the specific contents, the number of premises, the events in the premises, and the number and type of temporal relation, the 4 one-model problems and 4 multiple-model problems were matched.

Participants in Experiment 1 were given three separate conclusions: a valid conclusion, a foil, and a null conclusion, i.e., "there's not enough information to conclude anything". Participants in Experiment 2 carried out the same problems, but instead generated their own responses to questions of the form:

What is the relationship between when the subject hired the lawyer and when he destroyed the laptop?

Participants' natural responses were coded for accuracy.

## Method

*Participants*. A total of 61 participants (31 in Experiment 1 and 30 in Experiment 2) were recruited through Amazon Mechanical Turk. Participants who failed to answer attention checks, misunderstood the task, or performed the entire study under 2 minutes were dropped from analysis. This resulted in data from 56 participants (28 in Experiment 1 and another 28 in Experiment 2).

Design, procedure, and materials. Each participant was presented with 10 three-premise causal inference problems: 4 were predicted to be one-model problems and 4 that were multiple-model. The other 2 were practice problems that also served as attention checks, and were discarded from analysis. Each problem consisted of three premises describing temporal relations that were randomly selected from a pool of events that described the activities of a criminal suspect, e.g., "shredded the documents", "transferred the drug funds", "build the explosive", and so on. Each premise consisted of a pair of activities linked by 1 of 4 temporal connectives (*before, after, during,* and *while*). The activities were chosen such that they could be interpreted as durative or punctate (see Kelly et al., 2020), and yield a coherent narrative no matter how they were ordered (e.g., in the example above, the
narrative would be coherent even if the suspect hired a lawyer before he transferred the drug funds). The order in which the participants carried out the 10 problems was randomized, as was the assignment of the contents of the premises.

*Task.* Experiment 1 provided participants with three response options: a valid conclusion, an invalid conclusion, and a null conclusion. In Experiment 2, participants typed out their responses to a question relating two events in the problem, i.e., "What is the relationship between \_\_\_\_ and \_\_\_?" I coded their responses for accuracy blind to the specific condition.

Open science. Data, materials, experimental code, mReasoner code. and synthetic data derived from computational simulations are available at https://osf.io/26ckg/.

### Results

Both experiments showed that participants were more accurate for one-model problems than multiple-model problems (in Experiment 1, one- vs. multiple-model: 70% vs. 37%; Wilcoxon test, z = 5.08, p < .001, Cliff's  $\delta = .33$ ; in Experiment 2, one- vs. multiple-model: 78% vs. 44%; Wilcoxon test, z = 5.25, p < .001, Cliff's  $\delta = .35$ ). The results corroborate the model theory of temporal reasoning. In addition, Experiment 2 captured the response time between when participants read the three premises and when they began to type out a response. Analysis of Winsorized response times revealed that participants were faster to respond to one-model problems (51.77 s) than multiplemodel problems (60.01 s; Wilcoxon test, z = 3.15, p = .002, Cliff's  $\delta = .19$ ). These results, too, corroborate the model theory's difficulty prediction. For brevity, I omit further analyses of the data in favor of describing the mReasoner's simulations of the two studies.

### Simulation of Experiments 1 and 2

To simulate the 8 problems in Experiments 1 and 2, mReasoner generated datasets by systematically varying the settings of two of its parameters (Busemeyer & Diederich, 2010), i.e., the *atypicality* and *search* parameters described above, along with a *size* parameter that stochastically limited the size of each model. The parameter settings were quantized to span their ranges as follows:

size: 2.0, 2.5, 3.0, 3.5, 4.0, 4.5\*, 5.0\*\*
atypicality: 0.0, 0.2, 0.4, 0.6, 0.8, 1.0
search: 0.0, 0.2, 0.4, 0.6, 0.8, 1.0

```
* Exp 1 ** Exp 2
```

Hence, the system generated  $7 \times 6 \times 6 = 252$  separate simulated datasets. The system carried out the 8 problems 100 times for each of the 252 parameter settings. A grid search was used to locate the best fitting parameter settings for the data for Experiments 1 and 2. The grid search depended on minimizing the root mean squared error (RMSE) between the dataset and the proportions of responses in each simulated dataset across the 8 problems. Once the grid search located the best-fitting parameter settings (which were quite similar, and bolded above), the parameters were fixed and mReasoner carried out the 8 problems 1000 times each. Figure 2 plots the computational modeling simulations against the results from each dataset.

The computational model yielded a close fit to the data (r = .95, RMSE = .22 for Experiment 1; r = .93, RMSE = .20). And the optimizing parameter values located from by the grid search were sensible: the computational model fit the data when the size of the models was large (> 4), when the system considered atypical models 40% of the time, and when the system never engaged in a search for counterexamples. Searching for counterexamples is demanding, and most of the time, particularly for complex problems, reasoners appear to satisfice and base their inferences on the first model they construct.



### Experiment 1

Figure 2. The proportions of correct responses to the 8 problems in Experiments 1 and 2, along with the proportions of correct responses generated by mReasoner's best-fitting simulations (rs = .95 and .93 for Experiments 1 and 2, respectively). The 8 problems in the two experiments are provided using schematic formulas in place of the natural language sentences participants received, e.g., participants saw premises akin to, "The suspect destroyed the laptop after he closed his bank account" instead of after (y, x). Participants' evaluated a given response – denoted by the question mark – in Experiment 1, and specified the relation between two events in Experiment 2.

### **General discussion**

Human reasoning about time is complex: events can be punctate or stretch across other events; they can be cyclical, as in the passage of seasons; and they can endure across fixed units that can be enumerated. Nevertheless, humans must make rapid inferences about relations to understand narratives and plan for future scenarios. I describe a theory of temporal cognition that relies on the construction, maintenance, and manipulation of event models. The theory accounts for how people represent durations and what makes reasoning about time difficult. The theory is embodied in mReasoner, a computational cognitive implementation of the model theory of thinking and reasoning (Khemlani & Johnson-Laird, 2022). In this paper, I described innovations to the system that can predict which temporal reasoning problems prompt reasoners to make errors; I described experiments designed to test the theory's central predictions; and I showed how the computational model fit the data from those studies.

The computational model explains only a small subset of temporal reasoning phenomena: it doesn't account for how people rapidly process and interpret tense and aspect, or how they cope with information about metric time. But, the theory does explain how people without any background in temporal logic can make valid deductions from temporal premises. Previous psychological accounts of reasoning have argued that people maintain axiom systems and build proofs to make temporal deductions (see, e.g., Rips, 1994). Meanwhile, probabilistic frameworks of reasoning either build off such logical frameworks, or else eschew any consideration of temporal inference whatsoever (see Knauff & Gazzo Castańeda, 2022). Neither approach can explain the systematic errors people make in Experiments 1 and 2. The system I describe therefore serves as a cognitive process model of temporal reasoning: it specifies both the structure of the mental simulations people build, as well as the algorithms they use to process those representations.

#### Acknowledgments

This research was supported by funding from the Naval Research Laboratory to study temporal reasoning. I'm grateful to Kalyan Gupta, Danielle Paterno, Kevin Zish, and Bryan Auslander at the Knexus Research Corporation for their help in conducting the experiments. For their helpful suggestions and critiques, I also thank Bill Adams, Branden Bio, Gordon Briggs, Ruth Byrne, Hillary Harner, Tony Harrison, Phil Johnson-Laird, Laura Kelly, Markus Knauff, Andrew Lovett, Marco Ragni, and Greg Trafton.

#### References

- Allen, J. F. (1983). Maintaining knowledge about temporal intervals. Communications of the ACM, 26, 832–843.
- Allen, J. F. (1991). Time and time again: The many ways to represent time. International Journal of Intelligent Systems, 6, 341–355.
- Briggs, G. & Khemlani, S. (2019). A cognitively plausible algorithm for casual inference. In T. Stewart (Ed.), *Proceedings of the 17th International Conference on Cognitive Modeling.*
- Bringsjord,S., & Govindarajuli, N. S. (2020). Rectifying the mischaracterization of logic by mental model theorists. *Cognitive Science*, *44*, e12898.

Busemeyer, J. R., & Diederich, A. (2010). Cognitive modeling. Sage.

- Cain, K., & Oakhill, J. (1999). Inference making ability and its relation to comprehension failure in young children. *Reading and Writing*, 11.
- Fisher, M. D., Gabbay, D. M., & Vila, L. (2005). Handbook of temporal reasoning in artificial intelligence. Amsterdam: Elsevier.
- Freksa, C. (1992). Temporal reasoning based on semi-intervals. Artificial Intelligence, 54, 199–227.
- Garnham, A. (2013). Mental models and the interpretation of anaphora. Psychology Press.
- Goodwin, G.P., & Johnson-Laird, P.N. (2005). Reasoning about relations. Psychological Review, 112.
- Goranko, V., Montanari, A., & Sciavicco, G. (2004). A road map of interval temporal logics and duration calculi. *Journal of Applied Non-Classical Logics*, 14, 9–54.
- Graesser, A. C., Millis, K. K., & Zwaan, R. A. (1997). Discourse comprehension. Annual Review of Psychology, 48, 163-189.
- Johnson-Laird, P.N. (2006). How we reason. NY: Oxford University Press.
- Johnson-Laird, P. N., Khemlani, S., & Goodwin, G.P. (2015). Logic, probability, and human reasoning. *Trends in Cognitive Sciences*, 19.
- Kelly, L. & Khemlani, S. (2020). Directional biases in durative inference. In B. Armstrong, S. Denison, M. Mack, and Y. Xu (Eds.), *Proceedings of* the 42nd Annual Conference of the Cognitive Science Society. Austin, TX: Cognitive Science Society.
- Kelly, L., & Khemlani, S. (under review). Iconicity bias and duration. Manuscript under review.
- Kelly, L., Khemlani, S., & Johnson-Laird, P.N. (2020). Reasoning about durations. *Journal of Cognitive Neuroscience*, 32, 2103-2116.
- Khemlani, S., Byrne, R. M., & Johnson-Laird, P. N. (2018). Facts and possibilities: A model-based theory of sentential reasoning. *Cognitive Science*, 42, 1887-1924.
- Khemlani, S., Harrison, A. M., & Trafton, J. G. (2015a). Episodes, events, and models. *Frontiers in Human Neuroscience*, 9, 590, 1-13.
- Khemlani, S., & Johnson-Laird, P.N. (2013). The processes of inference. Argument & Computation, 4, 1-20.
- Khemlani, S., & Johnson-Laird, P.N. (2017). Illusions in reasoning. Minds & Machines, 27, 11-35.
- Khemlani, S., & Johnson-Laird, P.N. (2022). Reasoning about properties: A computational theory. Manuscript in press at *Psychological Review*.
- Khemlani, S., & Johnson-Laird, P.N. (2019). Why machines don't (yet) reason like people. *KI Künstliche Intelligenz*, 33, 219-228.
- Khemlani, S., Lotstein, M., Trafton, J.G., & Johnson-Laird, P.N. (2015b). Immediate inferences from quantified assertions. *Quarterly Journal of Experimental Psychology*, 68.
- Khemlani, S., Mackiewicz, R., Bucciarelli, M., & Johnson-Laird, P.N. (2013). Kinematic mental simulations in abduction and deduction. *Proceedings of the National Academy of Sciences*, 110.
- Khemlani, S., Orenes, I., & Johnson-Laird, P.N. (2012). Negation: a theory of its meaning, representation, and use. *Journal of Cognitive Psychology*, 24.
- Khemlani, S., Wasylyshyn, C., Briggs, G., & Bello, P. (2018). Mental models of omissive causation. *Memory & Cognition*, 46.
- Knauff, M. (1999). The cognitive adequacy of Allen's interval calculus for qualitative spatial representation and reasoning. *Spatial Cognition and Computation*, 1, 261–290.
- Kowalski, R., & Sergot, M. (1986). A logical-based calculus of events. New Generation Computing, 4, 67–95.
- Peirce, C.S. (1931-1958). Collected papers of Charles Sanders Peirce. 8 vols. C. Hartshorne, P. Weiss, and A. Burks, (Eds.). Cambridge, MA: Harvard University Press.
- Phillips, J., Morris, A., & Cushman, F. (2019). How we know what not to think. *Trends in Cognitive Sciences*, 23, 1026-1040.
- Ragni, M., Khemlani, S., & Johnson-Laird, P.N. (2014). The evaluation of the consistency of quantified assertions. *Memory & Cognition*, 42, 53-66.
- Ragni, M., & Knauff, M. (2013). A theory and a computational model of spatial reasoning with preferred mental models. *Psychological Review*, 120, 561.
- Reichenbach, H. (1947). Elements of symbolic logic. New York: Free Press. Rips, L. J. (1994). The psychology of proof: Deductive reasoning in human thinking. Cambridge, MA: MIT Press.
- Schaeken, W., & Johnson-Laird, P. N. (2000). Strategies in temporal reasoning. *Thinking & Reasoning*, 6, 193–219.
- Schaeken, W., Johnson-Laird, P. N., & d'Ydewalle, G. (1996). Mental models and temporal reasoning. *Cognition*, 60, 205–234.
- Zwaan, R. A., & Rapp, D. N. (2006). Discourse comprehension. In Handbook of psycholinguistics (pp. 725-764). Academic Press.

# **Evolving Understandable Cognitive Models**

Peter C.R. Lane (p.c.lane@herts.ac.uk)

School of Physics, Engineering and Computer Science, University of Hertfordshire, College Lane, Hatfield AL10 9AB, UK

#### Laura Bartlett (l.bartlett@lse.ac.uk)

Noman Javed (n.javed3@lse.ac.uk)

Angelo Pirrone (a.pirrone@lse.ac.uk)

#### Fernand Gobet (f.gobet@lse.ac.uk)

Centre for Philosophy of Natural and Social Science, London School of Economics, Houghton Street, London WC2A 2AE, UK

#### Abstract

Cognitive models for explaining and predicting human performance in experimental settings are often challenging to develop and verify. We describe a process to automatically generate the programs for cognitive models from a user-supplied specification, using genetic programming (GP). We first construct a suitable fitness function, taking into account observed error and reaction times. Then we introduce post-processing techniques to transform the large number of candidate models produced by GP into a smaller set of models, whose diversity can be depicted graphically and can be individually studied through pseudo-code. These techniques are demonstrated on a typical neuro-scientific task, the Delayed Match to Sample Task, with the final set of symbolic models separated into two types, each employing a different attentional strategy.

**Keywords:** cognitive modelling, genetic programming, model visualisation

#### Introduction

Developing and verifying the behaviour of cognitive models is a non-trivial task. Ideally, a cognitive model will provide some explanation of how a human performs in a particular experimental setting, and even provide predictions for new settings. In many cases cognitive models are based around computer programs which need to be written. The area of *program synthesis* studies ways to generate executable computer programs from user specifications. In this paper we demonstrate how an evolutionary algorithm can generate programs representing candidate computational models in a typical neuro-scientific experiment. We present techniques to improve the understandability of the resulting programs, which enables their use as the starting point for developing scientific theories.

We use the evolutionary algorithm Genetic Programming (GP) (Koza, 1992) to search a space of programs. The fitness function guiding this search is designed to find programs which effectively simulate the behaviour of human subjects. Unlike many typical GP applications, this fitness function is not based directly on an input-output mapping for the program. In particular, human subjects do not achieve 100% success in our example task, and so the 'best' model is one which replicates this less-than-perfect accuracy. Also, as the responses made by a human take a certain amount of physical time, a simulated time for the program to convert each input into an output must be measured and compared with the observed response time. These two performance measures must

be captured in a combined fitness function: how to do this effectively is the first contribution of this paper.

The GP system often generates a large number of candidate models: we want to convert these into a small set of representative, understandable and qualitatively different models. We achieve this with a series of post-processing steps to remove unnecessary operators from the programs and remove duplicates. Finally, the programs can be changed to pseudo-code, for further analysis, and a visualisation made to highlight the relationships between the final solutions. These techniques form our second contribution.

# Background

Cognitive modelling is a process in which computational models of a target behaviour are sought in an attempt to understand human behaviour. These computational models are typically developed within a given framework, such as a symbolic (Simon, 1981) or connectionist (Rumelhart & McClelland, 1986) framework. In this paper, we consider a framework of symbolic models, typical of cognitive architectures such as ACT-R (Anderson & Lebière, 1998) or CHREST (Gobet & Simon, 2000). However, even within a single framework, there are still many possible models which could be developed, each with qualitatively different behaviour. For example, the manner in which a visual scene is scanned for information could be systematic and wideranging, or task-oriented and narrow, and either way could be sufficient for achieving the target performance: scientifically, it is useful to be aware of both possibilities, but often time constraints or natural bias (oversights) lead to models written by human programmers being constrained to particular groups of solutions.

Using search algorithms to explore a solution space for one or more candidates is a technique with a long history (Langley, Simon, Bradshaw, & Zytkow, 1987; Schmidt & Lipson, 2009). GP approaches to this exploration are also widely known, although there appear to be few studies in the area of cognitive science, exceptions being Frias-Martinez and Gobet (2007); Lane, Sozou, Gobet, and Addis (2016).

Our approach using GP appears unique in developing cognitive models which focus on symbolic, information-processing (Simon, 1981) explanations of human cognition.



Figure 1: Illustration of DMTS task. (Photos by Danny de Bruyne and Ronaldo Taveira, freeimages.com)

This contrasts with many current approaches in artificial intelligence which rely on connectionist (statistical) explanations based on large datasets: a recent study in this area is that of Peterson, Bourgin, Agrawal, Reichman, and Griffiths (2021).

# **Model Development System**

Our proposed system for automatically developing cognitive models is an example of *program synthesis*. Such systems can be conveniently divided into three parts (Gulwani, 2010): the task definition (*user intent*), to express what makes a good program; a *search space* of candidate programs; and a *search technique*, to explore the given search space for good programs. Here, the developed programs form the control structure for the cognitive models.

### Task definition: DMTS

The task studied in this paper is a typical neuro-scientific experiment, popular for studies of short-term memory, which tests the accuracy and reaction time for subjects to recognise images: this is the Delayed Match to Sample (DMTS) task (Chao, Haxby, & Martin, 1999). In this experiment, illustrated in Figure 1, a picture is presented for 1 second in the center of the screen. Then, after a delay of 0.5 seconds, two pictures are presented for 2 seconds, one on the left and the other on the right of the screen. The participant has to select which of those two pictures is the same as the first picture.

The cognitive model must coordinate the perception of time-sensitive information with accurate responses within expected response times. We simplify the task by abstracting away the recognition of images: we have six 'images', represented by the cardinal numbers from 1 to 6.

Although this task is an example of "programming-byexample", where the model must reproduce the example input-output behaviour, the overall quality of the model is not judged on the number of correct input-output pairs. As reported in Chao et al. (1999), across the complete set of presentations, human subjects only score 95.7% accuracy, with an average response time of 767ms: the model's accuracy and simulated response times are judged against these values.

### **Search Space: Cognitive Models**

Each individual cognitive model is defined by a control program to be interpreted within a simple cognitive architecture. This architecture has some task-specific input/output components: a set of *inputs* and a *response*. It also has some task-independent components: a fixed-size short-term memory (STM), and a working memory *current*. Finally, each model has a *clock*, to record its current in-task time.

The model control program is composed from a set of operators, listed in Table 1. These operators define a simple imperative programming language, where operators can be combined in sequence, selected with a conditional statement, and repeated in fixed-cycle loops. The model's current working value, STM and clock values can all be manipulated, inputs read and a response prepared: the current response is "made" when the program ends. Operators are arranged in groups, matching their simulated execution time (based on estimates from the psychological literature): *input operators* (100ms), *output operators* (140ms), *cognitive operators* (70ms), *STM operators* (50ms) and *syntax operators* (0ms).

# Search Technique: Genetic Programming

Genetic Programming (GP) (Koza, 1992) is an evolutionary search technique which works by creating a population of candidate solutions and then gradually evolving this population through several iterations until a termination condition, such as the number of iterations, has been met. The evolution step is loosely based on biological evolution, with candidate solutions selected based on their *fitness*. New candidate solutions are created from existing candidates through the processes of *crossover* and *mutation*, which respectively combine or modify existing solutions.

**Fitness Function** The fitness function is used to rank different candidate solutions when choosing which candidates should be combined or used when the GP process constructs the next population. The fitness function used here is constructed from three components: *accuracy, response time* and *program size*. Accuracy is the overall performance of the model, based on the proportion of input-output pairs that it gets correct: accuracy is assessed in the range [0,1]. Response time is measured in simulated milliseconds, and program size is the number of operators in the control program.

As described above, accuracy is compared with the performance of human subjects: the closer the value of accuracy is to 0.957, the better it is. Similarly, the closer the value of the response time is to the target average of 767ms, the better. For response time, because the values can become large, we use a half-sigmoid function to rescale the numbers into the range [0, 1]. Program size is treated like response time, with an arbitrary target of 10 operators. All three components are evaluated so that values closer to 0 indicate a 'better' fitness. Formally, the three components of the fitness function are:

1.  $f_a = |\operatorname{accuracy} - 0.957|/0.957$ : this is the difference of the model's and target accuracy, scaled to the range [0, 1].

Name	Function	Туре
input-X	sets model 'current' to value of left/right/target input, if it is visible	input
respond-X	sets model 'response' to "R"/"L", if inputs are visible	output
access-N	sets model 'current' to STM item N (N $\in \{1, 2, 3\}$ )	stm
compare-M-N	compares value of STM items M and N (M $\neq$ N $\in$ {1,2,3})	cognitive
	and sets 'current' to 1 if equal, or 0 if not	
nil	sets model 'current' to 0	cognitive
put-stm	pushes value in model 'current' to top of STM	stm
dotimes-N	repeats a given expression (N $\in \{2, 3, 5\}$ )	syntax
if	executes condition, executes one of two expressions	syntax
	based on value in model 'current'	
prog-N	sequence of expressions (N $\in \{2,3,4\}$ )	syntax
wait-N	advances model clock (N $\in \{25, 50, 100, 200, 1000, 1500\}$ )	syntax

Table 1: Overview of operators used in DMTS models.

- 2.  $f_t = \text{half-sigmoid}(|\text{response-time} 767|/\text{RT})$ : this is the difference of the model's and target response time, with a variable scale factor *RT*.
- 3.  $f_s = \text{half-sigmoid}(|\text{program-size} 10|/\text{PS})$ : this is the difference of the model's and an arbitrary target program size of 10, with a variable scale factor *PS*.

where half-sigmoid(x) = 2 × (1/(1 +  $e^{-x}$ ) - 0.5) is the usual sigmoid function which we rescale from [0.5,1] to [0,1], because all our values of x are positive. The variable scale factors in  $f_t$  and  $f_s$  control the steepness of the sigmoid slope.

The *overall fitness* is computed as a combination of these three, with multipliers a + b + c = 1 ensuring that the overall fitness is in the range [0, 1]:

$$f = a \times f_a + b \times f_t + c \times f_s$$

**Phased Evolution** In earlier experiments, GP struggled to find solutions using this overall fitness function. The difficulty appeared to be that the requirement to minimise program size or meet a target response time would override the need to observe and predict a correct response. Hence, the idea of what we call *phased evolution* was created, to break this multi-component problem into stages. The evolutionary process is separated into three phases based on which of the three components are used in the fitness function: phase 1 uses one component ( $f_a$ ), phase 2 uses two components ( $f_a$  and  $f_t$ ), and phase 3 uses all three. The system starts in phase 1. It moves to the next phase when the best model's fitness is less than a threshold value (0.1 here).

More precisely, in:

**Phase 1** fitness  $f = f_a$ 

**Phase 2** fitness  $f = (a \times f_a + b \times f_t)/(a+b)$ 

**Phase 3** fitness 
$$f = a \times f_a + b \times f_t + c \times f_s$$

The intention of this phased introduction of fitness components is that the GP system should first evolve models to perform accurately on the task, when compared to the target behaviour. Once models have been created which meet the required threshold (f < 0.1), then they must additionally match the required reaction time. When the final component is added in, the GP system should already have a population of models able to achieve good accuracy and response times, and can now concentrate on reducing the size of the models.

### **Post-Processing**

Genetic programming (GP) is highly effective at locating candidate programs which fit target behaviour in complex applications. However, the range of interesting solutions is obscured by the large number of evolved candidates, formed from a combination of dead code (bloat), functionally similar program segments with varying contents, and genuine differences in possible solutions. In order to make the candidate programs more understandable, we introduce a series of postprocessing steps to generate fewer, high-quality solutions.

### **Dead code removal**

A standard problem with GP systems is that of "bloat" (Langdon & Poli, 1998): an example of bloat is where programs contain operators which are not executed when the task is run. This *dead code* can occupy the majority of a program, frequently over 90%. One way to remove dead code is to add the program size as one of the components in the fitness function. However, as we find in our experiments, this is not completely effective.

A more effective way to remove dead code starts by tracing the operation of each evolved program on our task and recording those parts of the program which are not executed: it is important that our task is *deterministic* so this can be done reliably. All non-executed code is then replaced with the special node "UNUSED". Conveniently, all non-executed code must be on one branch of an IF-statement: the code is not run because the condition on the IF-statement always returns a value which uses just one branch of the IF-statement. For example, if some CONDITION always returns a true value, its else branch will never be executed:

#### (IF (CONDITION) (SOME-CODE) (UNUSED))

The programs can be simplified by replacing all such code to remove the UNUSED branch:

```
(PROG2 (CONDITION) (SOME-CODE))
```

The condition must still be executed as it could contain side-effects and takes up some execution time, which is critical for the timing performance of the model.

This step helps in two ways:

- 1. The population of candidate models is reduced dramatically, by removing those which differ only in the contents of the dead code.
- 2. Each individual model is simplified, with only important parts of its control program remaining.

### **Time-only code removal**

There is a further aspect of the candidate models which can be simplified. As the model is optimised to perform against time, some of the operators within the control programs can be important only for their timing – they do not affect the performance. For example:

(PROG2 (INPUT-LEFT) (INPUT-RIGHT))

In this program, the model first looks at the left input, and then looks at the right input. The second operation will *always* override the behaviour of the first operation, and hence the first operation only affects the model's clock and not its accuracy. Other operators could be used in place of INPUT-LEFT to take up a similar amount of time, e.g. INPUT-RIGHT, but these would, superficially, look like different models. However, by replacing each operator with a specific WAIT operator we get the same timing and performance behaviour but with a clearer model. i.e. the previous example is replaced with:

(PROG2 (WAIT-INPUT) (INPUT-RIGHT))

This step has two advantages:

- 1. The programs of the candidate models are made clearer, with all time-only operations written as WAIT- operators. This improves the comprehensibility of the final model.
- Behaviourally similar models have syntactically similar control programs. This means more redundant models can be removed from the candidate models.

#### **Pair-wise similarity**

Clustering and visualisation can be helpful to understand the models' programs as a group. We introduce a pair-wise similarity measure between programs to make this possible. Each program is separated into a set of node+child-labels segments. For example, the following program is converted into eight segments of two parts and six individual node names:

```
(if (access-1)
  (prog2 (input-right) (input-left))
  (input-target))
parts: (if access-1 prog2 input-target)
      (prog2 input-right input-left)
names: if access-1 prog2 input-target
      input-right input-left
```

The pair-wise similarity (Jaccard Index) divides the number of common segments in the two programs (the set intersection) by the total number of segments (the set union).

#### Pseudo-code

As shown in the preceding examples, individual models are represented internally as abstract-syntax trees: we can rewrite each model in a more readable pseudo-code. Although not fully automated, this step also combines consecutive WAIT operators, further simplifying the models. An example is shown in Figure 4.

### **Simulation Experiments**

Table 2 shows a typical set of results, where we have varied the hyperparameters a, b, c and RT, with each run using a population of 500 individuals and 2000 generations. Recorded are the generation and performance measures for the best models found in each run. Most of the runs produced "good" models, with excellent fits to both accuracy and response time. However, due to the stochastic nature of the search algorithm, the last two runs failed to converge: over 5 repeats of the 6 shown sets of parameters, 5 runs failed to converge to a model with good accuracy, and a further 9 runs failed to converge to a good model of response time.

# **Phases in evolution**

Table 3 gives summary statistics on which generation each phase was reached. In some cases phases 2 and 3 were reached very quickly, in less than 100 generations.

By analysing results against generation, we can investigate how the phases affect or reflect changes in the fitness function. Figure 2 shows overall fitness,  $f_a$ ,  $f_t$  and  $f_s$  against generation number for the best model in each generation, for the first 100 generations.

Phase 1 of evolution lasts only up to generation 10, where the accuracy is optimised (the red line). As the accuracy improves, it improves the overall fitness (the green line) beyond the threshold of 0.1, and phase 2 begins.

Phase 2 lasts from generation 10 to 60, and optimises both accuracy and response time (the blue line). Around generation 50 the response-time accuracy starts to improve, as does the overall fitness. As the threshold of 0.1 is crossed by the best model, phase 3 begins. Notice how the program size appears to grow from generations 20 to 50 before the response time can begin to improve. Due to the phased introduction of the components, this increase in program size does not affect the fitness.

2	h	C	DC	Generation	Fitness (f)	Accuracy	Response Time	Program Size
a	U	C	13	Ocheration	1 micss (1)	Accuracy	Response Time	1 logram Size
0.80	0.1	0.10	100	202	0.040	1.00	775.0	17
0.85	0.1	0.05	100	376	0.040	1.00	770.0	26
0.89	0.1	0.01	100	491	0.040	1.00	830.0	18
0.80	0.1	0.10	500	176	0.040	1.00	760.0	17
0.85	0.1	0.05	500	78	0.140	0.92	8695.0	53
0.89	0.1	0.01	500	56	0.140	1.00	8595.0	72

Table 2: Table of results from 'phased' evolution simulation (PS = program size parameter).

Table 3: Generation when phase reached (out of 30 runs).

Phase	Frequency	Minimum	Maximum	Average
2	25	6	271	79.28
3	16	55	1529	294.56



Figure 2: Progress of fitness against generation for the best model. Only the first 100 generations out of 2000 are shown.

Phase 3 lasts from generation 60 to the end. As is evident in Figure 2, there are still some gains to be made in the response-time, which falls to an almost negligible error by generation 90, and, en passant, halves the overall fitness. The main change from this point is a steady reduction in program size: when phase 3 starts, at generation 60, the best model has 46 nodes, whereas by generation 2000 the best model only has 24 nodes, almost halving its complexity.

# Effects of post-processing

Combining the candidate models from each of the six runs means the GP system produces 1164 distinct models with a good fitness value (less than 0.1). This set of models is too large to analyse and understand. In particular, the programs are obscured with bloat (only 40% of the population has less than 10% dead code) and the intention of different parts of the solution (to solve the accuracy or the reaction-time) is hidden.

Our two post-processing techniques reduce this number dramatically: removing the dead-code leaves 248 distinct models, and further removing the time-only operators reduces these to 11 distinct models.



Figure 3: Visualisation of model diversity: Model distance is inverse of similarity.

### Visualisation of models

Figure 3 depicts model diversity in a graphical form, using multi-dimensional scaling to convert pair-wise similarity into cartesian coordinates. What is most striking about this image is that the models have split into three distinct groups. The top model is an outlier, there are three models in the left-hand group, and the remaining seven models are in the right-hand group. Figure 4 shows an example from the left-hand group.

Analysing the pseudo-code of these models helps to understand the two types of solution. One (shown in Figure 4) uses a fixed delay between reading the target and the input: initially the model reads the target, then places this into STM. The model then uses a loop to wait the required time before it can see the input, followed by some processing to set up the appropriate response. The second type uses a more general perceptual loop, which tries to first read the target and then the input stimulus in turn. Because of how the environment timings work, the input stimuli will only be available in a later loop of the program and so the model will arrange the target and input in its STM as required to complete the task.

The remaining models fit these patterns, mostly with negligible differences in the ordering of operations and whether the model looks at the left or right stimulus. The outlier model is a variation on those of the second kind, but uses one outer loop repeated multiple times, rather than having a long delay within the outer loop, as in the second kind of model.

A concern when confronted with these multiple

```
if target is visible:
  set model 'current' to target
wait for 140ms
push model 'current' onto top of STM
loop 3 times:
  loop 5 times:
    if stimuli are visible:
      set model 'current' to left input
if stimuli are visible:
  set model 'response' to "R"
push model 'current' onto top of STM
if first item in STM equals second item:
  set model 'current' to 1
else:
  set model 'current' to 0
if model 'current' is 1:
  if stimuli are visible:
    set model 'response' to "L"
else:
  wait for 70ms
wait for 70ms
```

Figure 4: Example Program: Pseudo-Code

automatically-generated models is whether they are explainable or qualitatively match human behaviour. This is a topic we intend to address with improved heuristics and constraints in the GP system. However, we do not see the system as standing in isolation, but as a tool to aid the cognitive scientist. The system generates a range of *candidate* models, and the cognitive scientist using the system has the responsibility to select from or modify the generated models to create a final model and/or theory.

### Discussion

A weakness of our approach is that the empirical data are the result of averaging across several individuals (e.g. Gobet, 2017; Gobet & Ritter, 2000; Siegler, 1987): one model represents that average individual. One way to simulate group behaviour is to modify GP to manage several programs instead of just one; each program would represent a single person, and the average performance of these programs would be compared to the given average.

However, more recently, psychologists have begun to publish more of their empirical data, including the performance of individual subjects. The analysis process developed in this study can use runs capturing not just one but multiple subjects, and combine the candidate solutions to see how similar or different the behaviour of different individuals is. In particular, as of now, the 'preferred' type of data are choice and reaction times, which are extremely popular outputs in fields such as decision making or psychophysics, and will be areas where the approach in this paper should be beneficial.

Further areas for future work include a co-evolution approach, to optimise the operator time parameters, and

domain-specific heuristics for the GP algorithm.

### Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. ERC-ADG-835002).

### References

- Anderson, J. R., & Lebière, C. (Eds.). (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum.
- Chao, L., Haxby, J., & Martin, A. (1999). Attribute-based neural substrates in temporal cortex for perceiving and knowing about objects. *Nature Neuroscience*, 2, 913–20.
- Frias-Martinez, E., & Gobet, F. (2007). Automatic generation of cognitive theories using genetic programming. *Minds* and Machines, 17, 287–309.
- Gobet, F. (2017). Allen Newell's program of research: The video game test. *Topics in Cognitive Science*, 9, 522–532.
- Gobet, F., & Ritter, F. E. (2000). Individual data analysis and Unified Theories of Cognition: A methodological proposal. In N. Taatgen & J. Aasman (Eds.), *Proceedings* of the Third International Conference on Cognitive Modelling (pp. 150–57). Veenendaal, The Netherlands: Universal Press.
- Gobet, F., & Simon, H. A. (2000). Five seconds or sixty? Presentation time in expert memory. *Cognitive Science*, 24, 651–82.
- Gulwani, S. (2010). Dimensions in program synthesis. In Proceedings of the 12th international ACM SIGPLAN symposium on principles and practice of declarative programming (p. 13-24).
- Koza, J. R. (1992). Genetic programming: On the programming of computers by means of natural selection. Cambridge, MA: MIT Press.
- Lane, P. C. R., Sozou, P. D., Gobet, F., & Addis, M. (2016). Analysing psychological data by evolving computational models. In A. Wilhelm & H. Kestler (Eds.), Analysis of large and complex data. Studies in classification, data analysis, and knowledge organization (pp. 587–97). Springer, Cham. doi: 10.1007/978-3-319-25226-1\_50
- Langdon, W. B., & Poli, R. (1998). Genetic programming bloat with dynamic fitness. In W. Banzhaf, R. Poli, M. Schoenauer, & T. C. Fogarty (Eds.), *European conference on genetic programming: Eurogp* (pp. 97–112).
- Langley, P., Simon, H. A., Bradshaw, G., & Zytkow, J. (1987). *Scientific discovery: Computational explorations of the creative processes*. Cambridge, MA: MIT Press.
- 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, 1209 - 1214.
- Rumelhart, D. E., & McClelland, J. L. (Eds.). (1986). *Parallel distributed processing* (Vol. 1 and 2). Cambridge, MA: MIT Press.

Proceedings of the 20th International Conference on Cognitive Modelling (ICCM 2022)

- Schmidt, M., & Lipson, H. (2009). Distilling free-form natural laws from experimental data. *Science*, *324*.
- Siegler, R. S. (1987). The perils of averaging data over strategies: An example from children's addition. *Journal of Experimental Psychology: General*, 250–264.
- Simon, H. A. (1981). Information-processing models of cognition. *Annual Review of Psychology*, *30*, 363-96.

# Towards a Computational Model of a Dynamic Feeling of Knowing

Justin Li (justinnhli@oxy.edu) Bryce Boyle (bboyle@oxy.edu) Occidental College, 1600 Campus Road Los Angeles, CA 90041 USA

#### Abstract

Feelings of knowing (FOKs) are metamemory judgments that suggest an answer could be retrieved from memory with more effort. This paper reviews the psychological accounts of FOKs and maps them onto sources from the memory mechanisms of the common model of cognition. Two widely accepted accounts of FOK, that of cue familiarity and accessibility, map directly onto properties of the retrieval cue and result respectively. In considering these models of FOK, we identify an omission from the literature: the possibility that FOK changes over time while answering a question. We discuss the implications of this dynamic account and conclude with the difficulties of evaluating computational models of FOK.

**Keywords:** Feeling of Knowing; Metamemory; Knowledge Search; Common Model; Cognitive Architecture

#### Introduction

Feelings of knowing (FOKs) are a memory phenomenon where, despite not retrieving the answer to a question in the moment, people *feel* that they will be able to do so with more effort. FOKs have been studied as a topic of its own and as a way to gain insight into how memory is used in decision making (Nelson & Narens, 1994), with research focusing on the sources that underlie FOK and how it is influenced by the retrieval context. More recently, researchers have proposed the cognitive-heuristic account of metamemory: that FOKs serve the function of guiding memory retrieval, allowing for early failure without expending resources if the probability of finding an answer is low (Schwartz & Metcalfe, 2011). At the same time, although cognitive architectures such as ACT-R (Anderson, 2007) grew out of early models of declarative memory, metamemory phenomena has not received much attention and has not been the topic of cognitive modeling.

This paper complements the existing literature by exploring how feelings of knowing might be instantiated in a common model of cognitive architecture. We begin by placing FOKs in the context of knowledge search, and in doing so identify an omission from our current understanding of the phenomenon, that of how FOKs might change during question answering. We then briefly summarize a computational representation of memory, before committing the bulk of this paper to considering potential sources of FOK in the common model, as described by existing psychological theories. Returning to a dynamic FOK, we explore the consequences of such a theory, and conclude by discussing the obstacles that exist for evaluating a computational model of FOK.

# FOK and Strategic Knowledge Search

In order to understand FOKs, its function within the memory systems of an agent must be understood: that of helping an agent retrieve knowledge and engage in knowledge search. First proposed by Newell and Simon (1972), knowledge search is the process of finding knowledge that is relevant and could be applied to the current problem solving context. Newell and Simon do not elaborate on the processes of knowledge search, and knowledge search has received scant attention as compared to problem space search. Instead, the main advances come from psychological research on how people use memory in naturalistic settings, especially on how problem solving and decision making are intertwined with memory. The results show a rich landscape of memory processes: beyond basic recognition and recall, participants described determining recall specifications, gauging their own familiarity with the topic, relating multiple relevant memories, and verifying that a potential answer is in fact correct (Burgess & Shallice, 1996). All of these processes, together with actual memory retrievals, are necessary to answer one question. This account validates the idea that metamemory judgments such as FOK are used for the control of retrieval processes, such as selecting a search strategy and deciding whether to terminate search (Nelson & Narens, 1990). More recently, FOK has been shown experimentally to influence search termination and decision of what to rehearse (Singer & Tiede, 2008; Hanczakowski et al., 2014), further corroborating the cognitive-heuristic account of metamemory.

For example, consider a question such as What film was nominated for seven Academy Awards in 1999?<sup>1</sup> (Norman et al., 2016). To answer the question, an agent might retrieve cultural events in 1999, famous directors and actors/actresses, generally acclaimed films, and so on. Some of these results will be useful for answering the question; others may turn out to be irrelevant or lead to dead ends. In between these retrievals, FOKs play the role of determining the search strategy or whether to terminate search. Here, however, there is a mismatch between this hypothesized role of FOKs and how we conceptualize it for experiments. Thus far, experimental procedures for FOKs tend to only solicit a single judgment, either before or after the participants attempt to answer the question. If we accept that FOK is used to guide the multiple retrievals needed to find an answer, it raises the question: at which retrieval was the FOK solicited, and to which retrieval was the FOK indicating that an answer exists? When a participant reports their feeling of knowing, is it to the original question, or to any of the sub-questions that they ask themselves as they engage in the strategic search for the answer?

<sup>&</sup>lt;sup>1</sup>Answer: *Life is Beautiful* 

Here we propose that the reported FOK is to the original question, and not to any of the other retrievals during the search process. This interpretation is more obvious for a question such as What is the capital of Australia?<sup>2</sup>. Most people will suggest answers such as Sydney, Melbourne, and Brisbane before giving up. Despite these successful retrievals for Australian cities, participants will report that their FOK goes down over time before they terminate their search. That is, it seems clear to us that FOK is a dynamic signal that changes throughout the strategic memory search process: as additional retrievals are used for problem solving, the FOK fluctuates for the overarching goal of answering the original question. This is also consistent with the cognitive-heuristic account of FOK: in order for FOK to be a reliable signal for search termination, it *must* change over the course of the process to reflect whether an answer is still likely to be found. Again, this stands in contrast to how FOK is usually studied: all psychological experiments we have found only solicit participants' FOKs once, either before or after they are given the chance to attempt to answer the question. While we have no doubt that such reports of FOK will still be correlated with the state of memory, ignoring the time course of FOK will likely omit crucial aspects of how the signal is determined. For the remainder of this paper, we will therefore assume this dynamic view of FOK as we consider how it might be modeled computationally.

# Memory in the Common Model of Cognition

We now describe the agent framework in which we would like to model FOK, namely, that of the common model of cognition (Laird et al., 2017). The common model defines a set of representations and processes for modeling cognition, as implemented in cognitive architectures such as ACT-R and Soar (Anderson, 2007; Laird, 2012). Of particular interest to this paper are the declarative long-term memory (LTM) processes, specifically that of semantic memory, which we describe below.

Formally, the contents of LTM is an edge-labeled directed graph, defined by the tuple  $\langle S, P, L, E \rangle$ : *S* the set of entities or concepts (we use these terms interchangeably), which corresponds to the internal nodes of the graph; *P* the set of predicates, which corresponds to the edge labels of the graph; *L* the set of literals, such as numbers and strings, which corresponds to the leaf nodes of the graph; and *E* the set of direct edges from one entity to another entity or to a literal,  $\langle s, p, o \rangle \subset S \times P \times O$ , with  $O = S \cup L$ . Borrowing from the knowledge representation literature, we will also refer to edges as *triples*, and refer to the elements of a triple  $\langle s, p, o \rangle$  as the *subject*, the *predicate*, and the *object* respectively.

An agent has two ways of getting knowledge from LTM. First, for any entity *s*, the agent can *retrieve* all outgoing edges  $\{\langle s, p, o \rangle \in E\}$  for which that entity is the subject. This mechanism is for accessing related information of a known concept, but to find an unknown concept that has certain Within this framework, we can define the general process through which an FOK might be generated. When the agent is presented with a question, the agent would execute a sequence of queries and retrievals to LTM to attempt to answer the question. For clarity, we call the answer to the overarching question the *answer*, while an individual query will have a *result* (the entity that is returned) out of a set of *candidates* (other entities that match the cue). We assume that the FOK for the original question will change with each query and retrieval, and we are therefore interested in the computational processes that occur at those times and how they might affect the overall FOK.

A quick note on terminology: the term *retrieval* is overloaded in both psychology and cognitive architecture literature to sometimes mean both queries (with a cue) and retrievals (of a known concept in LTM). Retrieval will be used in the psychological sense in this paper; we will disambiguate the term as needed when talking about the specific computational mechanism.

# **Psychological Accounts of FOK**

This section explores how psychological accounts of FOK might be realized within the common model. Within the psychology literature, there are three main accounts of FOK: cue familiarity, accessibility, and competition. For each, we first discuss the relevant psychological literature, before exploring how it may be translated computationally into long-term memory mechanisms. Since the literature primarily assumes a static FOK for a question, instead of one that changes over time, these computational models are all calculated from a single retrieval. A summary of these sources of information for FOK can be found in Table 1.

Two mathematical caveats must be considered. First, FOK may be a function of multiple parameters. Since we are primarily interested in what those parameters might be, and less interested in how they might be combined into a single FOK, we will assume that the function is a monotonic summary statistic denoted as f(). Although the choice of summary statistic may affect the FOK calculation — the mean will be more sensitive to outliers than the median, for example — we consider this detail too low level for this paper. We do note that the competition account seems to be better modeled as the variance of a distribution than the mean or median, and

properties, the agent must *query* LTM instead. To do so, the agent creates a query *cue*  $Q = \{q \in P \times O\}$ , which describes the predicates and corresponding objects of the desired entity *s* such that  $\forall \langle p, o \rangle \in Q$ ,  $\langle s, p, o \rangle \in E$ . We designate all matching entities of a query Q as  $S_Q$ , the set of retrieval *candidates*. If more than one such retrieval candidate exists, the entities with higher *base-level activation* are preferentially returned. Base-level activation is determined by  $A(s) = \ln(\sum t_i^{-d})$ , where  $t_i$  is the time since the entity *s* was last retrieved, and *d* is a decay rate parameter. Activation thus captures the recency and frequency of use of a concept, and is often used as a proxy of the importance of the concept to the agent at a particular time.

<sup>&</sup>lt;sup>2</sup>Answer: Canberra

	Cue	Candidates	Result
Count		Accessibility	
1/Count		Competition	
Activation	Cue Familiarity	Competition	Accessibility
Connectivity	Cue Familiarity	Competition	Accessibility

Table 1: Difference sources of information for calculating FOK, and which corresponding psychology FOK theory they belong in. Gray cells represent sources that exist but not meaningful for FOK (e.g., the number of retrieval results, which is constant).

it is an open question whether or how other properties of the distribution might contribute to FOKs. Second, the domain of the output of the FOK function is unclear. The main constraint is that the agent should be able to determine whether an FOK judgment is high or low and thereby make retrieval decisions. The output could theoretically range over the real numbers — such as if FOK was the activation of a concept — with the agent learning decision thresholds over time. As we consider a dynamic FOK that may shift between difference sources of information, however, normalizing the FOK may be necessary, as the domain of the sources main differ wildly. For each account of FOK below, we will therefore also consider the population against which an FOK might be normalized.

### The Cue Familiarity Account of FOK

As the name implies, the cue familiarity account of FOK focuses on the contribution of the retrieval cue to the feeling of knowing (Reder & Ritter, 1992; Metcalfe et al., 1993; Koriat & Levy-Sadot, 2001). For the purpose of this paper, we include all FOK sources that are based on the cue, including familiarity and domain knowledge (Schwartz, 1994). The intuition is that FOK is a summary of the amount of knowledge the agent might have about a topic, as estimated from the terms of the question. The more the agent is familiar or knowledgeable about the topic, the more likely that they will know the answer, leading to a higher FOK. Computationally, an FOK based on cue familiarity must be a function of the cue  $Q = \{ \langle p, o \rangle \in P \times O \}$ . In general, FOKs based on the cue familiarity account may be normalized against all concepts in LTM, as it would indicate the agent's familiarity with these cues in particular, although care must be taken to account for cues that do not exist in LTM. We consider two metrics that might signify that the agent is "familiar" with the entities in the cue: their activations and their connectivities.

Activation One possible metric for the familiarity of the cue is the activation of each individual concept in the cue. Since activation reflects how recently and frequently a concept has been encountered, concepts with a high activation would be ones that are presented often, which in turn suggests that the agent would be familiar with them. Formally, this metric for FOK could be defined as:

$$FOK = FOK(Q) = f(A(o_1), ..., A(o_n)) \ \forall \langle p, o \rangle \in Q$$

**Connectivity** In contrast to activation, connectivity captures how much knowledge the agent has of each concept in the

cue. A concept in which an agent is knowledgeable would be connected to many other concepts, while a concept of which the agent is ignorant would only be sparsely connected. In the extreme, the simple presence or absence of the concept (i.e., whether the agent recognizes the concept) may be a sufficient signal to terminate search, and it has been shown that recognition is can be a useful heuristic for knowledge search (Li et al., 2012).

The connectivity of a concept is measured by its *fan*, the number of incoming (*fan-in*) and outgoing (*fan-out*) edges. Arguments could be made for only considering fan-in or fan-out. The fan-in would represent the prevalence of the concept in different contexts, while the fan-out might represent its generality. It is also possible to consider the overall fan of a concept, regardless of the direction of the edges. More generally, connectivity may not just be the immediate neighbors of the cue, but the number of concepts within some neighborhood. We leave these implementation details as future work, and leave the precise meaning of the fan( $s \in S$ ) function undefined. Formally, this metric for FOK could be defined as:

$$FOK = FOK(Q) = f(fan(o_1), ..., fan(o_n)) \ \forall \langle p, o \rangle \in Q$$

# The Accessibility Account of FOK

Unlike the cue familiarity account that depends on the cue, the accessibility account of FOK considers information that is only available during or after a retrieval, using the "byproducts" of the retrieval process (Koriat, 1993). The intuition behind the accessibility account is that the retrieval process may provide hints as to whether the agent could answer the question; if the first retrieval leads to a result with high confidence and certainty, this may lead to a high FOK even if additional retrievals are still necessary. Although the accessibility account includes uses of both properties of the result and metadata from the retrieval process, in practice the common model does not define universal a set of retrieval metadata that could be accessed. As a result, the models of FOK presented below are all functions of the retrieval result or the candidates.

While accessibility FOKs could also be normalized against other entities in LTM, a different reference group is also available: the set of candidates that matches the retrieval cue. This may reveal the relative importance of this result against other possible results. Such a comparison group would blur the difference between the accessibility account with the competition account, which we discuss in the next section. Activation As with the activation of the cue, the activation of the result of a retrieval may be a metric for an accessibilitybased FOK. Beyond summarizing the recency and frequency of use and therefore whether a concept is familiar, activation in this context may also represent the speed of the retrieval: the higher the activation, the more quickly the retrieval occurs. There is a large literature on the correlation between fluency and various memory phenomena (Alter & Oppenheimer, 2009), but here we consider it as equivalent to the activation of the retrieved result under the common model.

Formally, this metric for FOK could be defined as:

$$FOK = FOK(s) = f(A(s))$$

where s is the result of a retrieval.

**Connectivity** A different metric from the result of a retrieval is its connectivity, or the number of graph neighbors it has. As before, the connectivity of an entity represents the amount of knowledge that the agent has about the result. We note that the activation and connectivity of a retrieval result may be related due to spreading activation, but could also be the inverse of each other. A concept could be well-understood (i.e., have high connectivity) but irrelevant to the recent/current context (i.e., have low activation), as is the case when false memories are induced (Li & Kohanyi, 2016). Conversely, a concept could have low connectivity but high activation, such as when an agent is learning about a new concept.

Formally, this metric for FOK could be defined as:

$$FOK = FOK(s) = f(fan(s))$$

where *s* is the result of a retrieval.

**Retrieval Candidates** The accessibility account has an additional possible metric compared to cue familiarity: the number of candidates in the retrieval. The intuition for this metric is that if a retrieval cue matches many concepts, the agent might conclude that it has a lot of information at hand about the question, thus increasing the likelihood that it will be able to find the answer. Mathematically, this metric for FOK could be defined as:

$$FOK = FOK(s) = f(|S_Q|)$$

**Other Accessibility Sources** Other metadata of the retrieval process and the results have been proposed as FOK sources, although they do not map as cleanly onto the existing memory mechanisms of the common model. One such possibility is for FOK to be based on a *partial retrieval*, where some but not all information is retrieved (Hanczakowski et al., 2017). The intuition is that a partial retrieval suggests to the agent that a complete retrieval is possible, thus leading to an FOK. While this theory is psychologically plausible, we do not know of any common model cognitive architectures that support partial retrievals, leaving a model of such an FOK for future work.

Similarly, incorrect retrievals about the answer may contribute to FOK (Koriat, 1993). This source, however, may be difficult to model comptuationally, as the agent has no a priori knowledge of whether a result is correct or not. The idea of incorrect retrievals as a source of FOK is further complicated by the idea that multiple retrievals are necessary to answer a question, as the majority of these intermediate results will not be the answer to the original question. On the other hand, this more complex landscape of knowledge search also presents opportunities. If "incorrect retrieval" is interpreted as the agent encountering difficulties, the need to change search strategies may itself decrease FOK, as it may suggest that the question is more difficult than assumed. More generally, it is not impossible for an FOK judgment to take other metacognitive phenomena into account. A thorough exploration of how FOK might relate to other metamemory is beyond the scope of this paper.

Finally, the semantic content of the result itself may be a source of FOK, allowing the agent to infer additional knowledge that boosts FOK. The inference process is highly dependent on the question and the existing knowledge and capabilities of the agent, however, and given the large space of possibilities, we leave the relationship between semantic content and FOK for future work.

#### The Competition Account of FOK

Less commonly discussed than the cue familiarity and accessibility accounts of FOK is the competition account (Schreiber & Nelson, 1998). Unlike the accessibility account where a large number of candidates suggest robust knowledge, the competition account states that FOK is *inversely* proportional to the number of candidates. While the competition account of FOK may technically be a subcategory of the accessibility account, in that the number of candidates is a piece of metadata from the retrieval process, we consider the competition account sufficiently different to address it separately. In particular, unlike the accessibility account where the result of a retrieval plays a main role, the competition account (in the extreme) does not consider the result at all. Rather, only the set of candidates influence FOK; which specific concept is retrieved is irrelevant.

The intuition for the competition account is that more potential results to a query increases the uncertainty as to which result is correct, thus decreasing FOK. A direct translation of the competition account is to use the inverse of the number of candidates, which could be defined as:

$$\text{FOK} = \text{FOK}(Q) = f(\frac{1}{|S_Q|})$$

However, other metrics for the competition account is possible. Extending the idea of uncertainty caused by having many candidates, we could model competition using the distribution of the activation or connectivity of the candidates. A uniform distribution would indicate that no candidate is more likely than the other, suggesting uncertainty; in contrast, a peaked distribution would mean that the candidate with more probability mass is likely to be the correct answer. An activation-based competition metric for FOK could be defined as:

$$FOK = FOK(Q) = f(Var(\{A(s) \forall s \in S_Q\}))$$

such that the larger the variance in activation, the larger the difference between the most activated concept and other concept, and therefore the more certain that it is the answer. As with other accounts, variance could be replaced with other summary statistics such as the interquartile range, as long as it correlated variance and inversely correlated with the uniformity of the activation values.

# **Hybrid Accounts**

Although we have considered activation and connectivity as separate sources of FOK, cue familiarity and accessibility accounts of FOK could incorporate both sources of information. For example, FOK could be calculated by averaging the activation of neighboring concepts, resulting in an FOK that takes both activation and connectivity into account, combining more information from the agent's knowledge base. Such a calculation is reminiscent of spreading activation, which bolsters its psychological plausibility. A systematic exploration of FOK metrics that combine sources, and their psychological plausibility, is beyond the scope of this paper.

Mixing and matching FOK accounts may apply at the higher level as well. While the cue familiarity and accessibility accounts each only take one type of memory metadata as input, in practice FOK may be the result of more complex combinations of these sources that together lead to the FOK that people report. This idea is not new, as it has been noted that cue familiarity is available after the question is asked but before a retrieval, while accessibility is only available during or after a retrieval. It has therefore been suggested that these could be used sequentially: that FOKs solicited earlier are a result of cue familiarity, and FOKs solicited later are a result of accessibility (Florer & Allen, 2000; Koriat & Levy-Sadot, 2001). These multiprocess theories hint at how the two accounts are not as independent as previously suggested. It is a small step from there to our proposed dynamic account of FOK, which we now turn our attention to.

# **A Dynamic FOK Account**

As mentioned earlier in this paper, the psychology literature has focused on FOK as a single measurement during the process of question answering. We now return to the idea that FOK may instead be dynamic, changing over time as different strategies and multiple retrievals are used to answer a single question. The hybrid account of FOK hints at this possibility, by suggesting that FOK uses different sources depending on when it is solicited. One issue with this account, however, is the assumption that FOK is constant throughout answering a question, which would require different sources to somehow lead to the same resulting FOK. Instead, we reframe FOK as a judgment that is always changing while answering a question: different sources of FOK are used but do not have to agree with each other, and these fluctuations may in fact be part of how FOK guides knowledge search. This hypothesis makes the our understanding of FOK more parsimonious, as the previously retrieved results (used by the accessibility account) are then used as cues for the next retrieval (used by the cue familiarity account), thus unifying the different accounts. This section considers the ramifications of this hypothesis, and proposes additional possibilities for how FOK may be determined.

First, we note that while results from past experiments are likely not invalid, they may only provide a narrow view of FOK. These measurements may only be accurate to the state of knowledge search at the time of solicitation, and without a detailed understanding of the memory search state of the participant, it is difficult to infer how the FOK was generated. Even assuming the cue familiarity or accessibility accounts, it raises questions as to what cues were used for familiarity judgments, or what retrieval metadata were used when accessibility was measured. The possibility of multiple retrievals that occur in sequence also muddle the distinction between retrieval cues and retrieval results, since the result of one retrieval may become the cue for the next retrieval. New experimental paradigms will need to be created to determine how FOK changes over time, before existing empirical can be integrated.

A dynamic FOK has implications not just for which sources are used (if they are indeed different sources at all), but what information each source provides. During the course of problem solving, the activation of entities will change based on the results of previous retrievals. A cue that initially had low activation may be boosted if multiple retrieval results are connected to it; conversely, previously highly activated entities may become less so over time. While the connectivity of LTM is less affected by retrievals, it is also not impossible that new connections could be made during problem solving, for example if an agent realizes that blue whales are not fish in answering *What is the largest fish on earth*?<sup>3</sup> In sum, the sources do not only provide a single value, but a *history* of values which could be combined into an FOK judgment.

Access to a history of memory metadata raises the possibility that FOK could be based on previous FOK values, or at least some summary thereof. Consider again the question of what is the capital of Australia, and where an agent guess with several large Australian cities before giving up. This could be explained by the accessibility account using activation: more prominent cities such as Sydney are guessed first, before lesswell-known cities like Perth, until the activation drops below some threshold and the agent terminates the search. However, another model of FOK is possible: that the search termination is not just due to the activation of the last retrieved concept, but due to the overall downward trend of activation. In this case, the FOK judgments are not based purely on activation, but is additionally modulated by how the FOK itself has changed over time. Mathematically, we might define FOK to be a function of time,  $FOK_t$ , with t being the number of steps in the past. In this example, the fact that  $FOK_{-3} > FOK_{-2} > FOK_{-1}$ 

<sup>&</sup>lt;sup>3</sup>Answer: Whale sharks

would further decreases the FOK judgment. More generally, FOK could be defined as

$$FOK = f(FOK_{-1}, ..., FOK_{-T})$$

up to some time T in the past, plus additional inputs corresponding to the cue familiarity and accessibility accounts. Given the importance of history in this account, modeling FOK may therefore require a deep understanding of the landscape of memory processes and how they behave over time.

## **General Discussion**

This paper has explored the possibilities for modeling feelings of knowing within the common model of cognition. The three main accounts of FOK — cue familiarity, accessibility, and competition — map well onto the existing architectural memory mechanisms. At the same time, the assumption that FOK is constant breaks down when multiple retrievals from long-term memory are needed to find an answer. As a result, we proposed the possibility of a dynamic FOK that changes over time as retrievals are made, and also raise the possibility that FOKs could take history into consideration.

Defining the mathematical space of FOK is a step forward, but evaluating computational models will be difficult. Matching human data may be possible if we restrict the model to questions that can be answered by a single retrieval. Experiments such as those reported in Schwartz et al. (2014) and Florer and Allen (2000) manipulate FOK by varying the amount of artificial context, thus creating new connections in LTM and also inducing unequal activation among the new concepts. Matching data on more complex questions, however, will be complicated by the multi-step retrieval process and uncertainty around which retrieval the FOK should be computed. Alternately, we can also foresee evaluation FOK on artificial agents, by examining which accounts most accurately predicts whether the agent is able to eventually find an answer. The advantage of this approach is that it can be applied on top of existing models of memory, and it may provide insight into how FOK may change over time. In the long run, cognitive models of FOK will have to meet both of these evaluation criteria in order to accurately reflect its function as a heuristic for memory retrieval in people.

# References

- Alter, A. L., & Oppenheimer, D. M. (2009). Uniting the tribes of fluency to form a metacognitive nation. *Personality and Social Psychology Review*, 13(3).
- Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press.
- Burgess, P. W., & Shallice, T. (1996). Confabulation and the control of recollection. *Memory*, 4(4).
- Florer, F. L., & Allen, G. (2000). Feelings of knowing in the Ranschburg effect. *The American Journal of Psychology*, *113*(2).
- Hanczakowski, M., Zawadzka, K., & Cockcroft-McKay, C. (2014). Feeling of knowing and restudy choices. *Psych. Bullet. & Rev.*, 21(6).

- Hanczakowski, M., Zawadzka, K., Collie, H., & Macken, B. (2017, 01). Metamemory in a familiar place: The effects of environmental context on feeling of knowing. *J. of Exp. Psych.: Learning, Memory, and Cognition*, 43(1).
- Koriat, A. (1993). How do we know that we know? The accessibility model of the feeling of knowing. *Psych. Rev.*, *100*(4).
- Koriat, A., & Levy-Sadot, R. (2001). The combined contributions of the cue-familiarity and accessibility heuristics to feelings of knowing. *J. of Exp. Psych.: Learning, Memory, and Cognition*, 27(1).
- Laird, J. E. (2012). The Soar cognitive architecture. MIT.
- Laird, J. E., Lebiere, C., & Rosenbloom, P. S. (2017). A standard model of the mind. *AI Magazine*, *38*(4).
- Li, J., Derbinsky, N., & Laird, J. E. (2012). Functional interactions between memory and recognition judgments. In *Proceedings of the 26<sup>th</sup> AAAI conference*.
- Li, J., & Kohanyi, E. (2016). Towards modeling false memory with computational knowledge bases. In *Proceedings of the* 14<sup>th</sup> ICCM.
- Metcalfe, J., Schwartz, B. L., & Joaquim, S. G. (1993). The cue-familiarity heuristic in metacognition. *J. of Exp. Psych.: Learning, Memory, and Cognition, 19*(4).
- Nelson, T. O., & Narens, L. (1990). Metamemory: A theoretical framework and new findings. *Psychology of Learning and Motivation*, 26.
- Nelson, T. O., & Narens, L. (1994). Why investigate metacognition? In *Metacognition* (pp. 1–25). MIT.
- Newell, A., & Simon, H. A. (1972). *Human problem solving*. Prentice Hall.
- Norman, E., Blakstad, O., Johnsen, O., Martinsen, S. K., & Price, M. C. (2016). The relationship between feelingsof-knowing and partial knowledge for general knowledge questions. *Frontiers in Psychology*, 7.
- Reder, L. M., & Ritter, F. E. (1992). What determines initial feeling of knowing? Familiarity with question terms, not with the answer. *J. of Exp. Psych.: Learning, Memory, and Cognition*, *18*(3).
- Schreiber, T. A., & Nelson, D. L. (1998). The relation between feelings of knowing and the number of neighboring concepts linked to the test cue. *Memory & Cognition*, 26(5).
- Schwartz, B. L. (1994). Sources of information in metamemory: Judgments of learning and feelings of knowing. *Psych. Bullet. & Rev.*, 1(3).
- Schwartz, B. L., & Metcalfe, J. (2011). Tip-of-the-tongue states: retrieval, behavior, and experience. *Memory & Cognition*, 39(5).
- Schwartz, B. L., Pillot, M., & Bacon, E. (2014). Contextual information influences the feeling of knowing in episodic memory. *Consciousness & Cognition*, 29.
- Singer, M., & Tiede, H. L. (2008). Feeling of knowing and duration of unsuccessful memory search. *Memory & Cognition*, 36(3).

# Do Models of Syllogistic Reasoning Extend to Generalized Quantifiers?

### Maximilian Mittenbühler (max.mittenbuhler@gmail.com)

Department of Computer Science, University of Freiburg, Germany

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

Predictive Analytics, Chemnitz University of Technology, Germany

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

Predictive Analytics, Chemnitz University of Technology, Germany

### Abstract

Over the last century, a large variety of cognitive models for syllogistic reasoning have been developed, thereby advancing our understanding about the way humans process reasoning tasks. Most of the research was performed on a restricted set of quantifiers from first-order logic, which simplified model evaluations and comparison due to a well-defined set of tasks and the availability of complete and extensive datasets. However, as everyday reasoning and communication relies on a large variety of quantifiers, the scope and potentially also the generalizability of the models was severely limited. The present work aims at extending the domain of syllogistic reasoning to a wider set of quantifiers by (I) presenting a benchmarking dataset that includes the quantifiers "Most" and "Most not", (II) evaluating two state-of-the-art models (the Probability Heuristics Model and mReasoner) with respect to their ability to account for individual reasoners and (III) set the predictive performance of the cognitive models into perspective by comparing them to upper bounds and providing in-depth insights about their strengths and weaknesses.

**Keywords:** Syllogistic Reasoning; Generalized Quantifiers; Cognitive Modeling; Probability Heuristics Model; Mental Model Theory; mReasoner

#### Introduction

Syllogistic reasoning is one of the oldest domains for researching human reasoning capabilities, with a history of over a century (Störring, 1908). As an example, consider the following syllogism:

- (1) Most Mammals are Land Creatures.
- (2) Most Mammals are Intelligent Creatures.

What, if anything, follows from these two premises?

In general, syllogisms consist of two premises making a quantified statement about the relation between two terms (e.g., mammals and land creatures in the first premise), that are connected via a term occurring in both statements (middle-term; e.g., mammals). In this example, the task would be to infer the relation between the two end-terms (*land creatures* and *intelligent creatures*), which can be done by considering how each of them relates to *mammals*. In this case, it can be concluded that at least *some* land creatures are also intelligent (and therefore some intelligent creatures live on land). Generally, research has shown that humans systematically deviate from logic (e.g., Khemlani & Johnson-Laird, 2012), which prompted the development of theories that describe and explain how humans reason about such tasks.

Throughout the article, we will use common abbreviations (e.g., Pfeifer, 2006) for the syllogisms, using single letters for

the quantifiers: A, I, E, O, T and D for All, Some, No, Some...not, Most and Most...not, respectively. Furthermore, we denote the order of the terms in the premises with a so-called *figure*. In this article, we use the definition of figures used by Khemlani & Johnson-Laird (2012), which is shown in the following table (leading to the abbreviation TT4 for the syllogism in the example):

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

Most research about syllogistic reasoning focused on a restricted subset of syllogisms that only considered the quantifiers from first-order logic (All, Some, No and Some ... not, which we refer to as classic quantifiers) while excluding generalized quantifiers like most and few. This restriction has allowed researchers to investigate a well-defined subset of 64 possible syllogisms with 9 possible conclusions: 8 quantified conclusions (4 quantifiers with 2 directions each) and the option that there is no valid conclusion (NVC). Currently, a multitude of theories explaining how humans solve these syllogistic tasks exist (for an overview see Khemlani & Johnson-Laird, 2012), which were thoroughly evaluated in terms of their ability to predict general human behavior as well as adapt to individual reasoners (e.g., Khemlani & Johnson-Laird, 2012; Riesterer, Brand, & Ragni, 2020a). For these evaluations, complete datasets, i.e., where each participant solved all tasks of the domain (64 in this case), are pivotal as they allow an investigation on the level of individual participants without introducing a potential bias due to the task selection. Furthermore, purely data-driven models that require a rich data foundation can also be included as an upper bound for performance (Riesterer, Brand, & Ragni, 2020b).

Unfortunately, restricting the research focus to only four, firstorder logic based quantifiers limits the applicability of the resulting theories to everyday communication and reasoning (e.g., Pfeifer, 2006), which involves a variety of qualitatively different quantifiers. The restriction therefore severely limits the scope of the understanding we obtained from our theories. However, while it would be beneficial to extend the set of quantifiers, it comes at a cost: Each additional quantifier exponentially increases the number of tasks, making the collection of a complete dataset challenging if not impossible. Selecting the quantifiers is also an arbitrary decision, as they are not part of an established framework, such as first-order logic that justifies the distinct restriction to a certain set.

To address this issue, we have collected a complete dataset with the additional quantifiers Most and Most ... not, amounting to a total of 144 syllogisms per participant, in a recent study (Brand et al., in press). Importantly, these generalized quantifiers can not be expressed in first-order logic for sets of unknown sizes, which is usually the case for syllogistic tasks. Therefore, they could provide insight into a different facet of human syllogistic reasoning. Our analyses showed that the inclusion of additional quantifiers did not change the behavior on the classic syllogisms, leaving the validity of previous research efforts unchallenged. However, the vast majority of theories explaining syllogistic reasoning have exclusively been evaluated on the narrow set of classic syllogisms, and it remains unclear if these theories still apply to the wider domain of *generalized syllogisms*. To this end, the present work makes the following contributions: First, we repeated the study and collected additional participants in order to compile a dataset that is suitable for model evaluation and benchmarking in the domain of generalized syllogisms. Second, we evaluate two of the most prominent models for human syllogistic reasoning, mReasoner and the Probability Heuristics Model (PHM), which are both able to handle the additional quantifiers. We specifically focus on their capability to account for individual reasoning behavior as opposed to a distribution over a population. Finally, we analyze and discuss where the models succeed and where they fail at explaining human data by comparing them to several baseline models.

### **Related work**

### **Probability Heuristics Model**

The Probability Heuristics Model (PHM Chater & Oaksford, 1999; Oaksford & Chater, 2001) assumes that people's everyday reasoning does not follow logical validity of quantified assertions, but their probabilistic validity instead. The probabilistic validity (or p-validity) of a conclusion is defined by the conditional probability of the end-terms, which in term is determined by the conditional probabilities described in the premises (where an end-term is one of the terms that are to be connected in the syllogistic task). The PHM proposes that people do not deduce pvalidity mathematically but instead use a number of heuristics that converge to p-validity. These heuristics are based on the notion of p-entailment, describing that certain quantifiers probabilistically follow from others (for instance, "All" entails "Some"), and the notion of informativeness, detailing that less probable and therefore more specific quantified assertions are more informative. This yields the informativeness order of quantifiers: All > Most> Most not > Some > No  $\ge$  Some not. To generate a conclusion candidate, the PHM uses the following three generative heuristics (G1-G3): First, the min-heuristic (G1) identifies the premise with minimal informativeness (min-premise) to determine the quantifier of the conclusion. Second, an alternative candidate quantifier that probabilistically follows the quantifier from G1 is proposed (p-entailment, G2). Finally, the direction of the conclusion is determined by the attachement heuristic (G3). If the min-premise from G1 starts with an end-term, the respective term is used as the subject of the conclusion. Otherwise, the end-term of the remaining premise (max-premise) that features the most informative quantifier is used as the subject of the conclusion.

The PHM also assumes that people may test their initial deductions. It proposes that this process comprises a further two heuristics (T1 and T2), which evaluate how much confidence should be granted to the conclusion candidate (either the candidate with the quantifier determined by G1 or G2). To this end, the informativeness of the max-premise is considered by the *max-heuristic* (T1). It is assumed that confidence and the informativeness of the max-premise are coupled, which means that NVC can be concluded if the confidence is too low (Copeland, 2006). Additionally, the *O-heuristic* (T2) postulates that *Some not* (O) should generally be avoided in conclusions due to their lack of informativeness. However, given the mechanism of the max-heuristic, O-conclusions already are the conclusions with the lowest confidence, which makes the O-heuristic more a refinement than an independent heuristic.

It is important to note that the interpretation of the quantifiers assumed by PHM excludes *All* from the quantifier *Most* (i.e., if *Most A are B*, then *All A are B* does not hold). However, *Some* also includes the possibility of *All*, following the traditional interpretation from first-order logic. Negated quantifiers are treated analogously.

#### mReasoner

Another prominent theory for syllogistic reasoning is the Mental Model Theory (MMT; e.g., Johnson-Laird, 2010). MMT assumes that reasoners construct a mental model representing the information provided by the premises of the syllogism that is then used to derive a conclusion. It thereby follows a four-step procedure (Copeland, 2006): The first premise is used to create a mental model representing the information by an instantiated set of entities that are assigned to the syllogistic terms of the premise based on the respective quantifier. Then, the mental model is extended by the second premise, thereby integrating information about the third syllogistic term. In the third step, a conclusion candidate is derived from the mental model. Finally, the conclusion candidate is tested by a search for counterexamples, that checks if the conclusion candidate holds up to alternative mental models that are consistent with the premises. If a counterexample is found, the mental model is either corrected and a new conclusion candidate is derived, or the process is aborted and NVC is concluded. If no counterexample is found, the candidate is accepted as the conclusion to the syllogism.

This process is implemented in the LISP-based cognitive model mReasoner<sup>1</sup> (Khemlani & Johnson-Laird, 2013). It uses four parameters associated with the inference process (Khemlani & Johnson-Laird, 2016):  $\lambda$  determines the maximum number of entities in the initial mental model by specifying a Poisson distribution from which the number of entities is drawn.  $\varepsilon$  then determines the completeness at which the premise information is represented within the entities. Finally,  $\sigma$  controls the likelihood to engage in the search for counterexamples.  $\omega$  then controls the behavior of mReasoner in the case that a counterexample was found by specifying the probability of weakening the conclusion quantifier and re-engaging in the search for counterexamples. If

<sup>&</sup>lt;sup>1</sup>https://github.com/skhemlani/mReasoner

a counterexample was found and the conclusion quantifier is not weakened, NVC is concluded instead.

**Expanding mReasoner to generalized quantifiers** Building mental models of quantified assertions containing generalized quantifiers poses a particular challenge to mReasoner because of the ambiguity of the quantifiers *most* and *most not* under certain circumstances (S. Khemlani, personal communication, March 3, 2022). To incorporate this, the Authors have equipped mReasoner with a more general model-building system than that required for syllogisms that only contain the classic quantifiers. More specifically, it takes advantage of its ability to generate mental models of different sizes (governed by its  $\lambda$ -parameter) as well as its stochastic mode. By incorporating the ability to parse generalized quantifiers in the stochastic model-building system, mReasoner can represent statements containing "most" in figure 2 or 3, which it would not be able to do otherwise (S. Khemlani, personal communication, March 3, 2022).

Method

# Data

In a previous study, the responses of 31 participants to 144 syllogisms were collected over the course of three sessions in order to minimize fatigue (Brand et al., in press). The study comprised all 64 syllogisms with the first-order logic quantifiers All, Some, No and Some not as well as 80 additional tasks consisting of syllogisms with the generalized quantifiers Most and Most not. To minimize biases due to the content of the syllogisms, hobbies and professions were used for the terms. The study thereby covered all syllogisms that could be constructed from the 6 quantifiers. Participants were asked to give either a quantified conclusion following from the premises or to respond with No valid conclusion, if no conclusion was possible. For the present work, we re-ran the study and extended the dataset by another 34 participants. The following analysis is therefore performed on a dataset consisting of 65 participants (mean age: 39.1, age SD: 14.0, female: 52.3%), where each responded to all 144 syllogistic tasks. The dataset and materials for the analysis are publicly available on GitHub<sup>2</sup>. Note that for assessing the correctness of participants' responses, we use the common interpretation that Most(A,B) for finite sets A and B as  $|A \cap B| > |A - B|$ , with  $|\cdot|$  being the size or the number of their elements (e.g., Westerståhl, 1989; Novák, 2008). Therefore, we are treating Most as More than half, which means that All also implies Most. However, no specific interpretation for the quantifiers was instructed in the study, so that the participants' understanding of the quantifiers are reflected in their response behavior.

### **Model Evaluation**

For the following analyses, we used the Cognitive Computation for Behavioral Reasoning Analysis (CCOBRA) framework<sup>3</sup> and its *coverage* evaluation type (see Riesterer, Brand, & Ragni, 2020a). In this type of evaluation, the parameters of both PHM and mReasoner are first optimized for each participant by grid searching the parameter space and selecting those parameter settings that

191

yield optimal mean accuracy. Using the optimal parameter settings for each participant, the models are then queried for predictions of the responses that the participant gave for all tasks. Overall model predictive performance is assessed via the achieved accuracy. Technically, the models were thereby fitted to the exact responses that it later has to predict. This means that a fully data-driven model with no restrictions on the number of parameters would be able to achieve a perfect prediction. However, cognitive models are restricted by the number and expressiveness of their parameters: The parameters should reflect and control meaningful mechanisms in the model's processes. Therefore, the *coverage* evaluation assesses the models' capabilities to represent the individual response patterns within the framework of their assumed processes and mechanisms and by that explaining the individual behavior.

# PHM

In the following analyses we build upon a recent Python-based implementation of PHM, which used binary parameters to fit the model to individual reasoners (Riesterer, Brand, & Ragni, 2020a). In their implementation, a parameter for each confidence in a certain quantifier was implemented. Additionally, a parameter was introduced for the p-entailment, which specified if the conclusion based on the min-heuristic or the p-entailment should be used. While the parameters are usually continuous and interpreted as probabilities, the implementation was aiming at individual reasoners instead of a group of reasoners. Therefore, the parameters could be binary: As each participant usually only solves each task once, a prediction of the specific response has to be achieved by a model, instead of a distribution of possible responses. This simplifies the fitting process, as the number of parameters is quite low and allows for a exhaustive grid search in the parameter space. Additionally, the parameter space is further restricted by the additional constraint that the confidences follow the same ordering as the informativeness. Therefore, the confidence for Some can never be higher than the confidence for All. As the original implementation by Riesterer, Brand, & Ragni (2020a) only considered the 4 quantifiers from first-order logic, we extended the model to the generalized quantifiers Most and Most not. It is important to note that we incorporated Most not in the same way as Few was used in the original description of PHM by Chater & Oaksford (1999).

Furthermore, Chater & Oaksford (1999) also consider *weak p-entailment*, which would allow *Most* and *Most not* to follow from the quantifiers *Some* and *Some not*. In our implementation, we do not consider weak p-entailment, which implies that generalized quantifiers in conclusions are never considered for the classic syllogisms.

#### mReasoner

For mReasoner, we used the Python-based model by Riesterer, Brand, & Ragni (2020a) which internally relies on the original LISP-implementation of mReasoner in order to rule out differences in the model behavior. The model was then extended to the quantifiers *Most* and *Most not*, and the updated version of mReasoner was used. The parameters were fitted using a grid-seach with 6 steps for each parameter. For  $\varepsilon$ ,  $\omega$  and  $\sigma$  which have a range from 0 to 1, this yiels a stepsize of 0.2. The range

<sup>&</sup>lt;sup>2</sup>https://github.com/Shadownox/iccm-syl-genquant-models

<sup>&</sup>lt;sup>3</sup>https://github.com/CognitiveComputationLab/ccobra

for  $\lambda$  was chosen to be between 3 and 8 (which leads to a stepsize of 1). While Riesterer, Brand, & Ragni (2020a) used the full range of  $\lambda$  with  $\lambda \in [0,8]$ , the extension to generalized quantifiers required higher values to work. Furthermore, it was required that  $\varepsilon < 1$ . To account for the randomized nature of the inference process, each configuration was sampled 10 times.

# **Baseline Models**

Similar to existing benchmarking settings for syllogistic reasoning (e.g., Brand et al., 2020; Riesterer, Brand, & Ragni, 2020a,b), we included a Random model as a lower bound of the performance, which uniformely selects one of the possible response options, as well as the most-frequent answer (MFA), which uses the most frequently given response to a syllogism as a prediction. The MFA is also the best model when not fitting to individual participants. To assess the maximum predictive performance (theoretically) achievable with the present dataset, we included a purely data-driven model as an upper bound (for a similar application of data-driven models, see Riesterer, Brand, & Ragni, 2020b). We used a user-based collaborative filtering model (UBCF), which is a neighborhood-based model from the field of recommender systems that relies on the behavior of other users to predict a targets' behavior (for an in-depth description, see Aggarwal, 2016). Based on the responses given to all syllogisms except for the one to be predicted, a neighborhood of the k most similar participants is created. When predicting the response of a target participant to a syllogism, each neighbor votes for the responses, where the vote is weighted by the respective similarity to the target participant. To discount less similar neighbors even more, the similarity can be raised to the power of an exponent-factor exp. The final prediction is then the response with the most votes. For this analysis, we used the parameters k = 12 and exp = 3, which was found by applying a grid-search for the best parameters. One advantage of the UBCF is the similarity to the MFA, as the MFA can be interpreted as a special case of the UBCF: If no information about the target participant is available, the similarity is not defined, leading to the neighborhood consisting of all other participants available. Therefore, the prediction would just be the most frequently given response. Therefore, the UBCF can be considered as an extension of the MFA to the individual level.

### Analysis

# **Overall Model Performance**

Figure 1 shows how well mReasoner and PHM, as well as the three baseline models, were able to predict participants' responses. Both mReasoner, with on average 39.7% correct predictions, and PHM, with 41.7% correct predictions, performed noticeably above chance-level at 7.7% and were able to surpass the MFA-model at 35.6%. The general performance indicates that both models can at least partly explain peoples' responses. The difference to the MFA-model did, however, not reach significance (Mann-Whitney-U test: U = 1882.5, p = 0.29 for mReasoner, U = 1780.5, p = 0.12 for PHM, respectively), which shows that the ability to adjust to individual response behavior is still lacking, which is also corroborated by the performance of the UBCF model with 45.2%. It becomes



Figure 1: Predictive accuracy of the tested models on all syllogisms. Each point represents the accuracy for predicting a specific participant. The triangle denotes the respective mean.



Figure 2: Predictive accuracy of the tested models for classic syllogisms (blue) and generalized syllogisms (orange). Each point represents the accuracy for predicting a specific participant. Triangles depict mean accuracy scores.

apparent that there is still a substantial amount of information available in the data, which is not yet covered by the models' mechanisms. Despite the general problems with adapting to individual reasoners, both cognitive models seem to be able to adapt to a small group of reasoners exceptionally well, indicating that the models generally are able to adapt to individuals, but still miss out on important mechanisms. This highlights the potential for further improvements of cognitive models for syllogistic reasoning.

### Performance for Classic and Generalized Quantifiers

As our focus was on expanding mReasoner and PHM to the domain of generalized quantified syllogisms, the differences in the model performance between the two domains are especially important. Therefore, Figure 2 depicts the results broken down by the respective task domain (i.e., classic syllogisms and syllogisms with generalized quantifiers). Note that, like in the general performance analysis, the models are still fitted based on all tasks, as we aim at evaluating the models' abilities to generalize across the different task types. It becomes apparent that all models perform worse on generalized quantified assertions by about five percentage points (except for the chance-level baseline).



Figure 3: Errors when predicting responses for classic syllogisms (a) and syllogisms with generalized quantifiers (b) for PHM and mReasoner, respectively. Darker blue colors denote a higher number of errors. Both models were fitted on the full set of syllogisms.

However, the fact that the UBCF model's performance dropped to a similar extent indicates that this drop could be attributed to the participants' response behavior being less clear. This is corroborated by the fact that classic syllogisms were easier for the participants to solve (mean correctness: GenQuant = 0.25; Classic =0.34), which in turn can minimize individual differences for some tasks (i.e., if there is an obvious answer). Yet again, a wider range of responses to generalized syllogisms could not be found: We compared the entropy (see Shannon, 1948) as a metric for uncertainty of the participants' response distributions for both, the classic and the generalized quantifiers, in order to check for a systematic difference in the range of responses. The entropies showed no substantial difference between both task types (GenQuant = 3.30; Classic = 3.22). However, easier tasks can nevertheless help to improve the consistency within participants' responses (i.e., the participant would reliably show the same response patterns), which makes it easier for models to replicate the response pattern, which might explain the differences between both task types.

#### **Error Analysis**

To see where the predictions of the cognitive models did not capture the human responses well, we investigated for which responses the most errors occurred (see Figure 3). For PHM, an indistinct picture emerges. While it seems that PHM generally tends to respond NVC too frequently, it does so for both task types in a comparable fashion. It also seems to misjudge the direction of the conclusion when not responding with NVC in both task types. However, while the errors based on NVC and the direction explain the majority of the errors on the classic syllogisms (65,4%), this does not hold for the generalized quantifiers (49%): Here, PHM also often mixes the quantifiers up, especially between I, D and O. It seems to be the case that participants are more variable in their use of these quantifiers as to the fixed order of informativeness PHM relies on.

When focusing on the results for mReasoner, a much clearer picture emerges. While the errors on the classic syllogisms are rather similar to the errors shown by PHM, NVC accounts for the vast majority of errors for the generalized quantifiers. NVC is the logically correct response for the majority of tasks, especially for generalized syllogisms (*GenQuant* = 76.3%, *Classic* = 57.8%), which seems to be reflected in mReasoner's mechanisms. However, this is not reflected in the participants responses,



Figure 4: Parameter distributions after fitting mReasoner and PHM to each participant for the classic syllogims (blue) and syllogisms with generalized quantifiers (orange).

which do not show a difference in their NVC response behavior (GenQuant = 21.2%, Classic = 21.6%). Furthermore, mReasoner's mechanisms for giving NVC-responses seem to be too coarse: If it needs to respond with NVC for several tasks, it seems to overshoot substantially. The differences between classic and generalized syllogisms also seem to reflect that mReasoner handles generalized quantifiers differently than the classic quantifiers.

#### **Parameter Analysis**

Based on previous analysis, we investigated the parameters that the models would use for both task domains when fitted to them separately. Figure 4 shows the parameter distributions for both models when fitted to the responses of each individual participant on the classic syllogisms and the generalized syllogisms, respectively. Interestingly, the parameters of mReasoner do not show substantial differences except for  $\omega$ , which controls behavior when a counterexample is found. While mReasoner was shown to respond with NVC too frequently, the difference in  $\omega$  indicates that NVC was in fact moderated by the parameters, as it means that a conclusion in case of a found counterexample is rather weakened than directly concluding NVC. Generally though, the parameters indicate that the mReasoner's performance would not change much if fitted to the generalized quantifiers directly, which implies that the performance was not impeded by a generalizability problem (i.e., having to find parameters that work for both, classic and generalized syllogisms), but rather due to a general inability to account for certain response patterns occurring for generalized syllogisms. For PHM, the results are generally more shifted towards responding with NVC for the generalized quantifiers, by having a lower confidence for all quantifiers. Although differences between both task types show, the adaption to generalized tasks is mainly done by the specific parameters for the quantifiers T and D, which do not affect the classic tasks, as T and D can only become conclusion candidates if they are present in the premises (note that this would change if weak p-entailment was considered). In this regard, PHM has a distinct advantage over mReasoner, as it utilizes parameters that are specific for the extension to generalized quantifiers, while mReasoner relies on the same core parameters for all tasks.

### Discussion

In this work, we performed a thorough evaluation of the predictive capabilities of PHM and mReasoner when confronted with syllogistic reasoning tasks that include the generalized quantifiers Most and Most not. The evaluation was performed on a benchmarking dataset that contains the responses to all 144 syllogisms for all participants, which allowed an analysis on the level of individual participants. The cognitive models were compared with the most-frequent answer and an estimated upper-bound given by a data-driven model based on user-based collaborative filtering. Both cognitive models performed within expectations, as they managed to slightly surpass the MFA, although not significantly. However, a more detailed look into the performance for individual participants, it appears that they are able to capture some of the participants well and seem generally able to adapt to individual participants. However, their performance fell short of to the UBCF, which highlights the potential that is still left in the domain and indicates that the models' mechanisms are still not sufficient to cover the variety of response patterns shown by different individuals.

When focusing on the generalized quantifiers, the performance of all models dropped substantially (including the UBCF), which indicates that the noise-levels are higher on these tasks. This is supported by the lower correctness on these tasks, which can lead to less consistent response behavior. However, the cognitive models still managed to surpass the performance of the MFA, which shows that their general mechanisms can generalize from the four first-order logic quantifiers to an extended set of quantifiers. This is corroborated by an analysis of their parameters, which showed no substantial differences when fitted to the classic tasks or the generalized tasks only.

Given the performance of both models, no difference, on neither the classic nor the generalized syllogisms, is noticeable. Therefore, based on the predictive performance, the assumed underlying processes both seem to be equally plausible. However, when the errors of both models are analyzed in detail, differences become apparent. As it was already shown that models have difficulties with correctly predicting the NVC-response on the classic syllogisms (Riesterer, Brand, Dames, & Ragni, 2020), it was likely that the problem carried over and thereby accounted for a part of the errors. This shows for both models across both task types, with NVC being an important source of error. However, the magnitude of the problem greatly differs between the models: On the one hand, the type of errors of PHM remain largely the comparable between classic and generalized syllogisms with NVCand direction-related errors, despite an increase in noise-like errors on the generalized tasks. On the other hand, mReasoner fails to replicate the participants' NVC-behavior and drastically overshoots with the frequency of NVC responses on generalized syllogisms, while being comparable to PHM on classic tasks. This indicates that its mechanism for handling generalized syllogisms is currently inferior to PHM, although the problem seems to be covered by the high number of NVC responses that make predicting NVC frequently a rather safe strategy.

However, even though mReasoner currently seems to lag a bit behind, it is important to note that PHM utilizes specific parameters for the respective quantifiers, while mReasoner relies on a fixed set of parameters and its core mechanisms. This can greatly affect the future development, as it will be important to further extend the scope of the domain in order to advance our understanding in the field of syllogistic reasoning. While PHM can be rather easily adapted to additional quantifiers, it also means that the complexity of the model increases directly with the number of supported quantifiers, which can become an important factor when extending the domain further.

By providing a complete dataset and an evaluation of two stateof-the-art models, the present work aims at setting a starting point for extending modeling endeavors to an extended set of syllogisms. However, a large variety of other quantifiers are important for our everyday reasoning and communication, including more vague quantifiers like *Many* or counting quantifiers (e.g., *More than 3*). These possibilities have to be investigated in the context of syllogistic reasoning, in order to warrant the claim that the present models and our knowledge reaches beyond well-defined abstract tasks.

### Acknowledgements

Support to MR by the German Research Foundation, DFG (RA 1934/4-1, RA1934/8-1) is gratefully acknowledged.

### References

- Aggarwal, C. C. (2016). Neighborhood-based collaborative filtering. In *Recommender systems: The textbook* (pp. 29–70). Cham: Springer International Publishing.
- Brand, D., Mittenbühler, M., & Ragni, M. (in press). Generalizing syllogistic reasoning: Extending syllogisms to general quantifiers. In *Proceedings of the 44th Annual Conference of the Cognitive Science Society.*
- Brand, D., Riesterer, N., & Ragni, M. (2020). Extending Trans-Set: 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.
- 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.

- Johnson-Laird, P. N. (2010). Mental models and human reasoning. Proceedings of the National Academy of Sciences, 107(43), 18243–18250.
- 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.
- Khemlani, S. S., & Johnson-Laird, P. N. (2016). How people differ in syllogistic reasoning. In A. Papafragou, D. Grodner, D. Mirman, & J. C. Trueswell (Eds.), *Proceedings of the 38th Annual Conference of the Cognitive Science Society* (pp. 2165– 2170). Cognitive Science Society.
- Novák, V. (2008). A formal theory of intermediate quantifiers. *Fuzzy Sets Syst.*, 159, 1229-1246.
- Oaksford, M., & Chater, N. (2001). The probabilistic approach to human reasoning. *Trends in Cognitive Sciences*, 5, 349-357.
- Pfeifer, N. (2006). Contemporary syllogistics: Comparative and quantitative syllogisms. In G. Kreuzbauer & G. J. W. Dorn (Eds.), Argumentation in Theorie und Praxis: Philosophie und Didaktik des Argumentierens (pp. 57–71). Wien: LIT.
- Riesterer, N., Brand, D., Dames, H., & Ragni, M. (2020). Modeling human syllogistic reasoning: The role of "No Valid Conclusion". *Topics in Cognitive Science*, 12(1), 446–459.
- 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). Predictive modeling of individual human cognition: Upper bounds and a new perspective on performance. *Topics in Cognitive Science*, 12(3), 960–974.
- Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3), 379-423.
- Störring, G. (1908). *Experimentelle Untersuchungen über einfache Schlussprozesse*. W. Engelmann.
- Westerståhl, D. (1989). Aristotelian syllogisms and generalized quantifiers. *Studia Logica*, 48(4), 577–585. doi: 10.1007/BF00370209

# Modeling Optimal Arousal by Integrating Basic Cognitive Components

Kazuma Nagashima (nagashima.kazuma.16@shizuoka.ac.jp),

Junpei Nishikawa (nishikawa.jumpei.16@shizuoka.ac.jp),

Rvo Yoneda (voneda.rvo.17@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

#### Tetsuya Terada (terada.t@mazda.co.jp)

Mazda Motor Corporation

#### 3-1 Shinchi, Fuchu-cho, Aki-gun, Hiroshima-ken, 730-8670, Japan

#### Abstract

Mind-wandering occurs as emotional arousal decreases, which is related to the level of mastery of the current task. As a worker becomes more proficient in a task, the cognitive resources required to perform the task decrease. Then, surplus resources emerge and are naturally directed to "defaultmode thinking," which people usually engage in outside the task. As mind-wandering continues, this default-mode thinking becomes more active and affects the task performance. In this study, we describe this process by combining the basic functions of the cognitive architecture Adaptive Control of Thought-Rational (ACT-R). The chunk activation mechanism represents the on- and off-task thinking loops. Furthermore, we introduce stochastic fluctuation in the chunk activation to change the transition probability between these loops. This fluctuation is assumed to be driven by parasympathetic activity, which increases over time and is suppressed by novel stimuli. To develop this physiological change, this study uses the ACT-R temporal module. Simulations using these modules demonstrate the inverse U-shaped relations between task performance and task continuation. Such a process is consistent with theories of optimal levels of arousal.

**Keywords:** optimal level of arousal, homeostasis, mind-wandering, cognitive resource, ACT-R

### Introduction

People often think and dream about things unrelated to the current task. This state is called mind-wandering and is reported to occur more than half the time humans are awake (Killingsworth & Gilbert, 2010). Therefore, mind-wandering can be considered the normal state (default mode) of humans. Although mind-wandering is assumed to promote creative thinking (Baird et al., 2012), it leads to a decline in task performance and triggers accidents caused by distraction from the task.

Mind-wandering is one phenomenon caused by decreased emotional arousal during a task. A similar process is sometimes expressed as mental fatigue, boredom, or habituation. These wide varieties of mental activities are related to an optimal level of arousal for better task performance (Yerkes & Dodson, 1908; Hebb, 1955; Easterbrook, 1959). Some researchers have proposed that the optimal level of arousal is influenced by the difficulty of the task (Oxendine, 1970; Csikszentmihalyi, 1990). Both excessive and insufficient arousal levels for the current task difficulty negatively affect performance. In other words, the task performance is related to arousal level by an inverse U-shaped function, the peak of which shifts depending on the task's difficulty. This inverted U-shaped curve is considered to apply to changes in task performance over time. As the task proficiency progresses, the task performance increases and becomes easier for the current workers. Simultaneously, the level of arousal (attention or cognitive resources) required to accomplish the task decreases. Then, surplus cognitive resources emerge, and they are naturally directed to "default-mode thinking," which workers prefer to use in their everyday life. As this process repeats, they lose motivation to continue the task, and their task performance gradually degrades. This transition eventually creates a inverse U-shaped curve relating the attention directed to the task (the arousal level required by the task) and the task continuation (similar mechanism is proposed by Shenhav et al. (2013)).

Many studies have been conducted concerning human cognitive functions related to the theory of the optimal level of arousal. However, detailed computational models describing the changes in performance and arousal level over time have not been fully developed. In this study, we represent this process using a cognitive architecture, ACT-R (Adaptive Control of Thought-Rational; Anderson, 2007). Like many other cognitive architectures (Kotseruba & Tsotsos, 2018 for a review), ACT-R provides modules corresponding to functions used repetitively across several tasks. ACT-R has multiple modules involved in learning tasks, and the combination of these modules can represent the complex nonlinear relationships between mastering the task and task motivation. Based on this idea, this study tries to describe these arousal changes by integrating the primitive cognitive modules provided in ACT-R.

In the following section, we will introduce related studies concerning the abovementioned goal of the study. Following this, the target human behaviors concerning the optimal level of arousal will be presented. Then, the ACT-R model integrating several primitive cognitive components to simulate these specific behavior patterns will be described. The simulation results will present a case of a U-shaped task performance change. In the final section, we will discuss the implications and limitations of this study.

### **Related Works**

This study aims to model the optimal arousal level by combining primitive functions in ACT-R. This section presents two directions of previous studies: a human physiological mechanism and research on ACT-R.

#### **Computational Models of Human Homeostasis**

Physiological processes drive human arousal. Therefore, the optimal level of arousal described in the previous section can be interpreted as the maintenance of homeostasis in biological systems, which is a self-regulating process that fluctuates to maintain its optimal state (Billman, 2020; Cannon, 1929). Because of homeostasis, organisms can adapt to changing environments.

Computationally, homeostasis has been explained by the theory of predictive coding, also known as the free energy principle. Predictive coding is the concept that the brain minimizes the prediction error between sensory signals and internal prediction signals by which the brain perceives the environment (Friston, 2010). An organism is assumed to desire the minimization of long-term prediction errors caused by mismatches between predictions from experience and perceptions of current conditions. Mismatches also decrease as the organism masters the task. Thus, predictive coding describes human behavior in terms of a balance between minimizing the prediction error for the task and increasing the prediction accuracy. This relationship is also compatible with the exploration-exploitation relation discussed in the study of reinforcement learning (Sutton & Barto, 1998).

We consider that the above concepts of homeostasis and prediction errors explain the inverted U-shape of arousal level and task performance. Continuation of the same task leads to the saturation of prediction errors and increases the desire to explore new environments. However, the theory of homeostasis has difficulty describing the process of arousal changing over time. To solve this problem, we review the models developed using ACT-R.

### **ACT-R Models Regarding Arousal Change**

Recently, some researchers have developed mind-wandering models using the activation mechanism of ACT-R. Van Vugt et al. (2015) implemented a model that recalls memories unrelated to the task while the task is being executed. Through simulations using this model, they represented how the task continuation induces mind-wandering and how it affects the task performance.

Other studies have focused on fatigue, which is also closely related to arousal changes over time. Gunzelmann et al. (2009) constructed a model representing the effects of fatigue on the execution of procedural memories. Specifically, they manipulated the parameters relating to the computations of utilities for production to represent the degree of fatigue. Gunzelmann et al. (2012) also constructed a mechanism for fatigue in memory activation, which affects the success of memory retrieval during the task. These changes in subsymbolic parameters over time affect the performance of the task and can define a inverse U-shaped curve representing the relation between the task continuation and reaction time (Atashfeshan & Razavi, 2017).



Figure 1: Task interface. Screenshot of task window (left) and overall view of the line (right).

However, these studies have not explicitly discussed the correspondence of these parameter changes to human physiological mechanisms. Concerning the logic behind these models, Ritter (2009) defined emotion as physiological substrates affecting cognitive parameters, such as activation. This idea has been instantiated in ACT-R/ $\Phi$  (Dancy et al., 2015), which combines cognitive processes in ACT-R with physiological mechanisms. Although this ACT-R extension successfully demonstrates the complex dynamics that emerge from interactions between physiology and cognitive components, it does not explain how those relations change over time.

As described in the first section, mind-wandering can be assumed to be a side effect of mastering a task. From this viewpoint, cognitive models of skill acquisition should be integrated with the models of arousal changes. Several computational models (Anderson et al., 2019; Kim & Ritter, 2015) has been proposed in ACT-R to represent a nonlinear theory of mastery (Fitts, 1964). Specifically, Anderson et al. (2019) recently proposed an ACT-R module enabling mastering primitive perceptual and motor coordination. We consider that an integrated account of optimal arousal theory can be developed using this module.

In summary, ACT-R has been used for various cognitive function models in different situations. By referring to these studies, we believe that it will be possible to construct a detailed model for the target of this study.

# Human Data

# Objective

Before presenting our model, this section presents data concerning changes in human arousal in a simple perceptual and motor task. To collect data from various individuals, we recruited participants from a crowdsourcing service (Lancers.jp).

# Task

We set up a line-following task (Maehigashi et al., 2013) to examine fluctuations in arousal. Figure 1 shows the task interface. A polyline displayed on the screen automatically scrolls from top to bottom by one pixel every 40 ms (25-fps screen updates). The participants were required to follow the polyline (stay online) by moving the circular object left or right.

We chose this task because there is a publicly available ACT-R model (Morita et al., 2020). Moreover, it is relatively easy to modify the complexity of this task by manipulating parameters such as the ratio of vertical lines included in the polyline patterns. In this study, to induce arousal change in a short period, we set this parameter at 90%. The right panel of Figure 1 shows the overall pattern constructed. This pattern was repeated in a one-minute cycle in the following experiment and simulation.

In addition, to examine changes in arousal level during the task, we designed a pop-up window (probe) asking participants to respond to the degree to which they were focused on the task. The probe was presented at an interval of approximately 50 s, with randomized noise added to the interval.

#### Method

Eighty-one participants finished the experiment procedure, where they first accessed the online system and read the instructions for the task at their own pace. After completing a test to confirm their understanding of the task, they engaged in the line-following task for 30 min.

In this experiment, we set up three BGM conditions to examine environmental factors influencing the arousal changes during the task. The participants engaged in the task under the following conditions:

- No BGM: No music was presented (n = 27)
- Low BPM: The task environment included music at 80 beats per minute (BPM) (n = 25)
- High BPM: The task environment included music at 120 BPM (*n* = 29)

However, this paper did not focus on the difference between the conditions.

#### Results

Figure 2 shows the offline rate (the percentage of time that the circle did not follow the line). These results are shown for 30 segments of 1 min each of the 30-min task execution. In each condition, the offline rate decreased in the initial phase, suggesting that mastering the perceptual and motor coordination occurred during this early phase. Although the difference between conditions was not apparent, the average offline rate in the high-BPM condition (the thick red line) increased over time (after 18 segments), suggesting cases of U-shaped transitions <sup>1</sup>. The model presented in the following section tried to generate such a trend in task performance.



Figure 2: Participant performance in the line-following task. The thin line shows the cases per participant, and the thick lines show their mean. The vertical axis is on a logarithmic scale.

### Model

We constructed a model following the four previous ACT-R models: the perceptual-motor process (Morita et al., 2020), the mind-wandering mechanism (van Vugt et al., 2015), time perception (Taatgen et al., 2007), and mastering the motor process (Anderson et al., 2019). By combining the first two, the current model represented the execution of the task and the deviation from the task. We also represented arousal changes by applying the temporal module representing subjective time, while the effects of mastering the task on mind-wandering were also modeled as motor skill acquisition.

#### **Perceptual-Motor Process**

The model's state transitions were constructed based on the previous study (Morita et al., 2020), which are represented in Figure 3. As seen in the figure, the model consists of cyclic behaviors of perceptual and motor processing. These processes are realized by the functions implemented in the following modules.

**Visual Module** This module simulates interaction with the external environment. The visual module reads the symbols (e.g., the position of a circle or a turn in the line) necessary to perform the task from the external environment (in the model, a display on a virtually created window).

**Motor Module** This module simulates the operations required in the task. In the line-following model, the module executes key presses corresponding to the movement of the circle and responding to a probe.

**Declarative Module** This module stores symbolic chunks, a unit of symbolic information in ACT-R. These chunks

<sup>&</sup>lt;sup>1</sup>Because the study uses an offline ratio as the performance index, the observed U-shaped curve corresponds to the inverse U-shape curve discussed in the introduction.

include episodic memories, semantic knowledge, and the model's goals. The last chunk is important for representing mind-wandering in the line-following task. As in the previous mind-wandering model (van Vugt et al., 2015), two types of goals are available in the model: the goal for the current task execution and the goal for default-mode thinking. In addition to these two goal chunks, the model has chunks corresponding to individual memories that are not relevant to the current task.

**Goal Module** This module holds one of the two goal chunks retrieved from the declarative module. In addition, the module stores the current states of the task that are required to control the flow of the line-following task. Those states include the states obtained from the visual module, such as the circle position and the next turn position.

**Production Module** This module 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. Importantly, each transition (corresponding to a single application of a production rule) requires a specific time cost (50 ms), following the default setting of ACT-R. By accumulating these time costs, the model can predict the line-following performance in time constraints that is compatible with the human experiment.

In this model, the modules shown so far are integrated in the following steps:

- 1. The model sees the state of the external environment in the visual module (Figure 3 ①),
- It updates the current state of the goal module (Figure 3 2),
- 3. 3. It requests a goal chunk for the declarative module (Figure 3 (3)), and
- 4. It performs the necessary operations (key press) for the task through the motor module (Figure 3 ④)

After the above steps, the visual module checks for a new state in the external environment and returns to Step 1. If the declarative module retrieves the goal chunk that directs attention to default-mode thinking, it does not perform the operations required for the task (key presses). Instead, it enters a state of continued recall of memories outside the task (mindwandering). When the goal chunk about the current task is accidentally recalled in this process, the model returns to the task. The mechanism of switching between these two loops is further described in the next subsection.

### **Mind-Wandering Mechanism**

As described above, the model loop related to the task competes with the loop related to mind-wandering. The conflicts



Figure 3: Block diagram showing the model processing.

between the loops are then resolved by the activation mechanism, following the previous mind-wandering model (van Vugt et al., 2015). In the ACT-R theory (Anderson, 2007), the activation of a memory depends on its recency and frequency of use. That is, memories that were most recently or frequently used are more likely to be recalled. Thus, when the goal for the current task is highly activated in the early stage of the task, the perceptual-motor loop (the upper part of Figure 3) continues to be strengthened.

However, when a memory involved in mind-wandering is accidentally introduced during mastering the task, and no penalty is imposed, the probability of selecting the goal for the default mode of thinking (an activity that was frequently engaged in outside the task) increases. When mindwandering continues and the goal for the current task is no longer recalled, the model leaves the task.

### **Mastering Motor Control**

The accuracy of the perceptual-motor loop (the upper part of Figure 3) is improved by learning through the task. This learning is controlled by a tracker module in ACT-R 7.27, initially proposed by Anderson et al. (2019), based on the simulated annealing algorithm (Kirkpatrick et al., 1983). This module adjusts the continuous conditions for selecting motor operations based on positive and negative feedback from the environment.

In the model presented in Figure 3, the motor operations include "stop" (release key), "go right" (press the key assigned to the right), "go left" (press the key assigned to the left), and "continue" (continue the previous operation). In this motor operation selection, the distances between the circle and the line (a continuous value) obtained from the perceptual processes in Figure 3 ① are used as conditions. The current model specifically observes two distances, which are visible as two lines drawn on the screen (see Figure 1)<sup>2</sup>: the magenta line showing the distance between the circle and the nearest point on the line and the blue line showing the distance between the circle and the next turn on the line.

The tracker module automatically adjusts the boundaries of these values to select one of the four motor operations ap-

<sup>&</sup>lt;sup>2</sup>In the human experiment, these lines were removed.

propriately. If this adjustment is not appropriate, the model fails to follow the lines because the circle overshoots the line or executes the operation before it reaches the line. Appropriate coordination in this model is learned sequentially by receiving negative feedback for failing to follow the line. The tracker module has a subsymbolic parameter called temperature, which controls the fluctuations in the boundaries between motor operations. This parameter usually has a high value at the beginning of the task and decreases over time. In other words, the model engages in exploration in the early stages of the task, whereas it exploits the acquired coordination at the later stages. Therefore, it is assumed that the adjustment in boundaries between motor operations converges within a specific range at the appropriate temperature setting, leading to high perceptual-motor performance.

### **Homeostasis Through Time Perception**

As discussed above, the mind-wandering mechanism previously presented (van Vugt et al., 2015) has limitations in connecting physiological mechanisms. To address this issue, ACT-R/ $\Phi$  (Dancy et al., 2015) integrates ACT-R and physiological mechanisms. However, ACT-R/ $\Phi$  uses an entirely independent simulator of physiological variables. Therefore, we consider this model to have a unification problem between cognitive and physiological components. Furthermore, because it uses two separate components developed for different purposes, it seems difficult to claim that ACT-R/ $\Phi$  is a single consistent architecture. Therefore, this study attempts to construct the physiological mechanisms involved in mindwandering using only the basic modules incorporated in the original ACT-R while basing the concept on ACT-R/ $\Phi$ .

The concept of ACT-R/ $\Phi$  is that physiological mechanisms such as homeostasis play the role of modulators adjusting cognitive processes (Ritter, 2009). This idea assumes a correspondence between various physiological indices and subsymbolic parameters in ACT-R. A typical relation is the correspondence between the amount of epinephrine released when the sympathetic nervous system is activated (aroused) and ANS (activation noise s), one of the ACT-R noise parameters. The ANS parameter is used to determine the degree of fluctuation in recalling chunks from the declarative module. When ANS is low, the model exploits highly activated chunks, whereas when ANS is high, the model explores the various chunks. This behavior allows us to understand the arousal level of the model relative to the ANS. In this study, we adjusted the ANS according to the above ideas (small and large ANS representing high and low arousal, respectively) and modeled the arousal changes as the task progressed.

To implement the above relation, this study used the temporal module (Taatgen et al., 2007) built in ACT-R. It is pointed out that temporal cognition is modified by the attention directed to the main task. When the task is performed at high arousal levels, people feel that time flows quickly. In contrast, when the task is performed at lower arousal levels, they perceive a slower time flow. Therefore, we considered that a more integrated architecture could be achieved by expressing the arousal changes with the time perception module (temporal module).

Time perception in ACT-R is controlled by a mental timer (pacemaker). This timer counts the number of ticks (t) that have elapsed since it started, using the equation

$$t_n = a \cdot t_{n-1} + \varepsilon, \tag{1}$$

where a stochastic noise ( $\varepsilon$ ) is added for each count (*n*). This equation represents the nonlinear time perception explaining why estimates of time intervals over long periods are less accurate than estimates of time intervals within short periods.

To use this equation for arousal change, this study assumed that n was reset (n = 0) when the model perceived new events. Specifically, the reset occurred when the circle fell away from the line or a probe appeared on the display. Thus, the interval between counts increased with the increase in counts until the model received the above events.

In addition, we assumed that the decrease in the accuracy of time perception corresponded to a decrease in arousal over time. Specifically, we introduced the equation

$$ans = k \times t$$
 (2)

where k is a coefficient to adjust the decrease in arousal level with respect to time. By manipulating this, we explored the conditions in the U-shaped curve observed in the human experiment.

### Simulation

# Objective

We proposed that the model could represent an inverted Ushaped curve for task performance according to the optimal level theory. To confirm this behavior, we used the following four indices.

- (a) Concentration: Difference in activation between the goal chunks for the current task and default-mode thinking.
- (b) Mind-wandering ratio: Percentage of time default-mode thinking occurs in the goal module.
- (c) Offline ratio: Percentage of time where the circle does not follow the line.
- (d) ANS: Value calculated by Equation 2.

#### Settings

The simulation conditions were set up by changing the value of k in Equation 2 in three steps (0.01, 0.03, 0.06). The model with a small k corresponded to a highly focused situation, while the model with a large k corresponded to a distracting situation.

Feedback for the tracker module was determined when the model perceived the environment (Figure 3 ①). The tracker module gave the model positive feedback of 10 when it was online and negative feedback of 10 at the moment it went offline.



Figure 4: Simulation results. From the top, the concentration rate (logarithmic vertical axis), mind-wandering rate, offline rate (logarithmic vertical axis), and activation noise (logarithmic vertical axis) are shown. The red line is the mean of 10 runs; the error bars are the standard error multiplied by 0.5.

The simulation duration was set to 30 min to match that of the human experiment, and, as with the humans, 30 oneminute-long courses were used. In addition, at the beginning of the task, the goal for the current task was set to be more accessible to recall than the goal for default-mode thinking<sup>3</sup>.

#### Results

Figure 4 shows the simulation results. These results show the effect of k on the four indices. The smaller-k conditions (k = 0.01, 0.03) are associated with a higher concentration ratio, corresponding to less mind-wandering, a lower offline ratio, and lower ANS. In addition, the concentration ratio increased over time in those conditions. This trend is also reflected in the decreasing trend of the offline ratio in the smallest-k condition (k = 0.01), indicating that a small ANS fluctuation strengthens the current task's goal and keeps the task execution stable. In contrast, the task performance did not increase over time in the high-k condition (k = 0.06), though the motor learning progressed. Although the average trend of offline ratio is almost flat in the higher-k conditions (k = 0.03, 0.06), some cases improved the task performance in the middle of the task. The thick black line highlights the typical case, showing such improvement in the middle of the task. As seen in this case, some cases show a U-shaped curve, which was also observed in the human experiment.

### Conclusion

This study aimed to construct a model of arousal changes over time by integrating the primitive ACT-R modules. To achieve this goal, we first collected human behaviors in a simple perceptual-motor task and observed the U-shaped curves in some participants. We constructed a model of arousal change to reproduce such human behaviors by combining the perceptual-motor process, mind-wandering mechanism, time perception, and motor skill acquisition. These modules have different types of dynamics, and combining them is expected to reproduce the nonlinearity of arousal change, namely the theory of the optimal level of arousal. As a result, inverse U-shaped performance transitions over time in the task were observed in some cases.

The significance of this study is that physiological processes, which were previously considered independent modules, are represented in the ACT-R primitive modules. In contrast to previous studies (Gunzelmann et al., 2009, 2012) that used a computational physiological model, our model is original in that it integrates components that initially came from different backgrounds. We consider that to achieve a truly integrated understanding of the human mind, the approach of adding ad hoc parameters to the architecture is not exactly sufficient. This study can be viewed as an endeavor in refactoring complex cognitive architecture to be a unified theory of human cognition.

In the future, we need to proceed further with this approach. For example, we only manipulated the activation noise parameter reflecting the arousal level in this study. However, ACT-R includes several other noise parameters in the production, tracker, and temporal modules. Therefore, we need to explore methods of integrating such different noises. We also need to seek valid assumptions behind the correspondence between ACT-R's noise level and the physiological process through this process.

The experiment and model should also be improved. Although we manipulated environmental factors (background musics) in the human experiment, we did not find clear results. Revealing the robust factors leading to inverse Ushaped learning is critical for obtaining clear correspondence between human behavior and model simulation. By improving the experimental method and the model, we can explore a more plausible representation of the optimal arousal level.

### References

Anderson, J. R., Betts, S., Bothell, D., Hope, R., & Lebiere,

 $<sup>^{3}</sup>$ The activation was manipulated by parameters of "chunk creation time" in ACT-R. The chunk creation time for for the current task was set to 800, where as that for default-mode thinking was set to 300.

C. (2019). Learning rapid and precise skills. *Psychological review*, *126*(5), 727.

- Atashfeshan, N., & Razavi, H. (2017). Determination of the proper rest time for a cyclic mental task using ACT-R architecture. *Human Factors*, 59(2), 299–313.
- Baird, B., Smallwood, J., Mrazek, M. D., Kam, J. W. Y., Franklin, M. S., & Schooler, J. W. (2012). Inspired by distraction: Mind wandering facilitates creative incubation. *Psychological Science*, 23(10), 1117–1122.
- Billman, G. E. (2020). Homeostasis: the underappreciated and far too often ignored central organizing principle of physiology. *Frontiers in Physiology*, 200.
- Cannon, W. B. (1929). Organization for physiological homeostasis. *Physiological Reviews*, 9(3), 399–431.
- Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience*. New York: Harper & Row.
- Dancy, C. L., Ritter, F. E., Berry, K. A., & Klein, L. C. (2015). Using a cognitive architecture with a physiological substrate to represent effects of a psychological stressor on cognition. *Computational and Mathematical Organization Theory*, 21(1), 90–114.
- Easterbrook, J. A. (1959). The effect of emotion on cue utilization and the organization of behavior. *Psychological review*, 66(3), 183.
- Fitts, P. M. (1964). Perceptual-motor skill learning. In *Categories of human learning* (pp. 243–285). Elsevier.
- Friston, K. (2010). The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, *11*(2), 127–138.
- Gunzelmann, G., Byrne, M. D., Gluck, K. A., & Moore Jr, L. R. (2009). Using computational cognitive modeling to predict dual-task performance with sleep deprivation. *Human Factors*, 51(2), 251–260.
- Gunzelmann, G., Gluck, K. A., Moore Jr, L. R., & Dinges, D. F. (2012). Diminished access to declarative knowledge with sleep deprivation. *Cognitive Systems Research*, *13*(1), 1–11.
- Hebb, D. O. (1955). Drives and the cns (conceptual nervous system). *Psychological review*, 62(4), 243.
- Killingsworth, M. A., & Gilbert, D. T. (2010). A wandering mind is an unhappy mind. *Science*, 330(6006), 932–932.
- Kim, J. W., & Ritter, F. E. (2015). Learning, forgetting, and relearning for keystroke-and mouse-driven tasks: Relearning is important. *Human–Computer Interaction*, *30*(1), 1–33.
- Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671–680.
- Kotseruba, I., & Tsotsos, J. K. (2018). 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*.
- Maehigashi, A., Miwa, K., Terai, H., Kojima, K., & Morita, J. (2013). Experimental investigation of calibration and resolution in human-automation system interaction. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 96(7), 1625–1636.

- 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.
- Oxendine, J. B. (1970). Emotional arousal and motor performance. *Quest*, *13*(1), 23–32.
- Ritter, F. E. (2009). Two cognitive modeling frontiers emotions and usability. *Information and Media Technologies*, 4(1), 76–84.
- Shenhav, A., Botvinick, M. M., & Cohen, J. D. (2013). The expected value of control: an integrative theory of anterior cingulate cortex function. *Neuron*, 79(2), 217–240.
- Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT Press.
- 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.
- van Vugt, M. K., Taatgen, N. A., Sackur, J., Bastian, M., Borst, J., & Mehlhorn, K. (2015). Modeling mindwandering: a tool to better understand distraction. In *Proceedings of the 13th international conference on cognitive modeling* (pp. 252–257).
- Yerkes, R. M., & Dodson, J. D. (1908). The relation of strength of stimulus to rapidity of habit-formation. *Journal* of Comparative Neurology and Psychology, 18(5), 459– 482.

# **Estimating Phonological Awareness with Interactive Cognitive Models: Feasibility Study Manipulating Participants' Auditory Characteristics**

# Jumpei Nishikawa (nishikawa.jumpei.16@shizuoka.ac.jp)

Junya Morita (j-morita@inf.shizuoka.ac.jp)

Department of Information science and Technology, Graduate School of Science and Technology, Shizuoka University, 3-5-1, Johoku, Naka-ku, Hamamatsu, Shizuoka, 432-8011, Japan

#### Abstract

The difficulties encountered by children during language development varies among individuals. In particular, immaturity in phonological awareness, which supports speech perception, results in various speech defects. Accordingly, it is important to estimate the individual mechanism behind these problems to ensure proper support. In this study, we propose a method for estimating individual defects in the phonological process using cognitive models. As a preliminary step to targeting phonological processing difficulties in real world, we conducted an experiment with native adult speakers. Audio filters were applied to the output of the system to simulate phonological difficulties. This initial feasibility study revealed consistency in model preferences among participants when a particular audio filter was used. We consider that this study provides an important step toward the realizations of individualized cognitive modeling for mitigating various difficulties in language acquisition.

Keywords: ACT-R, Cognitive modeling, Phonological awareness, Individualized model

#### Introduction

Children (or second-language learners) face various difficulties during language acquisition. A prominent example is the segmentation of phonemes. In the early stages of language development, children perceive speech sounds as continuous but can gradually segment them into smaller units (Carroll, Snowling, Stevenson, & Hulme, 2003). In the process, sound can be segmented into various units (symbols), such as syllables and morae. As learners advance, they converge on a system of processing a series of units (e.g., mora in Japanese; Kubozono, 1989), as defined by their native language.

In the fields of developmental psychology and speechlanguage pathology, one of the abilities supporting this development is phonological awareness, which involves paying attention to phonological aspects of speech, such as phonemes and rhythm (Stahl & Murray, 1994). Some speech errors that occur during language development are attributed to a poorly formed phonological awareness of that particular language (Dynia, Bean, Justice, & Kaderavek, 2019; Kobayashi, 2018; Smith Gabig, 2010). In children with autism spectral disorder (ASD), an overall delay in phoneme acquisition and a partial inability to use some phonemes may occur (Grandin & Panek, 2013; Mugitani et al., 2019). To effectively support the formation of abilities that vary greatly among individuals, it is important to consider the cognitive characteristics of the individual child.

This study is a part of the studies aiming to develop a method of constructing a cognitive model adapted to the individual's phonological problem. Specifically, the current study is based on a model of Japanese phonological awareness (Nishikawa & Morita, 2022). The representation of phonological awareness is based on a general memory retrieval mechanism implemented in a cognitive architecture, Adaptive Control of Thought-Rational (ACT-R: Anderson, 2007), which is a framework for developing different models adapted to specific tasks and individuals.

The present study proposes a method of selecting a cognitive model and fitting it to individual phonological problems by varying the parameters in a previous model. The method is validated through an experiment in which participants' auditory traits are artificially manipulated. Finally, we discuss the feasibility of using the proposed method to estimate the users' state of phonological awareness in a summary of the experimental results.

# **Related Research**

In this section, we introduce the literature relevant to the method used this study. We first present reports from clinical and experimental research on phonological awareness. Next, we introduce cognitive models of phonological awareness and previous research on tracing individual cognitive processes using cognitive models.

#### **Research on Phonological Awareness**

Several clinical reports and investigations have utilized experimental methods to investigate phonological awareness formation. Cases of children confusing certain morae have been reported in clinical speech–language pathology practice. Grandin reported that she could not distinguish silent consonants well in her childhood (cat, pat, and hat sounded like the same word) and stated that a child who can only utter vowels (consonant deletion) likely does not hear the consonants (Grandin & Panek, 2013).

In Japanese language, two- or three-year-old infants reportedly tend to confuse morae containing the consonants /r/ and /d/ (Kobayashi, 2018). This erroneous speech pattern diminishes when they reach four- or five-years. However, such phonological discrimination is sometimes delayed. A Japanese textbook (Oishi, 2016) for speech–language pathologists states that children with developmental disorders have difficulty distinguishing between vowels and vowel– consonant combinations with the same vowel (e.g., "a" and "ka"). These reports exhibit commonalities with the previ-

ously mentioned English reports (Grandin & Panek, 2013). Based on these reports, the current study focuses on phonological awareness in the development of Japanese language skills.

Several Japanese studies on phonological awareness have used the popular word game Shiritori as a task. Shiritori involves players taking turns uttering a word (noun); the word must begin with the mora that the previous word ended with. For example, after a player answers "ri-n-go" (meaning apple), the next player continues with "go-ma" (meaning sesame seeds). Takahashi (1997) examines the relationship between the stages of phonological awareness formation and the conditions necessary to playing Shiritori through a psychological experiment in cross-sectional development in children with typical development. Takahashi has shown that phonological awareness (especially the ability to segment sounds into morae and a mental lexicon indexed by morae) is a prerequisite for Shiritori. Takahashi also suggests that playing word games common to a specific culture, such as Shiritori, is important to the growth of phonological awareness in the mother tongue. Building upon this research, we utilize Shiritori as a task to be applied to the phonological awareness model.

#### **Cognitive Modeling of Phonological Awareness**

As noted in the first section, a cognitive model that focuses on human internal processes in the formation of phonological awareness exists (Nishikawa & Morita, 2022). This model assumes innate and experiential constraints of language acquisitions based on parameters implemented in ACT-R. In their model, from the viewpoint of generative phonology (Chomsky & Halle, 1968), innate factors are associated with sound similarities between Japanese morae. Based on this assumption, a partial match mechanism of ACT-R retrieves erroneous phonological knowledge, and it exhibits commonalities with the reports regarding the phonological awareness formation process (Kobayashi, 2018; Oishi, 2016). In addition, the model assumes that such errors derived from an innate factor can be mitigated by an experiential factor, with repetitive practice strengthening correct phonological knowledge.

Although the above study suggests the possibility of describing error patterns in a unified cognitive architecture, there are limitations in the number of error patterns and their practical correspondence to actual individuals. In contrast, many cognitive modeling researchers are trying to represent various individual differences (Smith, Chiu, Yang, Sibert, & Stocco, 2020; Somers, Oltramari, & Lebiere, 2020; Mätzig, Vasishth, Engelmann, Caplan, & Burchert, 2018). These studies constructed models with varying cognitive architecture parameters to fit target individuals.

Such models have been utilized in studies on support systems involving real humans as users. Model-based systems have been developed in the same studies to identify the current state of individual users to guide their activities (Anderson, Boyle, & Reiser, 1985; Klaproth et al., 2020;

### Table 1: Model declarative memory

(a) Word knowledge		(b) edg	) Phonol ge	ogical know	wl-	(c) Word ship	l–mora re	elation-
word	sound		mora	sound		word	mora	position
ringo	"r <sup>j</sup> iŋgo"	-	/ri/	"r <sup>j</sup> i"		ringo	/go/	tail
goma	"goma"		/go/	"go"		goma	/go/	head
riku	"r <sup>j</sup> ikɯ <sup>β</sup> "		/ku/	"kɯ <sup>β</sup> "		goma	/ma/	tail

Morita, Hirayama, Mase, & Yamada, 2016; Morita et al., 2022). For example, the model-based reminiscence method by Morita et al. (2016) extends one of the existing mental health care methods for dementia patients. This study attempts to guide appropriate memory recollection by incorporating a cognitive model corresponding to individual users into a system of a photo slide show for reminiscence.

### **Individual Models of Phonological Awareness**

Building upon previous studies, the current study develops a method of estimating users' internal state by utilizing their interactions with cognitive models. To achieve this goal, this section describes the cognitive model of phonological awareness proposed by Nishikawa and Morita (2022) as the base model for fitting individual users.

#### **Phonological Awareness Model**

Nishikawa and Morita (2022) targeted the phonological awareness observed during a *Shiritori* game. In this paper, we only present the basic model functions necessary to achieving an individualized cognitive model and the settings for individualization. For details, please refer to the original article.

**Knowledge Representation Required for** *Shiritori* This model realizes *Shiritori* based on the general implementation method of the model using ACT-R. That is, declarative knowledge is expressed in *chunks*, and the *Shiritori* procedure is represented by the application of *production rules* that manipulate the ACT-R modules. In the following, we show these two types of knowledge representation in the model.

- **Declarative chunks** The model in this study retains three types of declarative chunks that relate to word (vocabulary), phonological (mora) knowledge, and the association between them (Table 1). These three types of chunks can be regarded as a network, consisting of the word chunk (chunk type (a) in Table 1) and the mora chunk (chunk type (b) in Table 1) nodes and the paths connecting them (Table 1 (c)).
- **Production rules** When the model receives a word as the partner's answer, it traverses the network of declarative chunks to search for a word that follows the rules of *Shiritori*. This process is performed by applying the model's production rules (procedural knowledge) in the following steps.
  - 1. Using the word chunk (chunk type (a) in Table 1) acquired by the aural module, the model retrieves a chunk

that connects the word and the ending mora (chunk type (c) in Table 1).

- 2. Using the retrieved word–mora association knowledge, the phonological knowledge (chunk type (b) in Table 1) corresponding to the word ending is retrieved.
- 3. Using this phonological knowledge, the word that begins with the mora is then retrieved, and the selected word is held as a candidate answer in the goal module.
- 4. Afterward, the model checks that the stored answer candidate is valid according to the rules of *Shiritori*, such as not having been previously answered in the current *Shiritori* trial.
- (a) If the current candidate violates these rules, the model again searches for a candidate answer.
- (b) When the candidate word is confirmed as valid, the model stores it in the declarative module as an answered word and outputs the word through the speech module.

**ACT-R Parameters for Knowledge Retrieval** In the process described above, the phonological awareness involved in paying attention to word endings corresponds to the retrieval of phonological knowledge from word knowledge. Knowledge retrieval in ACT-R is controlled by a parameter called activation that is assigned to each chunk, and it affects the success or failure of the retrieval. The values are computed as the sum of several terms, such as learning effects, contextual effects, similarity between chunks, and a noise term that gives stochastic fluctuations to the activation values.

The similarity term  $(P_i)$  is noteworthy in the model's representation of phonological awareness. As mentioned earlier, Nishikawa and Morita (2022) incorporated innate bias into the knowledge similarity between mora chunks.

$$P_i = PM_{ki} \tag{1}$$

This value is computed as the summation of the weighted degree of similarity  $M_{ki}$  for each retrieval request k to chunk i.  $M_{ki}$  is typically negative, and P serves as a penalty during similarity retrieval. In addition, the partial matching following the introduction of similarity makes it possible to reproduce flexible choices and certain types of errors.

### **Diversified Models**

We extend the phonological awareness model constructed in the previous study (Nishikawa & Morita, 2022) to account for the different problems in phonological awareness. One of the elements to be manipulated to construct an individualized model is the method of computing similarity between the morae. We prepared several *similarity tables* in this study for defining a method of calculating the similarities between morae.

It also manipulates the coefficient corresponding to the magnitude of the influence in the similarity table (P in Eq. 1). It is expected that the similarity table and coefficient P will result in a high level of similarity between certain mora



Figure 1: Concept of Individual Models and Tasks

pairs, which will allow us to address the real-world phenomena (i.e., confusion of /r/ and /d/, consonant deletion, etc.).

# Shiritori Game System for Model Selection

Figure 1 conceptualizes multiple models and *Shiritori* tasks. To confirm that the models constructed in the previous section can capture an individualized phonological process, we set up a task in which the participants play *Shiritori* with the models. This section describes the system developed to perform the task.

# **User Interface**

Figure 2 shows the user interface of the system. In this system, a word-choice-based Shiritori game is set as a task. The user responds by selecting the appropriate word from the candidate answers proposed by multiple models. This response format is based on the Shiritori used in phonological awareness studies (e.g. Takahashi, 1997).

#### **Procedure of the Model Selection**

Figure 3 shows the flow of system usage comprising the following six procedures.

- 1. First, the system presents the starting word (In Figure 2, "せみぶろ") to a set of individualized models and users by playing audio from the experimental window (Figure 3①).
- 2. Each model recognizes the starting word (Figure 3(2)) and its ending, and it answers according to *Shiritori* rules (Figure 3(3)).
- 3. The words answered by the model are displayed in the experiment window (Figure 3(4)) and serve as choices for the user to select as an answer.
- 4. The user answers (chooses) a word they deem appropriate based on the starting word and candidate words (Figure 3(5)). Here, selecting a word is equivalent to selecting the model that proposed the word.
- 5. When the system receives the user's response, it records the model that proposed the chosen answer (Figure 36).
- 6. After a series of processes, the user's answer is used as the next starting word, and the game is repeated (Figure 3⑦).



Figure 2: User interface of the system. The upper, middle, and bottom part of the screen show the question (a speaker icon), choices (robot icons and balloons), and game history, respectively. This screen shows that the user has selected a word from the green model for the fourth *Shiritori* word. The red strings are shows for explanatory purpose.

The above process is reiterated, and the best-matched model is ultimately selected according to the frequency of choice.

# Experiment

This section describes the experiment for testing the proposed method of estimating individual phonological processes by selecting cognitive models. The concept behind the experiment is shown in Figure 4. Because this experiment is an initial feasibility study, adult Japanese native speakers were placed in situations where the phonological process was artificially generated. In each turn, an audio filter receives the model (a word) to generate phonological processing difficulties for participants. In this setting, we test the following hypothesis: different audio filters produce different model preferences in a word-choice-based *Shiritori* task.







Figure 4: Concept behind the Experiment

# Method

**Experimental Design** The participants' behaviors were compared by manipulating model parameters and audio filters. The specifics of the manipulation are as follows:

- **Model Settings** Four models were prepared, as indicated in Table 2, and the following two factors were considered.
  - **Similarity table** This factor indicates the difference in computing similarity between morae. We used the Consonant–Vowel concatenation table (C–V concatenation table) and the Consonant–Vowel average table (C–V average table), both were presented by Nishikawa and Morita (2022).
  - **Similarity coefficient** This indicates the degree of error suppression caused by morae similarity (P = 10, 30 in Eq. 1).
- Filter settings We prepared two audio filter settings [+10] and -10 filter conditions]. These filters indicate the formant setting of Voice Transformer, which is a plug-in effect of Apple GrageBand (MacOSX 10.x). Negative and positive formants transform input voice into deep/muffled and high-pitched/thin tones, respectively.

**Participants** One male graduate student and one female undergraduate student participated in the experiments. Both were native Japanese speakers majoring in informatics. Henceforth, the two participants will be referred to as Participant A and Participant B.

	C–V concatenation table	C–V average table
similarity coefficient $P = 10$	P10-SIM1	P10-SIM2
similarity coefficient $P = 30$	P30-SIM1	P30-SIM2

Table 2: Individual model settings and their notaions

Table 3: Experimental procedures and conditions for each participant

		Participant A	Participant B
Instruction	(5 min)		
Shiritori 1	(25 min)	+10 condition	-10 condition
Break	(5 min)		
Shiritori 2	(25 min)	-10 condition	+10 condition
Questionnaire	(5 min)		

**Apparatus** The system was displayed on a external monitor connected to the laptop (Apple MacBookPro M1 2020, macOS Big Sur) and operated with a built-in touchpad. Google Cloud Text-to-Speech API was used for the audio output of the system. This was then input to GarageBand through a virtual audio driver called BlackHole. The audio input to GarageBand was distorted by a large formant shift, as described above. The affected audio was output from the speakers through BlackHole from the GarageBand monitor function.

**Procedure** The flow of the experiment is presented in Table 3. First, the participants were seated in front of the display showing the system, and the experiment objectives and how to use the system were orally explained. After, the participants confirmed that they had no questions, they performed the *Shiritori* task. Each condition was allocated 25 min for one *Shiritori* task, and the participants answered a questionnaire after completing two *Shiritori* tasks. The two *Shiritori* tasks were performed using different audio filter settings. Participant A was subjected to the +10 condition, followed by the -10 condition. Participant B was subjected to the -10 and +10 conditions in that order.

### Results

In this section, we analyze the effect of the audio filter by tabulating the model whose answers were selected by the participants.

**Number of model selections.** Figure 5 (a) is a stacked bar graph showing the model chosen by the participant. The four bars denote the different combinations of participants and filter conditions. The vertical axis corresponds to the total number of selections without considering the correctness of the model answers. Although the bars have different heights, we can find commonalities in all conditions. Regardless of the difference between participants/filters, the P30-SIM2 (thick green regions) was the most frequently selected, whereas the

P10-SIM1 (light blue regions) was chosen the least often.

To confirm such commonalities across the conditions, we constructed a correlation matrix for all combinations of the conditions (Figure 5 (b)). The Spearman correlation coefficients were computed, treating each model as a unit of analysis (n = 4). The figure indicates high correlations in all cells in the matrix, suggesting that frequently selected models were common across participants and filters.

This result agrees with the performance of the original models. As in Eq. 1, the large P suppresses errors derived from the similarities among morae. Regarding the similarity table, Nishikawa and Morita (2022) suggested that a high error rate in the C–V concatenation table is representative of consonant deletions that are similar to those that occur in children with ASD (Grandin & Panek, 2013; Mugitani et al., 2019).

Despite the above overall commonalities among conditions, slight differences exist in the correlation between condition pairs. In particular, Participants A's +10 filter condition has relatively weak correlations with the other filter (-10) conditions (Participant A's -10 filter condition: r = .73, p > .10; Participant B's -10 filter condition: r = .80, p > .10). These results validate our assumption, revealing that different audio traits lead to different preferences for phonological models.

Number of incorrect selections. Figure 6 (a) is the stacked bar graph showing the number of chosen models limited to the incorrect answer. Unlike in the previous graph, large differences exist between participants/filters in the graph. The same is confirmed in Figure 6 (b). The figure reveals a significant correlation (r = 0.97, p < .05) between participants in the +10 condition. No significant correlation exists among the other conditions. These results suggest that under the +10 filter, the preference for the models was consistent across the participants.

# Discussion

In this experiment, we tested the hypothesis that different audio filters (artificially generated individual differences in auditory traits) produce different model preferences. To this end, we compared the selection of the models under the different participants and audio filters. As shown in Figure 6 (b) there was a significant correlation for only the +10 condition among the participants, whereas no significant correlation was observed for the other combinations. In other words, when the +10 filter is applied, there is some match among participants regarding the ease of model selection. Therefore, the tested hypothesis has some validity.



(a) Number of participant choices for each model.



(b) Correlation matrix of the number of model selections (each bar in Figure 5a) across conditions.

Figure 5: Model selection by participants

# **Summary and Future Work**

We proposed a method for fitting a cognitive model to individual phonological problems. We prepared cognitive models for a *Shiritori* game and assigned participants to play a wordchoice-based *Shiritori* game. By the participants repeatedly selecting the words proposed by the model, we could estimate a model that structurally represents the participants' phonological awareness. We evaluated the feasibility of this method in an experiment with adult native speakers. The feasibility experiment was designed to simulate phonological processing difficulties by applying an audio filter to the words proposed by the model.

In this experiment, we tested the hypothesis that different audio filters produce different model preferences in a wordchoice-based *Shiritori* task. The experiment involving two participants revealed a significant correlation in model selection among the participants under certain audio filter conditions. This means that there is consistency among the participants in the model that is more or less likely to be selected as the incorrect choice (not appropriate as a *Shiritori* answer). Accordingly, the experiment hypothesis had some validity.

This research ultimately aimed to develop a phonological awareness support system that utilizes cognitive models that are adapted to individual error patterns and the individ-



(a) The number of choices of the model when a participant incorrectly selected.



(b) Correlation matrix of the number of incorrect model selections (each bar in Figure 6a) across conditions.

Figure 6: Incorrect model selection by participants.

uals themselves to estimate the users' phonological awareness. Accordingly, the results should be analyzed further. In this paper, we only analyzed the number of answers and the frequency of model selection by focusing on wrong answers in *Shiritori*. However, it is necessary to rigorously confirm the corresponding effects by performing tests with statistical methods.

Moreover, the method should be expanded. The extent of parameter exploration in the system presented in this study is limited. Only four models were prepared and selected by the participants. In the future, we intend to construct a method of automatically generating models by combining parameters related to phonological awareness. Experiments under such dynamic conditions are also necessary. After sufficient feasibility studies have been conducted, we will conduct an experiment involving learners, who are the original target of the method, such as children with phonological awareness problems and second-language learners.

### References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press.

- Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring systems. *Science*, 228(4698), 456–462.
- Carroll, J., Snowling, M., Stevenson, J., & Hulme, C. (2003). The development of phonological awareness in preschool children. *Developmental Psychology*, 39(5), 913–923.
- Chomsky, N., & Halle, M. (1968). *The sound pattern of english*. Harper & Row New York.
- Dynia, J. M., Bean, A., Justice, L. M., & Kaderavek, J. N. (2019). Phonological awareness emergence in preschool children with autism spectrum disorder. *Autism & Devel*opmental Language Impairments, 4, 2396941518822453.
- Grandin, T., & Panek, R. (2013). *The autistic brain: Thinking across the spectrum*. Houghton Mifflin Harcourt.
- Klaproth, O. W., Halbrügge, M., Krol, L. R., Vernaleken, C., Zander, T. O., & Russwinkel, N. (2020). A neuroadaptive cognitive model for dealing with uncertainty in tracing pilots' cognitive state. *Topics in cognitive science*, 12(3), 1012–1029.
- Kobayashi, H. (2018). Oninishiki no keisei to kotoba no hattatsu - "kotoba ga osoi" wo kangaeru-. Kodama shuppan.
- Kubozono, H. (1989). The mora and syllable structure in japanese: Evidence from speech errors. *Language and Speech*, *32*(3), 249–278.
- Mätzig, P., Vasishth, S., Engelmann, F., Caplan, D., & Burchert, F. (2018). A computational investigation of sources of variability in sentence comprehension difficulty in aphasia. *Topics in cognitive science*, 10(1), 161–174.
- Morita, J., Hirayama, T., Mase, K., & Yamada, K. (2016). Model-based reminiscence: Guiding mental time travel by cognitive modeling. In *Proceedings of the fourth international conference on human agent interaction* (pp. 341– 344).
- Morita, J., Pitakchokchai, T., Raj, G. B., Yamamoto, Y., Yuhashi, H., & Koguchi, T. (2022). Regulating ruminative web browsing based on the counterbalance modeling approach. *Frontiers in Artificial Intelligence*, *5*.
- Mugitani, R., Homae, F., Hiroya, S., Satou, Y., Shirose, A., Tanaka, A., ... Tachiiri, H. (2019). *Kodomo no onsei* (No. 21). Corona Publishing Co., Ltd.
- Nishikawa, J., & Morita, J. (2022). Cognitive model of phonological awareness focusing on errors and formation process through shiritori. *Advanced Robotics*, *36*(5-6), 318-331.
- Oishi, N. (2016). Evaluation. In H. Ishida & I. Ishizaka (Eds.), Gengochokakushi no tameno gengohattatsushogaigaku (2nd ed., pp. 77–117). Ishiyaku shuppan.
- Smith, B., Chiu, M., Yang, Y., Sibert, C., & Stocco, A. (2020, 07). Modeling the effects of post-traumatic stress on hippocampal volume..
- Smith Gabig, C. (2010). Phonological awareness and word recognition in reading by children with autism. *Communication Disorders Quarterly*, 31(2), 67–85.
- Somers, S., Oltramari, A., & Lebiere, C. (2020). Cognitive twin: A personal assistant embedded in a cognitive architecture. In *Proceedings of ICCM 2020, 18th international*

conference on cognitive modelling.

- Stahl, S. A., & Murray, B. A. (1994). Defining phonological awareness and its relationship to early reading. *Journal of educational Psychology*, 86(2), 221–234.
- Takahashi, N. (1997). A developmental study of wordplay in preschool children: The japanese game of "shiritori". *The Japanese Journal of Developmental Psychology*, 8(1), 42–52. (In Japanese)
# Minerva-Q: A Multiple-Trace Memory System for Reinforcement Learning

Renan Ozen (renanozen@cmail.carleton.ca) Robert L. West (robert.west@carleton.ca) M. Alex Kelly (alex.kelly@carleton.ca) Department of Cognitive Science, Carleton University 1125 Colonel By Drive, Ottawa, Ontario, K1S 5B6 Canada

#### Abstract

We propose Minerva-Q, a multiple-trace memory model capable of perceptual-motor reinforcement learning. This model combines Q-learning with the Minerva family of memory models. In our simulations we found our Minerva-Q agent learned increasingly optimal solutions to the Cart Pole task and reproduced human-like performance when presented with minimal expert training examples in a sparse reward task.

**Keywords:** Minerva; reinforcement learning; memory; perception; motor; artificial intelligence; Q-learning

#### Introduction

Perceptual-motor learning can be observed in the behaviours of many animal species and may be responsible for the development of embodied skills including tool use, spatial navigation, hunting and foraging, or in humans, skills such as riding a bicycle or playing an instrument. Learning more broadly is known to require some form of memory and entails adaptive processes to reinforce, inhibit, or alter information stored in memory. We may thus posit a minimal perceptualmotor learning agent as consisting of a set of adaptive learning and action-selection processes and a set of sensory inputs and motor outputs connected to a memory store. In the fields of cognitive science, artificial intelligence, and robotics, we hold that it is useful to consider and study such minimal configurations, as most living organisms do not possess the "higher" cognitive skills of humans such as those related to semantic reasoning, yet non-human organisms are capable of completing tasks our best models and intelligent systems are either incapable of handling at present or may require many orders of magnitude more computational resources to perform. Moreover, human motor learning may be more akin to this type of system.

Towards this end, we propose Minerva-Q, a minimal memory model capable of perceptual-motor skill acquisition via an implementation of Q-learning (Watkins, 1989), which is a form of temporal-difference (TD) learning algorithm motivated by classical conditioning theories of animal skill acquisition. "Minerva" in Minerva-Q refers to a family of multiple-trace memory models largely based on MINERVA 2 (Hintzman 1984; Hintzman, 1986), which have been used to model associative learning (Jamieson et al., 2012) and decision-making (Dougherty et al., 1999), among a large set of other cognitive processes (see Jamieson et al., 2022). Given the versatility of Minerva models, we hypothesized that a Minerva-like model of perceptual-motor memory could be applied to the domain of reinforcement learning (RL). We found that our Minerva-Q RL agent could solve a dense-

reward and a sparse-reward RL task and observed its humanlike capacity to learn from a minimal number of expert examples. This latter feature sets Minerva-Q apart from models like the so-called "deep Q-network" (Mnih et al., 2015) that may require hundreds of training examples (or more) to reproduce expert task performance.

In this paper, we first describe the specifications of the Minerva family of memory models that set precedence for Minerva-Q. Next, we provide a detailed account of the structure and mechanisms of Minerva-Q. Finally, we outline the results of a Minerva-Q agent in two simulated RL tasks and discuss the wider implications of our model to the cognitive sciences.

# MINERVA 2

In MINERVA 2, each experience is stored as a separate item in memory, known as a memory trace (hence, multipletrace memory). More specifically, MINERVA 2 consists of two memory subsystems: primary memory (PM), which is a limited temporary memory store analogous to working memory, and a long-term secondary memory (SM) containing all memory traces. Information that passes into PM is sent to SM as a "probe," which returns a single "echo" to PM. This echo represents a retrieved memory instance and is constructed during each retrieval as the sum of activated traces. Thus, retrieval cannot be understood as a lookup process, and the addition of identical traces to SM has the effect of strengthening the influence of these traces during activation.

Memory traces in MINERVA 2 are represented as integervalued vectors with random values  $v \in \{-1,0,1\}$ . These traces are stored in an  $m \times n$  matrix M, where m is the number of memory traces and n is the number of values in each trace. In MINERVA 2, the similarity s between a probe p and a memory trace is computed as a normalized dot product,

$$s_{\rm i} = \frac{1}{n_{\rm R}} \sum_{j=1}^{\rm n} p_j M_{\rm i,j}$$
 (1)

where  $n_{\rm R}$  is the maximum number of nonzero values in the probe or memory trace  $M_i$ . This similarity metric is cubed to increase the signal-to-noise ratio in the echo, producing an activation vector a of the trace given the probe:

$$a_{i} = s_{i}^{3} \tag{2}$$

Finally, the echo *c* may be computed as the sum of activated traces:

$$c_{j} = \sum_{i=1}^{m} a_{i} M_{i,j} \tag{3}$$

Learning in MINERVA 2 is probabilistic, where each nonzero feature in the probe is added to memory with probability *L*. Forgetting is treated as the opposite of learning, where each nonzero feature has a probability *F* of being set to 0. According to Hintzman (1986), learning with L = 0.25 is equivalent to learning with L = 1.00 and forgetting with F = 0.75.

# **MINERVA 2 Variants**

Variants of MINERVA 2 typically commit to Hintzman's (1984; 1986) assumptions but may propose different functions with respect to encoding, activation, learning, and forgetting to model their targeted phenomena. For example, Jamieson et al. (2018) offer a Minerva variant that combines the retrieval operations of MINERVA 2 with the encoding scheme of BEAGLE (Jones & Mewhort, 2007), and compute activation under a cosine similarity function. Likewise, Collins et al. (2020) formalize a probabilistic learning mechanism that prioritizes surprise, conceptually grounded in the discrepancy encoding of Jamieson et al. (2012).

These examples set precedence for many of the formalizations developed for Minerva-Q, which we argue should be considered a Minerva-*like* model as it retains the core operations of MINERVA 2 but departs somewhat from Hintzman's (1984) theoretical assumptions about memory.

# Minerva-Q

#### **Q-Learning**

Minerva-Q implements Q-learning (Watkins, 1989) to handle perceptual-motor tasks. Q-learning is a model-free reinforcement learning method. More formally, Q-learning, when implemented in a table of Q-values, converges on an optimal action-selection policy for finite, discrete-time Markov decision processes (Watkins & Dayan, 1992). At each time step t, a Q-learning agent has the task of selecting some action  $A_t \in A$  given some state  $S_t \in S$ , where S is a finite set of discrete states and A is a finite set of discrete actions. After taking said action, the agent receives a reward r, which is used to update its policy according to the formula,

$$Q_{\text{new}}(S_t, A_t) \leftarrow (1 - \alpha)Q(S_t, A_t)$$

$$+ \alpha \left(r + \gamma \max(Q(S_{t+1}, A))\right)$$
(4)

where  $Q(S_t, A_t)$  (the Q-value) is the expected discounted reward for selecting action  $A_t$  given state  $S_t$  under the current policy,  $\gamma$  is a discount factor that evaluates rewards received earlier as higher than those received later (if  $\gamma < 1$ ) by a factor of  $\gamma^t$ , and  $\alpha$  is a learning rate which adjusts the impact of Q-value updates at each time step.

#### Retrieval

At each time step during a task, a state vector is passed into Minerva-Q's PM which is then encoded into a probe for echo retrieval from SM. Memory traces in SM are a concatenation of a state vector and  $n_A$  Q-values (the "action vector"), where  $n_A$  is the number of discrete actions the Minerva-Q agent can perform during the task. SM in Minerva-Q is thus represented as an  $m \times (n_S + n_A)$  matrix M, where m is the number of stored traces and  $n_S$  is the dimension of the encoded probe p.

Minerva-Q differs from other Minerva models in that the dimension of its probes is not equal to the dimension of its traces. This is handled by activating only the part of each trace vector corresponding to the probe, where the activation vector a is computed under a cubed cosine similarity function:

$$a_{i} = \left(\frac{\sum_{j=1}^{n_{S}} p_{j} M_{i,j}}{\sqrt{\sum_{j=1}^{n_{S}} p_{j}^{2}} \sqrt{\sum_{j=1}^{n_{S}} M_{i,j}^{2}}}\right)^{3}$$
(5)

We may justify this decision with the reasoning that Qvalues do not exist as objects in the environment, rather they are internal representations of the expected values of particular actions and therefore should not be considered in similarity measures between stored memory traces and what the agent is experiencing (or observing). Under this interpretation, we see the probe as fundamentally tied to perception, or more specifically to the observation and transduction of a state vector, and activation as an associative memory process, the latter position informed by Hintzman (1990). It may therefore be more helpful to understand the stored Q-values in the action vector as metadata attached to an observed state that influences action-selection rather than playing a role in associative perceptual memory processes. Further, we note precedence for this kind of partial activation in other Minerva models (e.g., Johns et al., 2016).

Given this interpretation, trace activation in Minerva-Q still must be understood as influencing all information in a stored memory trace, thus the retrieved echo vector c has a dimension of  $n_S + n_A$ , and is computed like in MINERVA 2:

$$c_{\rm i} = \sum_{j=1}^{n_S + n_A} a_j M_{\rm i,j} \tag{6}$$

#### **Perceptual-Motor Learning**

A Minerva-Q agent learns as follows. First, an initial state observation is passed into PM and copied into the bottom slot of a Q-buffer, which is a matrix B with a dimension of  $2 \times (n_S + n_A)$  and is initialized to all zeros. For the sake of clarity, we refer to the first row of B as its top slot and the second row as its bottom slot. Note that since the observation

probe has a dimension of  $n_S$ , the latter  $n_A$  dimensions of the top slot are left unchanged. Then, the probe is used to retrieve an echo (all zeros if memory is empty) from SM and the latter  $n_A$  values (the Q-values, or action vector) from the echo are copied into the latter  $n_A$  values of *B*'s bottom slot.

Next, the Minerva-Q agent is tasked with selecting an action. Here we leave the particulars up to modelers, as various heuristics may be implemented across different tasks to balance between exploration of an environment and exploitation of the learned policy (see Amin et al., 2021). However, in our simulations we used a decaying  $\epsilon$ -greedy heuristic which selects actions according to a random uniformly distributed variable  $x \in [0,1]$  and a variable  $\epsilon \in [0,1]$ . At each time step t, x is randomly chosen and an action  $A_t$  is selected according to the strategy,

$$A_{t} = \begin{cases} maxarg([B_{2,n_{S}+1} \dots B_{2,n_{S}+n_{A}}]), \text{ if } x < \epsilon \\ randarg([B_{2,n_{S}+1} \dots B_{2,n_{S}+n_{A}}]), \text{ otherwise} \end{cases}$$
(7)

where the function *maxarg* returns the index of the maximum value of the action vector, and the function *randarg* returns a random index of the action vector. Then,  $\epsilon$  decays at each time step by some fixed amount until it reaches a predetermined minimum value.

Before performing the selected action, the contents of *B*'s bottom slot are copied to its top slot. After performing the action at time *t*, the Minerva-Q agent receives a reward and observes a new state which is then copied to *B*'s bottom slot and used to probe memory to retrieve an echo  $c_{t+1}$ . As before, the action vector portion from  $c_{t+1}$  is copied into *B*'s bottom slot.

At this point, having performed its first action and received a reward r, the Minerva-Q agent can update the Q-value in B's top slot corresponding to  $A_t$  using the formula,

$$B_{1,n_{S}+A_{t}} \leftarrow (1-\alpha)B_{1,n_{S}+A_{t}}$$

$$+ \alpha (r + \gamma \max([B_{2,n_{S}+1} \dots B_{1,n_{S}+n_{A}}]))$$
(8)

where  $\alpha$  and  $\gamma$  represent the same parameters as in equation (4). Next, the action vector in *B*'s top slot is normalized to promote numerical stability, then finally the agent forgets (as explained below), and the top slot is added as a new item to SM without any information loss. From here, the algorithm may loop from the point of action selection.

#### Forgetting

Since there is no formal mechanism by which stored memory traces can be updated in the Minerva framework and thus no obvious way to update Q-values akin to replacing them in a table, Minerva-Q leverages forgetting to probabilistically clear memory traces most similar to what is being added (again, considering only the first  $n_s$  dimensions during activation) allowing for a somewhat noisy yet effective method of updating stored Q-values. This forgetting is implemented as logistic function similar to the discrepancy encoding function of Minerva-DE (Collins et al., 2020) but modified to enable targeted forgetting. Before an item is added to memory, each element of trace  $M_i$  (excluding those containing Q-values) has a probability F of being set to 0. This probability is determined by the trace's activation  $a_i$  to the item being added according to the function,

$$F_{\rm i} = \frac{1}{1 + \mathrm{e}^{-\phi a_{\rm i} + \beta}} \tag{9}$$

where the parameters  $\phi$  and  $\beta$  adjust the slope and bias respectively. Tuning these parameters allows for implementations ranging from those with hard, precise forgetting thresholds to those with softer, more broad targeting to manage memory capacity, as traces containing all 0's in their first  $n_s$  dimensions can safely be removed from memory.

# **Trace Encoding**

As outlined, each instance stored in memory is a concatenation of an observation and a normalized action vector containing Q-values. However, due to the use of vector cosine as a similarity metric in Minerva-Q, observation state spaces with a low dimension might result in suboptimal learning. In the most extreme case, a task providing only a single floating-point number as an observation will restrict the similarity function to the set of possible activations: {-1, 1}. Thus, it may be useful to process (i.e., transduce) an observation prior to its delivery to PM. In our simulations, observations are expanded such that each constituent floating-point value is represented as a bit vector with a dimension of b - 1, where b is the value's number of bits of precision (more accurately, the width of its exponent plus the width of its mantissa). The value's sign bit is not included in this expansion but is used to populate the delivered probe vector with either 1's and 0's or -1's and 0's. This preserves the semantics of the similarity metric (i.e., x and -x will have a similarity of -1.0). Thus, the dimension of each trace in memory in our simulations is  $(b-1) \times n_S + n_A$ .

This encoding method, though effective in our simulations, is likely suboptimal, as our similarity function is not sensitive to magnitude. For example, the similarity between 3.14 and 3.15 in our implementation is 0.58, whereas the similarity between 3.14 and 1.268x10<sup>-308</sup> is 0.60. Intuitively, we might expect numbers closer together to be more similar to each other than to numbers that differ by many orders of magnitude. We encourage future research to develop different representational schemes and/or similarity functions that improve our approach and preserve this expectation. A possible approach for future investigation may involve the use of fractional binding to represent continuous spaces (e.g., Komer et al., 2019).

#### Simulations

For our simulations, we implemented Minerva-Q in Python 3.10.2 using PyTorch (Paszke, et al., 2019). We chose the OpenAI Gym library (Brockman et al., 2016) environments

*CartPole-v1* (Cart Pole) and *MountainCar-v0* (Mountain Car) to simulate our tasks. Notably, OpenAI Gym has been designed specifically for reinforcement learning and is used across the artificial intelligence community as a benchmarking tool.

To improve learning, after each trial (or episode) we compared the agent's results to previous trials and if we found no significant improvement, cleared its memory up to the last best trial (or to the last trial better than some minimum performance threshold). Though this procedure is not mandatory for Minerva-Q to learn satisficing strategies, we decided to include it to demonstrate the potential of using Minerva-Q to rapidly optimize a strategy. We conjecture based on our limited tests that given enough time, Minerva-Q will tend towards increasingly optimal solutions, however this notion requires further corroboration and testing. As precedence, a conceptually similar approach is taken in instance-based cognitive models (see e.g., Gonzalez et al., 2003). Under this frame, our approach may be understood as setting the activation or utility of the memory chunks stored during a suboptimal trial to 0.

# **Cart Pole Task**

The Cart Pole (also known as inverted pendulum) task is a classic control problem that requires an agent controlling a cart to keep upright a pole attached to a joint fixed to the cart's center. In this environment, a reward of 1 is given at each time step. The task terminates if the pole falls past 12 degrees in either direction, when the cart moves past a certain position in either direction, or when the agent reaches the maximum of 500 steps. At each time step, the agent observes the cart position, cart velocity, the pole's angle, and its angular velocity, and must choose between one of two discrete actions: pushing the cart either left or right.

For this task, we set  $\phi$  to 11, and  $\beta$  to 8 in our forgetting function, informed by the parameters used in the Minerva-DE (Collins et al., 2020) simulations. We set  $\alpha$  to 0.8 and  $\gamma$  to 0.99 in our Q function to promote a long-term time-horizon.

At each trial 1, the Minerva-Q agent has zero knowledge of the task stored in memory, and chooses actions based on a decaying  $\epsilon$ -greedy policy. Figure 1 shows our results after 200 trials, averaged over 50 sets of trials;  $\epsilon$  values are shown in red, beginning at 0.99 and decaying to a minimum of 0.02.

Though our results do not show the agent achieving a maximum score of 500, we see the number of steps taken before the task terminates increasing as  $\epsilon$  decays, continuing in a clear upward trend toward trial 200, demonstrating learning.

# **Mountain Car Task**

The Mountain Car task originally appears in Moore (1990) and consists of a car on curved mountain-like onedimensional surface, which is consistent across trials. Starting at a random location at the bottom of a valley, an agent must maneuver the car to reach a goal position at the



Figure 1: Average Cart Pole results across 50 sets of 200 trials. Blue indicates the average number of steps per trial (higher is better); red indicates the average  $\epsilon$  value per trial.

mountaintop by accelerating back and forth across the valley until enough speed is gained to drive up a steep incline.

At each time step, the agent receives an observation describing the car's position along the x-axis and its velocity and must choose between three actions: accelerate to the right, to the left, or do nothing. Interestingly, Moore's solution to this problem, like Minerva-Q, stores each experience in its memory, though implements a much different learning algorithm and overall architecture.

In the Cart Pole task, the agent receives "dense" rewards (i.e., at each time step), but in the Mountain Car task is only rewarded once the car reaches the goal position. This is a subtle yet consequential distinction between the two tasks, as "sparse" reward paradigms like the Mountain Car task may necessitate more exploration of their state space to find rewarding solutions.

Using our memory clearing approach therefore does not make much sense for tasks with sparse rewards, at least until a satisficing solution is found from which a policy may be optimized. Thus, to assess Minerva-Q's performance on this task, we initialized our agent's memory with four expert trials of the task, which provided enough rewarding experience to optimize further in unsupervised trials.

Properly integrating this expert knowledge required a change of the task reward values. By default, this task environment returns a reward of -1 at each time step (up to a maximum of 200 steps, terminating the task) and a reward of 0 upon reaching the goal position. However, since our  $\epsilon$ greedy selection targets the maximum Q-value, we found that the agent actively avoided taking the actions made by expert players, as the combination of sparseness and negative rewards disincentivized selecting these actions. To resolve this, we modified the rewards such that the agent received a maximum reward of 1 at the goal position, otherwise it would receive a reward of 0. This resulted in the expected behaviour of reproducing the expert knowledge. Thus, we may conclude our model exhibits human-like motor learning in its capacity to learn from limited examples, given an appropriate reward structure.

# Discussion

Our results suggest that human-like perceptual-motor learning is possible to model with a minimal set of memory structures and functions, more specifically those of the Minerva family. As acknowledged, there are key differences in our implementation of Q-learning that necessitated an extension of the MINERVA 2 framework, though we hold that our solutions in Minerva-Q provide modest innovations that architectural purists are likely to find acceptable. More broadly, where other Minerva models were able to show successful empirical results for episodic and semantic memory tasks, we showed qualitative successes related to perceptual-motor memory tasks. This may have important theoretical implications to be explored in future works. However, we note some more immediate points for discussion.

First, we want to address the notion of optimization, which at least in the cognitive sciences is less of a priority compared to plausible satisficing strategies. Q-learning is typically implemented using a table of Q-values and in this form is guaranteed to converge on an optimal policy (Watkins, 1989; Watkins & Dayan 1992). More recently, Q-learning has been implemented in a deep Q-network (DQN; Mnih et al., 2015) that achieved human-level performance on 29 Atari games. In DQN, an approximation of the Q-learning policy is learned via iterated backpropagation. In response to stability and divergence issues in using nonlinear function approximators to derive Q-values (see e.g., Baird, 1995; Tsitsiklis & Van Roy, 1997), DQN employs "experience replay" and a novel loss function claimed to mitigate problematic correlations in training data that result in divergence from an optimal policy, though these are imperfect solutions.

There are important structural differences between Minerva and fully-connected artificial neural networks that suggest the divergence issues of DQN may not be relevant to Minerva-Q. According to Hintzman (1990), Minerva models can be understood as nonlinear localist neural networks. More specifically, Kelly et al. (2017) show that MINERVA 2 is equivalent to a distributed Hebbian associative memory. Conversely, neural networks like DON utilize backpropagation via gradient descent on distributed representations and are more formally understood as universal function approximators given a sufficient number of hidden layers (Hornik et al., 1989). Though powerful, this kind of backpropagation is prone to issues like catastrophic interference and is largely responsible for divergence issues when approximating the Q-value function (Baird, 1995; Tsitsiklis & Van Roy, 1997).

Critically, Minerva-Q does not update its Q-values using an approximated function, rather the function is directly implemented to store new Q-values in memory. Despite this advantage over DQN, Minerva models are not equivalent to lookup tables, therefore we cannot assume without further investigation that Watkins and Dayan's (1992) convergence proofs apply to Minerva-Q in its current form. Nevertheless, it is encouraging that Minerva-Q appears to at least satisfice, and its connection to other Minerva models suggest it is a worthwhile effort of cognitive science to study questions of convergence in greater depth.

However, there appears to be, as a first approximation, an isomorphism between the Minerva family of memory models (including Minerva-Q) and the transformer class of neural network architectures. Specifically, we conjecture there are similarities between the so-called "attention" functions of transformer networks (see: Vaswani et al., 2017) and the Minerva activation and echo construction mechanisms. Though this notion is yet to be corroborated, if true, it may help ground currently state-of-the-art transformer models in more psychologically plausible models of memory.

Lastly, we note that the presented Minerva-O model is subject to change, as it is under active development to accommodate theoretical considerations, incorporate more cognitively plausible mechanisms. and improve performance. For example, the current iteration is limited to discrete action spaces and thus cannot handle tasks requiring continuous-valued inputs. Likewise, there may be more plausible action-selection heuristics, such as ones motivated by optimism or surprise. Finally, we note that although our learning optimization approach (i.e., the deletion of suboptimal trials from memory) is not mandatory and has some conceptual precedence, it introduces structural changes that may be theoretically problematic. Thus, future iterations should explore solutions that achieve this behaviour in a more parsimonious and tenable manner.

# References

- Amin, S., Gomrokchi, M., Satija, H., Van Hoof, H., & Precup, D. (2021). A survey of exploration methods in reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.2109.00157
- Baird, L. (1995). Residual algorithms: Reinforcement learning with function approximation. *Machine Learning Proceedings 1995*, 30-37. https://doi.org/10.1016/B978-1-55860-377-6.50013-X
- Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba W. (2016). *OpenAI Gym.* arXiv. https://doi.org/10.48550/arXiv.1606.01540
- Collins, R. N., Milliken, B., & Jamieson R. K. (2020). MINERVA-DE: An instance model of the deficient processing theory. *Journal of Memory and Language*, *115*(2020:104151), 1-13.

https://doi.org/10.1016/j.jml.2020.104151

- Dougherty, M. R. P., Gettys, C. F., & Ogden, E. E. (1999). MINERVA-DM: A memory processes model for judgments of likelihood. *Psychological Review*, 106(1) 180-209. https://doi.org/10.1037/0033-295X.106.1.180
- Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instancebased learning in dynamic decision-making. *Cognitive Science*, 27(2003), 591-635. https://doi.org/ 10.1016/S0364-0213(03)00031-4
- Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359-366. https://doi.org/10.1016/0893-6080(89)90020-8

- Hintzman, D. L. (1984). MINERVA 2: A simulation model of human memory. *Behavior Research Methods*, *Instruments*, & *Computers*, 16(2), 96-101. https://doi.org/10.3758/BF03202365
- Hintzman, D. L. (1986). "Schema abstraction" in a multipletrace memory model. *Psychological Review*, 93(4), 411-428. https://doi.org/10.1037/0033-295X.93.4.411
- Hintzman, D. L. (1990). Human learning and memory: Connections and dissociations. *Annual Review of Psychology*, 41(1990), 109-139. https://doi.org/10.1146/annurev.ps.41.020190.000545
- Jamieson, R. K., Crump, M. J. C., & Hannah, S. D. (2012). An instance theory of associative learning. *Learning & Behavior*, 40(2012), 61-82. https://doi.org/10.3758/s13420-011-0046-2
- Jamieson, R. K., Avery, J. E., Johns, B. T., & Jones, M. N. (2018). An instance theory of semantic memory. *Computational Brain & Behavior*, 2018(1), 119-136. https://doi.org/10.1007/s42113-018-0008-2
- Jamieson, R. K., Johns, B. T., Vokey, J. R., & Jones, M. N. (2022). Instance theory as a domain-general framework for cognitive psychology. *Nature Reviews Psychology*, *I*(2022), 174-183. https://doi.org/10.1038/s44159-022-00025-3
- Johns, B. T., Jamieson, R. K., Crump, M. J. C., Jones, M. N., & Mewhort, D. J. K. (2016). The combinatorial power of experience. In Papafragou, A., Grodner, D., Mirman, D., & Trueswell, J.C. (Eds.) Proceedings of the 38th Annual Conference of the Cognitive Science Society. (pp. 1325-1330). Austin, TX: Cognitive Science Society.
- Jones, M. N., & Mewhort, D. J. K. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, *114*(1), 1-37. https://doi.org/10.1037/0033-295X.114.1.1
- Kelly, M. A., Mewhort, D. J. K., & West, R. L. (2017). The memory tesseract: Mathematical equivalence between composite and separate storage memory models. *Journal* of Mathematical Psychology, 77(April 2017), 142-155. https://doi.org/10.1016/j.jmp.2016.10.006
- Komer, B., Stewart, T. C., Voelker, A. R., & Eliasmith, C. (2019). A neural representation of continuous space using fractional binding. *Proceedings of the 41st Annual Meeting* of the Cognitive Science Society, 2019, 2038-2043.
- Moore, A. W. (1990). *Efficient memory-based learning for* robot control. [Ph.D. thesis]. University of Cambridge.
- Mnih, V., Kavukcuoglu, K., Silver., D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglu, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*, 529-542. https://doi.org/10.1038/nature14236
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Kileen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). PyTorch: An Imperative Style,

High-Performance Deep Learning Library. 33<sup>rd</sup> Conference on Neural Information Processing Systems, 721(2019), 8026-8037.

- Tsitsiklis, J. N., & Van Roy., B. (1997). An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5), 674-690. https://doi.org/10.1109/9.580874
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. 31st Conference on Neural Information Processing Systems (NIPS 2017), 1-11. https://doi.org/10.48550/arXiv.1706.0376
- Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. [Ph.D. thesis]. King's College.
- Watkins, C. J. C. H, & Dayan, P. (1992). Technical note: Qlearning. *Machine Learning*, 8(1992), 279-292. https://doi.org/10.1023/A:1022676722315

# Modeling Prominence Constraints for German Pronouns as Weighted Retrieval Cues

Umesh Patil (umesh.patil@gmail.com)

IdSL I, Linguistics, University of Cologne 50923 Cologne, Germany

Petra B. Schumacher (petra.schumacher@uni-koeln.de)

IdSL I, Linguistics, University of Cologne 50923 Cologne, Germany

#### Abstract

We propose an ACT-R model of processing German personal and demonstrative pronouns. The model extends existing cuebased retrieval models of sentence processing (Lewis & Vasishth, 2005; Lewis et al. 2006) and pronoun resolution (Parker & Phillips, 2017; Patil & Lago, 2021) by adding prominence constraints as weighted retrieval cues. We model data from an antecedent selection task reported in Schumacher et al. (2016). The experiment varied word orders (canonical vs. non-canonical) and verb types (active accusative vs. dative experiencer) to test the effect of varying referential prominence on antecedent preferences for personal and demonstrative pronouns. The model with weighted prominence cues captures key effects across two word orders and verb types, and demonstrates that the contrastive antecedent preferences of personal and demonstrative pronouns can be captured using weighted retrieval cues reflecting prominence constraints.

**Keywords:** pronoun resolution; prominence; German pronouns; ACT-R; cue-based retrieval

# **Pronoun resolution**

In a sentence such as *Peter wanted to go jogging with Paula, but he had a cold*, the task of finding out what the pronoun *he* refers to involves: (i) using the linguistic knowledge that the referent should prototypically have a masculine gender, (ii) maintaining the memory representation of all the referents encountered so far, i.e. *Peter* and *Paula*, and (iii) carrying out the computation of retrieving the correct antecedent, *Peter*, and identifying it with the personal pronoun *he*.

#### Personal vs. demonstrative pronouns in German

In German, apart from the personal pronouns (PPros, henceforth) *sie/er/es* (she/he/it), there are also demonstrative pronouns (DPros) *die/der/das* (she/he/it) which are used very productively. PPros and DPros differ in their antecedent preferences. In (1) the PPro *er* can refer to both the subject (*the firefighter*) and the object (*the boy*), but has a mild preference towards the subject antecedent. The DPro *der* on the other hand shows a strong preference towards the object antecedent.

(1) [Der Feuerwehrmann]<sub>i</sub> will [den Jungen]<sub>j</sub> retten, aber  $er_{\{i, j\}}/der_{\{2i, j\}}$  ist zu aufgeregt.

[The firefighter]<sub>i</sub> wants [the boy]<sub>j</sub> to-rescue, but  $he_{PPro\{i, j\}} / he_{DPro\{?i, j\}}$  is too nervous

'The firefighter wants to rescue the boy, but he is too nervous.'

In general, it has been claimed that PPros prefer, whereas DPros disprefer the most salient or prominent referent (Bosch, Rozario, & Zhao, 2003). Here, prominence is computed in terms of subjecthood (Bosch, Katz, & Umbach, 2007; Kaiser, 2011), agenthood (Schumacher, Dangl, & Uzun, 2016; Schumacher, Roberts, & Järvikivi, 2017), order of mention (Schumacher et al., 2016; Bader & Portele, 2019), topicality (Bosch & Umbach, 2007; Hinterwimmer, 2015), perspective taking (Hinterwimmer & Bosch, 2018; Hinterwimmer, Brocher, & Patil, 2020), or a combination of more than one of these factors (Schumacher, Backhaus, & Dangl, 2015; Portele & Bader, 2016 among others).

In (1) the factors of subjecthood and agenthood align such that *the firefighter* is the subject and agent of the sentence, whereas *the boy* is the object and patient of the sentence. However, when subjecthood and agenthood don't align, German pronouns show a mixed effect of subjecthood and agenthood (Schumacher et al., 2016; Patterson & Schumacher, 2021).

# Data: Schumacher et al. (2016) Expt. 1

Schumacher et al. (2016) carried out a set of offline studies to tease apart the effect of the factors of subjecthood, agenthood and the order of mention for German PPros and DPros. In Experiment 1, they used experimental items as in (2) where they varied the verb type — active accusative (2a and 2b) vs. dative experiencer (2c and 2d) — and the word order — canonical (2a and 2c) vs. non-canonical (2b and 2d). Each of these four conditions occurred in two variations such that the pronoun was either a PPro or a DPro. This lead to eight conditions in total.

- (2) a. Active accusative verb in canonical word order [AA-CA]
  Der Feuerwehrmann will den Jungen retten, weil das Haus brennt. Aber er/der ist zu aufgeregt. The firefighter wants to rescue the boy, because the house is on fire. But he<sub>PPro</sub>/he<sub>DPro</sub> is too nervous.
  - b. Active accusative verb in non-canonical word order [AA-NC]

Den Jungen will der Feuerwehrmann retten, weil das Haus brennt. Aber **er/der** ist zu aufgeregt.

It is the boy who the firefighter wants to rescue, because the house is on fire. But  $he_{PPro}/he_{DPro}$  is too nervous.

c. Dative experiencer verb in canonical word order [DE-CA]

Dem Zuschauer ist der Terrorist aufgefallen, und zwar nahe der Absperrung. Aber **er/der** will eigentlich nur die Feier sehen.

The spectator has noticed the terrorist, in fact next to the barrier. But  $\mathbf{he}_{PPro}/\mathbf{he}_{DPro}$  actually only wants to watch the ceremony.

d. Dative experiencer verb in non-canonical word order [DE-NC]

Der Terrorist ist dem Zuschauer aufgefallen, und zwar nahe der Absperrung. Aber **er/der** will eigentlich nur die Feier sehen.

It is the terrorist who the spectator noticed, in fact next to the barrier. But  $he_{PPro}/he_{DPro}$  actually only wants to watch the ceremony.

This design made sure that prominence cues are not always aligned for the two referents in the first sentence. In condition (a) the first-mentioned referent (*the firefighter*) has AGENT as the thematic role and SUBJECT as the grammatical role because the verb 'retten' (*to rescue*) is an active accusative verb with a canonical nominative-accusative order. On the other hand, in condition (c) the first-mentioned referent (*the spectator*) has AGENT as the thematic role, but OBJECT as the grammatical role since the verb 'auf(ge)fallen' (*to no-tice*) is a dative experiencer verb with a canonical dative-nominative order. Table 1 lists the thematic and grammatical roles of the two referents across conditions (a-d). Note that the authors followed the proto-role account of Dowty (1991).

Table 1: Thematic and grammatical roles of the two referents across four conditions (see (2) for details of the conditions). Ref. = referent; Th. role = thematic role; Gr. role = grammatical role; AGT = agent; PAT = patient; SUB = subject; OBJ = object.

Condition	Referent	Th. role	Gr. role
	Ref. 1	AGT	SUB
a. AA-CA	Ref. 2	PAT	OBJ
h AA NC	Ref. 1	PAT	OBJ
D. AA-NC	Ref. 2	AGT	SUB
a DE CA	Ref. 1	AGT	OBJ
C. DE-CA	Ref. 2	PAT	SUB
A DE NC	Ref. 1	PAT	SUB
u. DE-NC	Ref. 2	AGT	OBJ

In the experiment, participants saw sentences as in (2) and performed a two-alternative forced choice task where they indicated which of the two referents in the previous sentence they preferred as the antecedent of the pronoun. Antecedent preferences across eight conditions in terms of mean percentages of choosing the first referent listed in Table 3 in the column 'Data'. The percentages for selecting the second referent are complementary percentages since it was a two-alternative forced choice task.

In sum, three important results emerged: [Effect-1] for active accusative verbs, where subjecthood and agenthood align, PPros preferred the referent that was subject and agent (Ref. 1 in AA-CA and Ref. 2 in AA-NC), whereas DPros preferred the referent that was object and patient (Ref. 2 in AA-CA and Ref. 1 in AA-NC), [Effect-2] for dative experiencer verbs, the preferences were less straightforward such that in the canonical word order, the PPros preferred the first-mentioned referent (Ref. 1) which was object and agent, whereas DPros preferred the last-mentioned referent (Ref. 2) which was subject and patient; however, [Effect-3] in the non-canonical condition (for dative experiencer verbs), there was no preference for the first- or last-mentioned referent for either of the pronouns.

In terms of probability of choosing one referent over the other (Ref. 1 vs. Ref. 2), the effects could be listed as: **Effect-1**:

 $\begin{array}{l} P(\text{Ref. 1}|\text{AA-CA, PPro}) > P(\text{Ref. 2}|\text{AA-CA, PPro}) \\ P(\text{Ref. 1}|\text{AA-NC, PPro}) < P(\text{Ref. 2}|\text{AA-NC, PPro}) \\ P(\text{Ref. 1}|\text{AA-CA, DPro}) < P(\text{Ref. 2}|\text{AA-CA, DPro}) \\ P(\text{Ref. 1}|\text{AA-NC, DPro}) > P(\text{Ref. 2}|\text{AA-NC, DPro}) \\ P(\text{Ref. 1}|\text{DE-CA, PPro}) > P(\text{Ref. 2}|\text{DE-CA, PPro}) \\ P(\text{Ref. 1}|\text{DE-CA, DPro}) < P(\text{Ref. 2}|\text{DE-CA, DPro}) \\ P(\text{Ref. 1}|\text{DE-CA, DPro}) < P(\text{Ref. 2}|\text{DE-CA, DPro}) \\ \hline \\ \textbf{Effect-3:} \\ P(\text{Ref. 1}|\text{DE-NC, PPro}) \approx P(\text{Ref. 2}|\text{DE-NC, PPro}) \\ P(\text{Ref. 1}|\text{DE-NC, DPro}) \approx P(\text{Ref. 2}|\text{DE-NC, DPro}) \\ \hline \end{array}$ 

Schumacher et al. (2016) interpreted these results as providing evidence for the interaction of multiple prominence factors, and thematic role being ranked higher than other constraints for the interpretation of PPros and DPros.

# Antecedent preference as cue-based retrieval

The cue-based retrieval theory (CBR, henceforth) proposed in Lewis and Vasishth (2005) and Lewis, Vasishth, and Van Dyke (2006) has been successfully applied to model the memory retrieval processes involved in forming dependencies between two linguistic units such as nounverb agreements (Wagers, Lau, & Phillips, 2009) and pronoun-antecedent dependencies (Dillon, Mishler, Sloggett, & Phillips, 2013; Parker & Phillips, 2017; Patil, Vasishth, & Lewis, 2016; Patil & Lago, 2021). The CBR theory, which is implemented in the general cognitive architecture ACT-R (Anderson, Byrne, Douglass, Lebiere, & Qin, 2004), describes sentence processing as a series of activation-based skilled memory retrievals. Lexical knowledge and current partial representation of the input (the parse) is maintained in declarative memory, and psycholinguistic processes are represented in procedural memory. Incremental sentence processing occurs through selection of procedural memory rules (parsing procedures) that retrieve declarative memory representations and operate on them to update the sentence representation.

Here our goal is to use existing CBR models of pronoun resolution and test if they can be extended in a meaningful way to model the differences in terms of prominence constraints for pronouns in German. For doing so Expt. 1 from Schumacher et al. (2016) provides a suitable data set because it shows variations in antecedent preferences based on varying prominence features of the antecedents. Moreover, the data exemplifies the contrastive nature of the constraints for the two types of pronouns — PPros vs. DPros — used in the experiment (see Section (1) for details of the data).

# Model of Schumacher et al. (2016) Expt. 1

For modeling data from Schumacher et al. (2016), we carried out the following steps. First we implemented a baseline model, similar to the earlier CBR models of pronounantecedent dependency, which included a subset of the phi features as retrieval cues at the pronoun to retrieve the antecedent (cf. Table 2). Then we extended the model with prominence constraints, and finally with weighted prominence constraints. To avoid overfitting the model, we restricted to first implementing a model for the data from PPros. In general, PPros and DPros show opposite constraints for antecedents — PPros prefer a prominent referent and DPros disprefer a prominent referent. Hence, once a model for PPros is determined, the same model with contrasting retrieval cues should be able to capture the data for DPros. Such a complementarity seems to be warranted on the basis of experimental research on German (but see form-specific approaches to reference resolution that have been proposed for other languages, Kaiser & Trueswell, 2008).

A list of retrieval cues and their corresponding values used at the pronoun for all the models reported here is given in Table 2. All models assume that the referent retrieved by the retrieval process at the pronoun is the selected antecedent for the pronoun. Model predictions were generated by running 10000 simulations for each model which gave rise to 10000 possible choices between Ref. 1 and Ref. 2 for each condition. If a referent was retrieved significantly more often than the other (i.e. above chance) across all simulations, we assumed that the model selected that referent in that condition, and if there was no statistically reliable preference, we assumed that the model did not show preference for either of the referents. The statistical significance was tested using logistic regression (Dyke & Patterson, 1952). In the rest of the text, if a model is reported to prefer a referent then it means that the referent was selected significantly more often than the other referent. All ACT-R parameters had the same values as used in Lewis and Vasishth (2005) except for cue-weighting in Model 3.

Table 2: List of retrieval cues and their values across all models. The only difference in Model 2 and 3a was in terms of weighting — the cue *thematic role* was weighted to be 1.5 times higher than all other cues. Mod. = model; Cat. = (phrasal) category; Th. = thematic role; Gr. = grammatical role; Ord. = order of mention; DP = determiner phrase; M. = masculine; Sg. = singular; AGT = agent; PAT = patient; SUB = subject; OBJ = object.

Mod.			Retri	eval cue	;	
	Cat.	Gn.	No.	Th.	Gr.	Ord.
1	DP	M.	Sg.	-	-	-
2	DP	М.	Sg.	AGT	SUB	first
3a	DP	M.	Sg.	AGT	SUB	first
3b	DP	М.	Sg.	PAT	OBJ	last

#### Model evaluation

The evaluation of models was carried out qualitatively. For each condition we compared the referent selected by the model with the referent selected by the participants in the Schumacher et al. (2016). We considered that an effect is captured by a model if the model selected the same referent as in the data, disregarding the precise value of the proportions for the referent. That means if the data showed, for example, 62% preference for Ref. 1 and a model predicted the preference for Ref. 1 to be 55% and it was statistically significant, we considered that the model captured this effect. The rationale behind modeling only the categorical preference instead of the probability distribution of preferences was to avoid overfitting the parameters based on a single data set.

# Model 1: Baseline model

The baseline model assumed that the antecedent for the PPros is retrieved using the cues 'gender' (= masculine), 'number' (= singular) and 'category' (=DP, a determiner phrase). We considered this to be the baseline model because the specification of retrieval cues was the same as the earlier CBR models of antecedent retrieval (e.g. Patil & Lago, 2021) and it did not have any extension to consider the manipulation of prominence factors in the design of the Schumacher et al. (2016) experiment. The predictions of the model, in terms of the antecedent preferences, are shown in Fig. 1 and in Table 3 in the column for Model 1. The model showed an unanimous preference for the second referent and the preference was equal across four conditions. Although the retrieval cues were consistent with the features of both the referents, the model preferred the second referent more often because, being mentioned more recently, its activation was higher than the first mentioned referent. This effect was driven by ACT-R's cognitive principle of activation decay applied to language processing (Lewis & Vasishth, 2005). The model only partially captured Effect-1 for active accusative verbs in non-canonical word order, but did not capture any other effect for PPros (see Section (1) for the list of effects).



Figure 1: Data and model predictions for the antecedent selection task for PPros in Schumacher et al. (2016) Experiment 1. Bars represent the percentage of selecting a referent.

Table 3: Data and model predictions for the antecedent selection task in Schumacher et al. (2016) Experiment 1. Each cell represents the percentage of selecting the first referent (Ref. 1) in that condition. The percentages for selecting the second referent (Ref. 2) are complementary percentages since it is a two-alternative forced choice task. The first four rows are for PPros and the last four are for DPros.

				Model	
	Condition	Data	1	2	3a/3b
	a. AA-CA	62%	41%	82%	86%
Q	b. AA-NC	43%	42%	26%	24%
ЪР	c. DE-CA	59%	42%	55%	63%
	d. DE-NC	47%	40%	56%	48%
	a. AA-CA	23%	-	-	8%
ro	b. AA-NC	67%	-	-	60%
DP	c. DE-CA	35%	-	-	24%
	d. DE-NC	52%	-	-	34%

#### Model 2: Model with prominence constraints

We extended the baseline model by adding retrieval cues that reflected factors influencing prominence of the two referents. Effectively, we added cues for thematic roles, grammatical roles and order of mention (see Table 2). The predictions of the model, in terms of the antecedent preferences, are shown



Figure 2: Data and model predictions for the antecedent selection task for DPros in Schumacher et al. (2016) Experiment 1. Bars represent the percentage of selecting a referent.

in Fig. 1 and in Table 3 in the column for Model 2. The model captured *Effect-1* and *Effect-2* for the PPro: for active accusative verbs the PPro prefers the referent that was subject and agent (independent from canonicity) and for the canonical condition in the dative experiencer verbs the PPro prefers the first-mentioned referent that was object and agent. However, the model didn't capture *Effect-3*: for the non-canonical condition in the dative experiencer verbs it predicted a preference for the first-mentioned referent that was subject and patient whereas in the data there was no clear preference for either of the two referents. This model was clearly an improvement over the baseline model since it captured data better.

# Model 3a: Model with weighted prominence constraints

Schumacher et al. (2016) proposed that although multiple prominence-lending factors contribute to the reference resolution process, thematic role (e.g. agenthood) is a higher ranked factor among them. They suggested that the higher ranking of agenthood could be because of the general cognitive traits associated with (proto)agents because "Agents are a class of objects possessing sets of causal properties that distinguish them from other physical objects" (Leslie, 1995). The stronger effect of agenthood is evident in the data for condition (c) (AA-CA) as well — the PPro preferred the referent with agent and object roles over the referent with patient and subject roles (see Table 3).

In ACT-R all retrieval cues have the same weight, but in psycholinguistics it has been proposed that certain retrieval cues could be weighted higher than others (see for example: Parker, Shvartsman, & Van Dyke, 2017; Vasishth, Nicenboim, Engelmann, & Burchert, 2019; Patil & Lago, 2021). To incorporate the stronger effect of agenthood, we added cue weighting and weight the thematic role cue higher than the other cues and tested if that improved the model performance. We modified the default *strengths of association* equation in ACT-R from Equation 1 to Equation 2 and added an extra parameter for each retrieval cue (see Anderson et al., 2004 or Lewis & Vasishth, 2005 for details about the *strengths of association* equation and its influence on the retrieval process). In Equation 2, the *weight* term represents the weight of cue *j* during the retrieval of element *i*. This modification could also be seen as modifying the value of the ACT-R parameter *maximum associative strength* for a specific retrieval cue.

$$S_{ij} = S - ln(fan_j) \tag{1}$$

$$S_{ji} = weight_{i}S - ln(fan_{i})$$
<sup>(2)</sup>

In the modified model we weighted the retrieval cue of thematic role 1.5 times higher than other cues used to retrieve the antecedent. All other cues had a default weight of 1. Note that for cues with the default weight values, Equation 2 reduces to Equation 1, and hence the *strengths of association*,  $S_{ij}$ , is the same as it would be in default ACT-R; however, when the weight value is different than 1, the value for *strengths of association* reflects the weighted importance of that particular cue. The predictions of the new model are shown in Fig. 1 and in Table 3 in the column for Model 3a. The modified model, just like Model 2, captured *Effect-1* and *Effect-2*, but it also captured *Effect-3*: the model predicted no clear preference for either of the referents for the non-canonical condition in the dative experiencer verbs.

#### Model 3b: Model for DPros

In contrast to PPros, which prefer prominent antecedents, DPros are claimed to disprefer prominent antecedents. Because of this contrastive preference between the two pronouns, we predicted that the model for PPros to work for DPros with changes only in the values of the retrieval cues for prominence factors, and should not require any other changes. The corresponding modified values of the retrieval cues for DPros are listed in Table 2 in the row for Model 3b. The predictions of the model for DPros are shown in Fig. 2 and in Table 3 in the column for Model 3b. The model captured Effect-1 and Effect-2, however, it didn't capture Effect-3: for the non-canonical condition in dative experiencer verbs the model predicted a preference for the last-mentioned referent, whereas the data didn't show any clear preference. This may indicate that DPros and PPros do not entirely show complementary interpretation preferences and are subject to formspecific weightings. This should be addressed in future research.

#### General discussion and conclusions

The results from the modeling experiments showed that a modified cue-based retrieval model can capture important patterns in the data for German personal and demonstrative pronouns. We started with a baseline model, in the CBR framework, for data for PPros from Experiment 1 in Schumacher et al. (2016). Since the model did not capture

crucial patterns in the data that emerged due to the variations in the prominence of the referents, namely, the word order variation (canonical vs. non-canonical) and the verb type variation (active accusative vs. dative experiencer), we extended the model to include retrieval cues reflecting prominence constraints. The model that included prominence constraints performed better than the baseline model. A further improvement of the model was observed when we weighted the retrieval cues to assign a higher weight to the cue specifying the thematic role of the antecedent. Since the model for PPros with weighted retrieval cues captured all the crucial patterns in the data, we modified this model to reflect the contrast in prominence constraints between PPros and DPros, and tested its predictions for DPros. The model for DPros indeed captured two out of three crucial effects observed in the data. Extending the model to capture all three effects for DPros will certainly be the next crucial step.

Our final model had two main limitations: (1) it only captured the categorical preferences between two antecedents but not the probability distributions of preferences across the antecedents, and (2) it could not capture one of the three effects for DPros. Indeed, extending the model to overcome these limitations will be the next important step; however, such an extension should be based on results pooled from multiple experiments.

In sum, the model reported here: (1) captures crucial patterns in the data from an antecedent selection task with German personal and demonstrative pronouns, (2) shows that prominence constraints on pronouns can be translated to weighted retrieval cues in the cue-based retrieval framework, and (3) shows that the contrastive antecedent preferences of personal and demonstrative pronouns can be captured to a certain extent with contrastive retrieval cues. We consider the model as an important step towards modeling the processing of pronouns as a cue-based retrieval process.

### Acknowledgements

The research for this project has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 281511265 – SFB 1252 "Prominence in Language" in the project C07 "Forward and backward functions of discourse anaphora" at the University of Cologne, Department of German Language und Literature I, Linguistics. We furthermore thank the Regional Computing Center of the University of Cologne (RRZK) for providing computing time on the DFG-funded (Funding number: INST 216/512/1FUGG) High Performance Computing (HPC) system CHEOPS as well as support.

# References

- Anderson, J. R., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111(4), 1036–1050.
- Bader, M., & Portele, Y. (2019). The interpretation of german personal pronouns and d-pronouns. *Zeitschrift f
  ür Sprachwissenschaft*, 38(2), 155–190.

- Bosch, P., Katz, G., & Umbach, C. (2007). The non-subject bias of german demonstrative pronouns. In M. Schwarz-Friesel, M. Consten, & M. Knees (Eds.), Anaphors in text: Cognitive, formal and applied approaches to anaphoric reference (pp. 145–164). Amsterdam: John Benjamin.
- Bosch, P., Rozario, T., & Zhao, Y. (2003). Demonstrative pronouns and personal pronouns: German der vs er. In *Proceedings of the 2003 EACL workshop on the computational treatment of anaphora*.
- Bosch, P., & Umbach, C. (2007). Reference determination for demonstrative pronouns. In D. Bittner & N. Gagarina (Eds.), *Proceedings of the conference on intersentential pronominal reference in child and adult language* (pp. 39–51). ZAS Papers in Linguistics.
- Dillon, B., Mishler, A., Sloggett, S., & Phillips, C. (2013). Contrasting intrusion profiles for agreement and anaphora: Experimental and modeling evidence. *Journal of Memory and Language*, 69(2), 85-103.
- Dowty, D. (1991). Thematic proto-roles and argument selection. *Language*, 67(3), 547–619.
- Dyke, G. V., & Patterson, H. D. (1952). Analysis of factorial arrangements when the data are proportions. *Biometrics*,  $\delta(1)$ , 1–12.
- Hinterwimmer, S. (2015). A unified account of the properties of demonstrative pronouns in german. In P. Grosz, P. Patel-Grosz, & I. Yanovich (Eds.), *Workshop on pronouns at the 40th conference of the north eastern linguistic society, nels 40* (pp. 61–107). GLSA Publications.
- Hinterwimmer, S., & Bosch, P. (2018). Demonstrative pronouns and propositional attitudes. In P. Patel-Grosz,
  P. G. Grosz, & S. Zobel (Eds.), *Pronouns in embedded contexts at the syntax-semantics interface* (p. 105-144). Springer (Studies in Linguistics and Philosophy).
- Hinterwimmer, S., Brocher, A., & Patil, U. (2020). Demonstrative Pronouns as Anti-Logophoric Pronouns: An Experimental Investigation. *Dialogue & Discourse*, 11(2), 110-127.
- Kaiser, E. (2011). On the relation between coherence relations and anaphoric demonstratives in german. In I. Reich, E. Horch, & D. Pauly (Eds.), *Proceedings of sinn und bedeutung 15* (pp. 337–351). Saarbrücken: Saarland University Press.
- Kaiser, E., & Trueswell, J. C. (2008). Interpreting pronouns and demonstratives in finnish: Evidence for a form-specific approach to reference resolution. *Language and Cognitive Processes*, 23(5), 709-748.
- Leslie, A. M. (1995). A theory of agency. In (p. 121-149). New York, NY, US: Clarendon Press/Oxford University Press.
- Lewis, R. L., & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29, 1-45.
- Lewis, R. L., Vasishth, S., & Van Dyke, J. A. (2006). Computational principles of working memory in sentence comprehension. *Trends in Cognitive Sciences*, 10(10), 447–454.

- Parker, D., & Phillips, C. (2017). Reflexive attraction in comprehension is selective. *Journal of Memory and Language*, 94, 272-290.
- Parker, D., Shvartsman, M., & Van Dyke, J. A. (2017). The cue-based retrieval theory of sentence comprehension: New findings and new challenges..
- Patil, U., & Lago, S. (2021). Prediction advantage as retrieval interference: an ACT-R model of processing possessive pronouns. In T. C. Stewart (Ed.), *Proceedings of the 19th international conference on cognitive modeling* (p. 213-219). University Park, PA: Applied Cognitive Science Lab, Penn State.
- Patil, U., Vasishth, S., & Lewis, R. L. (2016). Retrieval interference in syntactic processing: The case of reflexive binding in english. *Frontiers in Psychology*, 7, 329.
- Patterson, C., & Schumacher, P. B. (2021). Interpretation preferences in contexts with three antecedents: examining the role of prominence in german pronouns. *Applied Psycholinguistics*, *42*(6), 1427?1461.
- Portele, Y., & Bader, M. (2016). Accessibility and referential choice: Personal pronouns and d-pronouns in written german. *Discours*, 18, 9188.
- Schumacher, P. B., Backhaus, J., & Dangl, M. (2015). Backward- and forward-looking potential of anaphors. *Frontiers in Psychology*, *6*, 1746.
- Schumacher, P. B., Dangl, M., & Uzun, E. (2016). Thematic role as prominence cue during pronoun resolution in german. In A. Holler & K. Suckow (Eds.), *Empirical perspectives on anaphora resolution* (pp. 121–147). Berlin/New York: De Gruyter.
- Schumacher, P. B., Roberts, L., & Järvikivi, J. (2017). Agentivity drives real-time pronoun resolution: Evidence from german er and der. *Lingua*, 185, 25–41.
- Vasishth, S., Nicenboim, B., Engelmann, F., & Burchert, F. (2019). Computational models of retrieval processes in sentence processing. *Trends in Cognitive Sciences*, 23(11), 968-982.
- Wagers, M. W., Lau, E. F., & Phillips, C. (2009). Agreement attraction in comprehension: Representations and processes. *Journal of Memory and Language*, 61(2), 206– 237.

# Predicting Learning in a Troubleshooting Task using a Cognitive Architecture-Based Task Analysis

# Frank E. Ritter (frank.ritter@psu.edu), Deja Workman, and Shan Wang

College of IST, Penn State University Park, PA 16802

#### Abstract

We present a new way to do task analysis that includes learning. This approach starts with a hierarchical task analysis of a troubleshooting strategy and applies a power law of learning to modify the time, mimicking the ACT-R learning equations. We apply this approach to finding faults in the Ben Franklin Radar (BFR) system, a 35-component system, designed to study troubleshooting and learning. In this task, faults are introduced into the BFR, and the participants are responsible for finding and fixing these automatic faults. Previous models in Soar took up to 6-9 months of graduate student to create. This model can be created more quickly and provides a model between GOMS and a full cognitive architecture-based model. The predictions will be compared to the aggregate and individuals' data (N=111) and lessons will be reported.

Keywords: ACT-R, learning, task analysis, troubleshooting

Acknowledgements: This work was supported by ONR, N00014-18-C-7015 and N00014-15-1-2275. Charles River Analytics implemented the MENDS simulation, provide it under license, and provided comments and support on the process.

# Introduction

Can we examine a particularly complex task and predict how long each trial will take while the task is learned without creating a full information processing cognitive model? In this paper we create predictions of a complex task using GOMS and the learning equations, thus extending the GOMS approach and providing a way to make approximate predictions of learning a task.

To illustrate this approach, we make detailed predictions about trouble shooting a complex task where the fault in the circuits may vary in difficulty. We use GOMS and learning equations.

The rest of the paper describes the task and a model used to create a series of predictions of doing the task and learning. We then present the study used to gather human performance data. We then describe the comparison we will make with the model's predictions to the human data in aggregate form and individually, and already can draw some insights.

# Task

We needed a complex task to study. The Ben Franklin Radar (BFR), shown in Figures 1 and 2, is a deliberately 5x larger system than the Klingon Laser Bank task that has been previously used to study problem solving, learning, and transfer (Bibby & Payne, 1996; Friedrich & Ritter, 2020; Ritter & Bibby, 2008). The Klingon Laser Bank task with 7 components initially takes about 30 s and with 20 trials takes about 7 s.

The MENDS simulator was created for the BFR, shown in Figure 1. The schematic and interface can be taught to participants in a 32-page online tutor created in D2P (Ritter et al., 2013). It takes about 30 min. to learn declarative information about it and the task (Ritter, Tehranchi, Brener, & Wang, 2019). The schematic shows five subsystems. The subsystems vary in their complexity and connectivity within them and across subsystems. The blue lines in Figure 1 are power connections; the red lines are information; the purple lines are both. The schematic also identifies certain components that have their status displayed on the front panel of the BFR.

Our task was created to support learning troubleshooting within the confines of a study, and to be more complex than the Klingon Laser Bank task, but not so complex that it would take more than an hour to learn. This system can be and has been realized in several ways with different complexity. The task that we will focus on is to find a single broken component. Single broken faults create a unique light configuration and are always solvable.

The task requires declarative knowledge about the schematic and interface. It also requires some recognition memory and perhaps recall of the components. The task also supports creating procedural knowledge from the declarative knowledge by doing the task.



Figure 1. Schematic of the Ben-Franklin Radar simulation.



Figure 2. A snapshot of the MENDS simulation. In this picture, the fault is in the Processor subsystem.

# A Simple Task Model of Learning and Fidelity

To troubleshoot the task (simplest method) the user clicks for the next problem, examines the lights, clicks on a tray, and examines its contents. They must then choose the broken component by clicking on it and clicking done.

We can start with some insights that are already apparent. The BFR task is designed to be more complex than the Klingon Laser Bank task. Thus, there are more predictions to generate (7 vs. 35). Ritter and Bibby (2008) found that one strategy matched the majority of participants. Seven you can do by hand, 35 requires more infrastructure. Friedrich and Ritter (2020) found with more relaxed instructions there were more strategies. With this complexity may come even further strategies to solve this task. Thus, this model will initially present just one of these strategies.

The model represents the structure of the BRF as a series of connections in a matrix. To create the structure of the model, a binary schematic was created in Excel to represent the component dependencies found in the BFR. Python converted this matrix into a data frame storing components' input and outputs—that is, the list of other components that a component itself expects to receive power from or send power to. The data structure also dynamically stores the component status and light, which respectively represent whether the component is functioning properly and whether it is receiving power, they are both binary variables.

In some sense, we thus create an ad hoc domain specific language (DSL) cognitive modeling language (Kaulakis, 2020) using Excel and Python for reading in circuit matrices. This approach would support creating models of similar structures and is at this point be fairly direct and quick to use. The model uses these structures to generate times to find a fault. Currently, the model can set up and run the task, and solve for a fault.

*MakeFault* is responsible for the creation of the fault, it begins by calling the *clearFault* function and then it generates a random number that is within the bound of the size of the data frame, which corresponds to one of the 36 components of the system—it cannot choose the power supply or any of the switches as a fault because these are not tasks we present participants within the actual study. *Propagate* is responsible for computing the effect of the fault. This function identifies all the outputs of the piece, and subsequently

turns their light off as they are no longer receiving all of their necessary inputs due to the fault.

The holistic responsibility of this operation is not only to recreate the way in which the system breaks itself, more importantly is the program's ability to locate and fix its own faults. This logic is stored predominantly in the *FindFault* function, though it calls upon external elements as well as the mental operator function. *FindFault* is meant to mimic the way in which we believe participants solve the problem using a simple strategy that they are presented with.

# Human Participant Data that We Have So Far

We have two sets of data. In the MENDS task, Ritter et al. (2019) saw a subject with 10 minutes of practice that went from 60 s to 22 s. And, we have finished a study (N=110) that has more data. We are analyzing it now and have learning curves for all participants that we will compare to this model. The BFR task does take about 5 times longer than the original task both initially and at 20 trials.

# **Discussion and Conclusion**

The model is still being developed, but offers a way to predict learning for static approaches. This model already predicts that later faults take longer, that learning is inevitable, that many faults take different times, and surprisingly, that many faults would take the same time but use different subtasks to get there. For example, fixing a component late in an early tray may take as long as a component in a later tray but displayed earlier within that tray.

# References

- Bibby, P. A., & Payne, S. J. (1996). Instruction and practice in learning to use a device. *Cognitive Science*, 20(4), 539-578.
- Friedrich, M. B., & Ritter, F. E. (2020). Understanding strategy differences in a diagrammatic reasoning task. *Cognitive Systems Research*, *59*, 133-150.
- Kaulakis, R. (2020). Domain-specific languages as a method for representation and evolutionary search among global solution space of cognitively plausible algorithmic behaviors. Unpublished PhD thesis, College of IST, Penn State, University Park, PA.
- Ritter, F. E., & Bibby, P. A. (2008). Modeling how, when, and what is learned in a simple fault-finding task. *Cognitive Science*, *32*, 862-892.
- Ritter, F. E., Tehranchi, F., Brener, M., & Wang, S. (2019). Testing a complex training task. In *Proceedings of the* 17th International Conference on Cognitive Modeling (ICCM 2019), 184-185.
- Ritter, F. E., Yeh, K.-C., Cohen, M. A., Weyhrauch, P., Kim, J. W., & Hobbs, J. N. (2013). Declarative to procedural tutors: A family of cognitive architecturebased tutors. In *Proceedings of the 22nd Conference on Behavior Representation in Modeling and Simulation*, 13-BRIMS-127. 108-113. Centerville, OH: BRIMS Society.

# **Discontinuities in Function Learning**

Roderick Yang Terng Seow (yseow@andrew.cmu.edu) John R. Anderson (ja0s@andrew.cmu.edu) Department of Psychology, Carnegie Mellon University, 5000 Forbes Ave Pittsburgh, PA 15213 USA

**Keywords:** function learning, exemplar, rule, abrupt, discontinuity

# Introduction

Function learning is the process by which humans acquire knowledge of functional relationships between continuous variables. For example, a frequent beachgoer might visit the beach on different nights and come to associate specific tide heights with specific moon phases. With experience, the beachgoer might then abstract an underlying functional relationship: the tide rises approaching the full moon, and lowers approaching the new moon.

Most theories of function learning largely focus on two types of models: exemplar-based and rule-based models. Exemplar-based models posit that humans learn to associate exemplar cues with their respective targets via error-driven updates of associative weights (Busemeyer et al., 1997). Rule-based models posit that humans instead begin with some parametric function and learn its coefficients through an error-driven update mechanism (e.g. polynomial rule model: Koh and Meyer (1991)). More recent studies have proposed hybrid models that combine associative learning with rules, and these models have been shown to better account for a wide range of function learning phenomena than their predecessors (e.g. EXtrapolation Association Model (EXAM): DeLosh et al. (1997); Population Of Linear Experts (POLE): Kalish et al. (2004)).

Despite their differences, existing process models mostly assume that function learning is a gradual and continuous process. In contrast, Brehmer (1974) proposed a two-staged hypothesis testing theory of function learning. The first stage involves discovering a suitable rule, and the second stage is concerned with learning the parameters of the rule. Although this theory has not been quantitatively formalized, it differs from the other theories by positing a discontinuity when the learner transitions from discovering a rule to applying a rule. In support of the role of rule discovery in human function learning, we present preliminary evidence of such discontinuities and demonstrate that existing process models do not adequately account for these observations.

### Experiment

The experiment was a replication of study 1a of McDaniel et al. (2014), but with Amazon Mechanical Turkers instead of undergraduate students. 59 participants, 21 females, ages ranging from 20 to 53 years old (mean = 32.3), completed the experiment. Participants were paid \$4.50 for completion and an accuracy bonus up to \$0.02 on every training trial.

Participants completed 10 training blocks followed by 1 transfer block. Each training block consisted of the same 20 trials presented in random orders. For each training trial, the cue value was represented using the height of a colored bar, and participants made their predictions using arrow keys to adjust the height of a separate response bar. Feedback was presented in three forms: the response bar at the target height, an error score consisting of the numerical difference between the response and the target values, and an accuracy score computed as  $100 - error^2$ . Transfer trials consisted of novel cue values, both within (interpolation) and beyond (extrapolation) the range of training cue values. No feedback was provided during the transfer block.

Cues and targets were related through a V-shaped function. For cue < 100,  $target = round(229.2 - 2.197 \cdot cue)$ . For  $cue \ge 100$ ,  $target = round(2.197 \cdot cue - 210)$ .

#### **Detecting discontinuities**

One potential behavioral correlate of rule discovery is an abrupt decrease in an individual's error rates as they proceed through the training phase. To detect such discontinuities if and when they occur, we fitted single- and double-function error curves for each participant. Error noise was assumed to be Poisson distributed and these functions specified how the error mean ( $\lambda$ ) changed with trial number (t).

The set of single-function curves comprised a constant mean  $(\lambda = c)$ , an exponentially decreasing mean  $(\lambda = a \cdot e^{-b \cdot (t-1)} + c)$ , and a mean that decreased according to a power function with increasing number of trials  $(\lambda = a \cdot t^{-b} + c)$ . The single-function curves were composed to create the set of double-function curves, with the restriction that the second function was a constant. All double-function curves required an additional change point parameter.

To determine if an individual demonstrated an abrupt decrease in error, we computed two measures. The first measure ( $\Delta BIC$ ) was the difference between the Bayesian Information Criterion of the best fitting single- and double-functions. A large and positive  $\Delta BIC$  indicated that the error curve was much better fit by a double- than a single-function. To quantify abruptness if a transition exists, the second measure ( $\Delta mean$ ) was the difference between the pre- and post-transition fitted means for the best fitting



Figure 1: Error curves for two participants. Colored lines show the best fitting single- and double-function curves. The top panel displays an abrupt decrease in error, whereas the bottom panel displays a gradual decrease in error over time.

double-function. A large and positive  $\Delta mean$  indicated an abrupt decrease in error around the estimated change point.

Using these two measures, we classified participants into those who did and did not show abrupt learning. The 59 participants were first separated into 28 learners and 31 non-learners according to the criterion in McDaniel et al. (2014): learners were those who attained an average absolute error of less than 10 on the last training block. We then determined a combined threshold on our two measures. The threshold was chosen to be as inclusive as possible with the constraint that only learners could be classified as abrupt learners. This yielded a threshold of  $\Delta BIC > 45$  and  $\Delta mean >$ 5. Based on the threshold, 7 out of 59 participants were classified as abrupt learners (Fig. 2).

# **Model comparisons**

To generate individual model simulations, we found the best fitting set of parameters per participant for three process models (polynomial rule, EXAM, and POLE) by maximizing the log-likelihood with respect to the participant's responses. Applying the same classification procedure as above, none of the model simulations were classified as abrupt learners.

### Conclusion

In this study, we identified a subset of participants that demonstrated abrupt decreases in error over the course of a function learning task. Our simulations of the existing process models confirmed that gradual update mechanisms cannot reproduce the observed discontinuities, which is consistent with the hypothesis that these discontinuities correspond to moments of rule discovery. To test this hypothesis, we are currently investigating the nature of rules and the role of rule discovery in human function learning.



Figure 2: Individual participants and model simulations on abruptness measures. The two participants from Fig. 1 are labeled. Thresholds for the two measures are represented by the dashed lines. Individuals in the upper right quadrants of each panel are classified as abrupt learners.

#### Acknowledgments

This research was supported by ONR Grant N00014-15-1-2151 and AFOSR/AFRL award FA9550-18-1-0251.

# References

- Brehmer, B. (1974). Hypotheses about relations between scaled variables in the learning of probabilistic inference tasks. *Organizational Behavior and Human Performance*, *11*(1), 1–27.
- Busemeyer, J. R., Byun, E., Delosh, E. L., & McDaniel, M. A. (1997). Learning functional relations based on experience with input-output pairs by humans and artificial neural networks. In K. Lamberts & D. R. Shanks (Eds.), *Knowledge, concepts and categories* (p. 408–437). The MIT Press.
- DeLosh, E. L., Busemeyer, J. R., & McDaniel, M. A. (1997). Extrapolation: the sine qua non for abstraction in function learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23(4), 968–986.
- Kalish, M. L., Lewandowsky, S., & Kruschke, J. K. (2004). Population of linear experts: knowledge partitioning and function learning. *Psychological review*, 111(4), 1072–1099.
- Koh, K., & Meyer, D. E. (1991). Function learning: induction of continuous stimulus-response relations. *Journal* of Experimental Psychology: Learning, Memory, and Cognition, 17(5), 811–836.
- McDaniel, M. A., Cahill, M. J., Robbins, M., & Wiener, C. (2014). Individual differences in learning and transfer: stable tendencies for learning exemplars versus abstracting rules. *Journal of Experimental Psychology: General*, 143(2), 668–693.

# **Exploring Multitasking Strategies in an ACT-R model of a Complex Piloting Task**

# Garrett Swan (garrett.swan@cubic.com)

Cubic Defense Applications, Inc San Diego, CA 92123 USA

# Christopher R. Fisher (christopher.fisher.27.ctr@us.af.mil)

Cubic Defense Applications, Inc Beavercreek, OH 45324 USA

#### Abstract

Multitasking is a challenging cognitive task, and there are many factors driving which strategy participants use to complete tasks concurrently. We utilized a model comparison approach to evaluate how participants decide which task to switch to next using the Air Force Multiple Attribute Battery (AF-MATB). We used the cognitive architecture, Adaptive Control of Thought - Rational (ACT-R), to simulate multitasking in the AF-MATB. We varied how the model decided which task to attend to next by comparing a purely top-down strategy, a purely reactive, bottom-up selection strategy, and mixtures of the two. We compared simulations of the model to data from Bowers, Christensen, and Eggemeier (2014). The best combination involved a mixture of top-down and bottom-up selection. Neither the purely top-down nor bottom-up selection models performed well. These results suggest that participants use a complex mixture of strategies to multitasking. The use of a top-down strategy suggests participants could develop efficient strategies to multitask successfully, and that participants may be using a more effortful serial search in the AF-MATB, as indicated by the model's serial processing implementation.

Keywords: ACT-R; AF-MATB; multitasking; cognitive architecture

#### Introduction

In our daily and professional lives, we often perform multiple concurrent tasks, such as eating while driving, listening to a coworker while reading an email, or piloting aircraft while monitoring numerous instruments. How individuals are able to multitask is an old and ongoing question in research because there is a vast space of human and environmental factors that impact one's ability to multitask (Meyer & Kieras, 1997; Koch, Poljac, Müller, & Kiesel, 2018; Fischer & Plessow, 2015). Multitasking is interesting from a theoretical perspective because it requires multiple cognitive systems to work together in service of a common goal and to adapt to changing circumstances. Moreover, the space of strategies one could use to accomplish multitasking can be quite large (Salvucci, Taatgen, & Borst, 2009; Smith et al., 2008).

There are different aspects of multitasking where strategies may manifest. For example, many studies have examined individuals' decision strategies to stop one task and switch to another, which could be serial without interruptions, when a sufficient amount of time has passed (Kushleyeva, Salvucci, & Lee, 2005), or when there are diminishing benefits of the currently attended task (Payne, Duggan, & Neth, 2007). Here, we are interested in the strategy that determines which task to switch to next. Individuals may search by top-down factors, such as serially moving attention from task to task, 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

"urgency" (Salvucci, Kushleyeva, & Lee, 2004), or activation (Altmann & Trafton, 2002), or by bottom-up factors, such as selective attention (Patsenko & Altmann, 2010). The continuum from a purely top-down strategy to a purely bottom-up selection strategy represents one slice of the problem space of how individuals decide where to allocate attention next. Determining which strategies participants use has theoretical and practical implications in training, the design of realistic simulations of human behavior, and in the development of instruments that could facilitate multitasking.

We used the Adaptive Control of Thought - Rational (ACT-R) cognitive architecture (Anderson et al., 2004) to examine multitasking strategies. One of the primary benefits of using a cognitive architecture such as ACT-R is that it provides a formal framework for developing and testing strategy use in multitasking. We simulated the Air Force Multi-Attribute Task Battery (AF-MATB, Miller 2010), which is a commonly used multitasking environment that has been used to explore different aspects of multitasking, such as the hysteresis effect (Bowers et al., 2014; Kim, House, Yun, & Nam, 2019) and the relationship between performance and physiological measures (Splawn & Miller, 2013). We compared a continuum of models ranging from a purely serial top-down strategy to a purely ballistic strategy driven by bottom-up attention to a combination of the two. When comparing our simulation with behavioral data (Bowers et al., 2014), we found that the best fitting models used a mixture of top-down and bottom-up strategies.

#### Methods

# **Participants**

We tested our model against behavioral data 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 Site Junior Force Council) participated in the 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 designed to investigate multitasking behavior in tasks similar to some of those encountered while operating aircraft. Full details regarding the AF-MATB can be found in Miller et al. (2010; 2014). Participants monitored subtasks for scripted events and responded to those events with keyboard presses and a joystick. In Bowers et al. (2014), the subtasks included System Monitoring, Tracking, Communications, and Resource Management. In the System Monitoring subtasks, participants had a limited time (3 and 6 seconds) to press a key on the keyboard when a Light (color change) or Gauge (exceeding a y-axis threshold) malfunctioned, respectively. In the Tracking subtask, participants used a joystick to adjust the position of a randomly moving reticle. In the Communications subtask, participants listened for audio files, adjusted and submitted the frequency and channel if the audio matched the participant's callsign. In the Resource Management subtask, participants monitored fluid levels in two tanks and adjusted the state of 8 pumps to control the fluid levels.

Parameters underlying these events and the frequency of these events were controlled by the experimenter. Events were distributed pseudorandomly, such that the same events could not overlap. Events from other subtasks could occur concurrently. Difficulty was primarily determined by increasing the frequency of events, which was the case in Bowers et al. (2014), resulting in greater overlap between events for the Hard difficulty compared to the Easy difficulty.

# **ACT-R Model**

The ACT-R architecture consists of discrete modules (e.g., visual, auditory) that are acted upon by production rules (ifthen statements) that control behavior. Cognition manifests as information flows between the different modules.

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 timing and visual properties as the AF-MATB. 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.

**Core Model** The core model selected subtasks in a strictly serial (i.e., top-down) manner. The model responded to the subtasks primarily through ACT-R productions, given that participants generally receive training in the AF-MATB prior to participation (i.e., participants completed six training sessions at 2 hours each in Bowers et al. 2014), which suggests that the rules for detecting and responding were well-learned and practiced. See Figure 1 for a high-level overview of how the model works.

First, the model turned on pumps (in this case, pumps 1 to 6) in the Resource Management subtask, which every participant did in Bowers et al. (2014). Next, the model searched for subtasks serially (in this case, clockwise) using two productions: (1) find a visual-location to attend to and (2) move visual attention to that location. This serial search was the



Figure 1: A high-level diagram of how the ACT-R model interacts with the AF-MATB task. The colored boxes correspond to productions for the subtasks. Green = Resource Management (Res. Man.), Yellow = Communications (Comm), Orange = Tracking, Blue = System Monitoring, Boxes = processes, Diamonds = decision points.

main loop that brought the model's attention to each subtask.

If the model was attending to one of the Lights or Gauges in the System Monitoring subtask and there was a malfunction, then the model responded by pressing the appropriate key on the keyboard with the left hand index finger.

If the model attended the Tracking subtask reticle, then the model moved the cursor towards the reticle, with the cursor simulating the behavior of a joystick by adding a constant x and y value in the direction of the center of the tracking panel to the reticle. The model's right hand was kept on the mouse.

If the model attended to one of the tank levels in the Resource Management subtask and the tank level was either too high or too low (i.e. 100 L), then the model cycled attention through the pumps with the intention of turning on pumps if the level was too low or turning off pumps if the level was too high. If too high, then the model checked and turned off pump 2 or 4 to slowly decrease the tank level. If too low, then the model checked and turned on pumps 1, 5, then 2 or 3, 6, then 4 to increase the level.

While attending to the above subtasks, the model listened for audio. If audio was played and started with the correct callsign, then the model stored the upcoming channel and frequency information in declarative memory. The next model production attempted to retrieve the channel and frequency from declarative memory. Successful retrieval of this memory switched the model's attention towards the Communications subtask and moved the model's left hand to the leftarrow key such that the model could reach the relevant keys. If necessary, the model first adjusted the channel using the up arrow key. Then, if necessary, the model adjusted the frequency using the left and right arrow keys. Once the frequency was correct, the model pressed the return key to submit the response and moved the left hand back to the 5 key to be able to reach keys for the System Monitoring and Resource Management subtasks.

Errors manifested in a few different ways. In the System Monitoring and Resource Management subtasks, errors could occur in two ways: (1) when the model initiated the production to press the key as the malfunction returned to normal automatically (3 seconds for Lights and 6 seconds for Gauges) and (2) there was motor noise (see below) such that the model could press the wrong key. If the model pressed the incorrect key, then the attended subtask would still be in a malfunctioned state and the model would attempt to press the correct key again. In the Tracking subtask, we assumed there was noise in the motor movements, which was modeled by activating the ACT-R cursor-noise parameter. In the Communication subtask, the model could fail to retrieve the stored chunk corresponding to the channel and frequency if the declarative memory was not sufficiently activated during retrieval.

#### **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). Given that malfunctions maintained in their state until corrected, we set the visual-onset-span parameter to 3.0 seconds, which represented the model being able to detect the malfunction after it had occurred in the model's peripheral vision.

For the Communications subtask, the model stored and retrieved from declarative memory. We set base-level learning (bll) to the recommended level (0.5). To achieve a retrieval rate that was approximately the same average as the behavioral data in Bowers et al. (2014) (approximately 92%), we altered the activation noise (ans = 0.2), base-level constant (blc = 2), and retrieval threshold (rt = 2.9) based on a gridbased search of a partial dataset from one of the participants in Bowers et al. (2014).

In simulating the Tracking subtask, we activated the incremental-mouse-moves parameter to more realistically capture joystick behavior. In addition, we activated the cursor-noise parameter to add motor movement noise. In simulating the joystick, the mouse cursor position in Cartesian coordinates were converted into Polar coordinates. The radius was multiplied by 0.125, then converted back into Cartesian coordinates. The resulting x and y values were capped at 10 pixels given physical limitations in joystick movements. The x and y were added to the Cartesian coordinates of the reticle each update.

Given the false alarm rate in System Monitoring (4.4% in Bowers et al.) and the percentage of times participants turned on/off pumps that took the fluid level away from the intended direction (9.2%), the model randomly pressed a key (F1-F6 and 1-8) on 5.0% of responses when responding to System Monitoring and Resource Management subtasks.

### **Strategy Space**

Here, we introduce reactive, event-driven strategies driven by bottom-up selection. In these models, the sequential selection process was interrupted if the model noticed a malfunction. That is, instead of the next subtask in the sequence determined by clockwise position, the model would attend to a different subtask. Each variant allowed interruption from a specific subset of the subtasks. Finally, we included a purely reactive variant of the model that only responded to tasks when malfunctions were passively noticed.

The different variants were referred to by which subtask had bottom-up selection (L = Lights and G = Gauges in System Monitoring, R = Resource, T = Tracking, N/A = for no bottom-up selection, and Only LGRT for only bottom-up selection and no top-down strategy)<sup>1</sup>. In total, there were 17 different variants (N/A, L, G, R, T, LG, LR, LT, GR, GT, RT, LGR, LGT, LRT, GRT, LGRT, and Only LGRT).

**Bottom-up Selection Simulation** We simulated bottom-up selection by using an ACT-R feature called "buffer stuffing", which is when the visual-location buffer is automatically populated with the location of a new stimulus instead of the model needing to search for a stimulus to add to the visual-location buffer (i.e., skipping the "find location" production in Figure 1). Note that following bottom-up selection, the next subtask would continue clockwise from the currently attended subtask and not from the previous subtask.

For the System Monitoring subtasks, "buffer stuffing" occurred any time there was a malfunction. For the Tracking subtask, "buffer stuffing" occurred when the tracking object was 27.5 pixels away from the origin. For the Resource Management subtask, "buffer stuffing" occurred when the tank levels were 700 L above or below the middle of the tank. These thresholds were tested on partial datasets to ensure they improved performance for that subtask.

Overall, as intended, each subtask had improved performance when that subtask had bottom-up selection (Table 1), indicating that the bottom-up selection was prioritizing that subtask.

Table 1: Average measures with and without bottom-up selection when that subtask has bottom-up selection active for all of the trials. Sys. Mon. = System Monitoring

Subtask	With	Without
Sys. Mon. Accuracy	0.76	0.66
Sys. Mon. RT	1.76	2.51
Tracking	36.06	46.73
Resource	181.74	230.73

#### **Performance Measures**

**Trial Simulation** We simulated the same number of participants (n = 16) and trials (t = 12) in Bowers et al. (2014). We used the same event lists generated from the participants, given that the event numbers differed between participants in

<sup>&</sup>lt;sup>1</sup>There was no bottom-up selection for the Communication subtask since the model already switches to that task as soon as it can

the Hard difficulty. The transitions from Easy to Hard (6 trials) and Hard to Easy (6 trials) were counterbalanced. Two of the participants did not fully complete a trial, so the total number of simulations was 190 trials for each of the 17 different variants.

**Performance Evaluation** We evaluated the model variants by calculating accuracy (accuracy = correct / total) and reaction time for correct responses in the System Monitoring and Communications subtasks. For the Tracking subtask, we averaged the Euclidean distance of the reticle to center across the trial. For the Resource Management subtask, we averaged the deviations of both tank levels across the trial.

In our model comparison to select the best model, we accounted for the difference in scales in the dependent measures from each subtask, *n*, by computing a mean normalized root mean square errors (NRMSE):

NRMSE = 
$$\frac{1}{n} \sum_{i=1}^{n} \frac{\sqrt{(\hat{x}_i - \bar{x}_i)^2}}{\bar{x}_i}$$

with  $\hat{x}$  being the mean predicted measure,  $\bar{x}$  being the mean observed DV, and *i* being the index for subtask. Given the division of the RMSE by the mean from the behavioral data, values closer to 0 indicate less error between the model and behavioral data.

After selecting the best model using NRMSE, we conducted a Bayesian repeated measures ANOVA using JASP (JASP Team, 2022) to analyze the effect of difficulty (Easy vs Hard) for the model and the behavioral data (Bx) to determine if the model is qualitatively showing the same effect of difficulty as the participants. Then, we conducted a Bayesian mixed factors ANOVA with difficulty (repeated: Easy vs. Hard) and group (Model vs. Bx) to see if the effect of difficulty was the same for the model and behavioral data.

# Results

We evaluated the model's performance for the dependent measures. In the figures below, the ordering of the model variants is based on ascending NRMSE. Based on NRMSE, the best models included bottom-up selection for one of the System Monitoring subtasks (Light or Gauge) and Resource Management subtasks. Specifically, the LR (NRMSE: 0.054) and R (NRMSE: 0.11), and GR (NRMSE: 0.14) models performed best.

The purely top-down model (N/A, NRMSE: 0.17) performed better than the model with solely bottom-up selection (Only LGRT, NRMSE: 0.37). This difference was largely driven by the very poor performance in the Resource Management subtask for the Only LGRT model, which was likely because the interruptions in the other subtasks took attention away from the Resource Management subtask.

There was a wide range of System Monitoring performance (see Figure 2). As indicated above, including either Light or Gauge bottom-up selection increased System Monitoring subtask accuracy and decreased reaction time. The LG (NRMSE: 0.33) variant had the closest System Monitoring accuracy to the behavioral data, which was expected given that the model attended more frequently to the System Monitoring subtask, but was a poor fit otherwise.



Figure 2: Simulation of the System Monitoring subtask for the different models (y-axis) and behavioral data (Bx, in red). System Mon. = System Monitoring

The Communications subtask simulation had the least amount of variability (see Figure 3). This was expected given there is no bottom-up selection to affect performance and once in the Communication subtask, the model was not interrupted by other subtasks. Some of the variance in the behavioral data suggests that some participants may have interleaved this subtask with other subtasks.



Figure 3: Simulation of the Communications subtask for the different models (y-axis) and behavioral data (Bx, in red). Comm. = Communications

Tracking performance tended to be precise (see Figure 4 Left), even without bottom-up selection. If there was bottomup selection in the Resource Management but not Tracking subtask, then the Tracking performance was significantly less precise but closer to the behavioral data. This likely occurred because of the clockwise search pattern of the model, which could effectively skip the Tracking subtask if the model attended to the Resource Management subtask.

The model tended to be imprecise in the Resource Man-



Figure 4: Simulation of the Tracking (Left) and Resource Management (Right) subtasks for the different models (y-axis) and behavioral data (Bx, in red). Px = Pixels

agement subtask (see Figure 4 Right). This may be due to the specific strategy of the current model. For example, participants tended to change the status of the pumps on average 53.7 times (sd = 12.3), whereas the model ranged from 40.7 (LR, best fit) to 27 (LGT, worst fit) changes.

#### The Effect of Task Difficulty

We evaluated the best fitting model's (LR) ability to show the task difficulty effect found in Bowers et al. (2014). In Table 2, we show descriptive results for the effect of difficulty and the Bayes Factor when difficulty was added to the null model. There was decisive evidence for the inclusion of difficulty in the model and behavioral comparisons (BF > 421.6) except for the following: for Communications subtask accuracy, there was moderate evidence for not including difficulty for the model (BF = 0.436) and moderate evidence for including difficulty for the behavioral data (BF = 2.6), and there was anecdotal evidence for including difficulty for Resource Management subtask (BF = 5.2). These results overall show the model captured the qualitative worsening of performance when the task became more difficult.

When quantitatively comparing the model to the behavioral data and the effect of difficulty using Bayesian mixed factor ANOVAs, there was decisive evidence for the effect of difficulty (BF > 1.1e + 4) except for Communications subtask accuracy (BF = 0.43), which was expected given the lack of a difference in Table 2. Evidence was leaning towards the null hypothesis for the effect of group (BF < 0.69) except for anecdotal and decisive evidence for the effect of group in System Monitoring subtask accuracy (BF = 1.9) and reaction time (BF = 5.1e + 2), respectively. This was expected given the minimal error between the best fitting model and the behavioral data. Interestingly, there tended to be decisive evidence for an interaction term (BF > 231.9) except for Communication subtask reaction time (BF = 58.1) and accuracy (BF = 2.4), which had very strong and anecdotal evidence respectively. These results suggest that the predicted magnitude of the effect of difficulty differed from that of the participants. Visual inspection of the table suggests that the model tended to be more accurate and faster on the Easy difficulty and less accurate and slower on the Hard difficulty.

Table 2: Average values for Easy and Hard difficulty for the model (M) and behavioral data (Bx). The Bayes Factor (BF) for including difficulty in a Bayesian repeated measures ANOVA is also shown, with higher values indicating evidence for the difficulty factor.

Subtask	Easy(m, sd)	Hard(m, sd)	BF
M: SysMon. Acc.	0.96(0.01)	0.68(0.03)	2.7e23
Bx: SysMon. Acc.	0.93(0.06)	0.79(0.1)	4.9e5
M: SysMon. RT	1.63(0.05)	2.28(0.05)	2.4e23
Bx: SysMon. RT	1.57(0.2)	1.98(0.1)	1.3e10
M: Comm. Acc.	0.92(0.06)	0.94(0.02)	0.463
Bx: Comm. Acc.	0.97(0.1)	0.92(0.1)	2.6
M: Comm. RT	8.4(0.11)	8.7(0.07)	5.4e6
Bx: Comm. RT	8.32(1.0)	9.21(1.2)	421.6
M: Tracking	26.1(2.3)	93.2(23.7)	2.1e10
Bx: Tracking	42.4(13)	81.8(17.8)	2.1e9
M: ResMan	85.5(4.9)	306(53.8)	3.6e14
Bx: ResMan	173.1(72)	221(97.0)	5.2

# **General Discussion**

Multitasking has captured the interest of researchers because it provides a rich environment for understanding the strategies people use to manage and prioritize multiple competing goals. We contributed to the understanding of strategy use in multitasking by comparing a continuum of strategies ranging from purely top-down (i.e. selecting tasks in a fixed order) to purely bottom-up (i.e., only selecting tasks that malfunctioned or changed). These strategies were instantiated in the ACT-R cognitive architecture in order to test their predictions quantitatively. Overall, we found that the best fitting model was neither using a strict top-down nor bottom-up selection strategy. Instead, the best model used a mixture of the two. That is, the model serially searched, but the serial search could be interrupted if a malfunction was detected in the model's peripheral vision.

Our simulation suggests a few things. First, these results indicate that task switching in the AF-MATB was largely driven by top-down strategies. While the best fitting model had bottom-up selection for two of the subtasks, the majority of the models we simulated with top-down strategies performed adequately in capturing the behavioral data. These findings corroborate other findings in the literature that suggest top-down factors, such as task instruction (Lehle & Hübner, 2009), affect and alter task performance, which highlights the needs for a better understanding of strategy use in multitasking. A better understanding of how strategies affect performance could result in the development of strategies and instruments that improve multitasking performance. Second, it suggests that a model in which tasks are completed in a serial fashion provides a satisfactory account of multitasking in the AF-MATB. This is consistent with prior work in which individuals opt for an effortful, serial strategy instead of a parallel approach to improve performance on some tasks (Lehle, Steinhauser, & Hübner, 2009). It is possible that an account in which task processing overlaps to a greater degree (e.g. Salvucci and Taatgen, 2008) could also provide a satisfactory account. Further research is needed to determine the extent to which concurrent processing is needed to account for performance in the AF-MATB and whether the AF-MATB is sensitive enough to distinguish these accounts.

There were limitations. First, the magnitude of the effect of difficulty was significantly different from the behavioral data. One explanation is that participants may be switching their strategy when the task became more difficult, such as using a bottom-up selection strategy when the task was difficult and a top-down strategy when the task was easy. Second, strategies likely differ between participant. It may be that some participants only serially searched while others only used bottom-up selection. There is likely not enough data in Bowers et al. (2014) to determine if this is the case or not.

# 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. The contents have been reviewed and deemed Distribution A. Approved for public release. Case number: AFRL-2022-1713.

#### References

- Altmann, E. M., & Trafton, J. G. (2002). Memory for goals: An activation-based model. *Cognitive science*, 26(1), 39–83.
- 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.
- 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).
- Fischer, R., & Plessow, F. (2015). Efficient multitasking: parallel versus serial processing of multiple tasks. *Frontiers in psychology*, *6*, 1366.
- JASP Team. (2022). JASP (Version 0.16.1)[Computer software]. Retrieved from https://jasp-stats.org/
- Kim, N. Y., House, R., Yun, M. H., & Nam, C. S. (2019). Neural correlates of workload transition in multitasking: An act-r model of hysteresis effect. *Frontiers in Human Neuroscience*, 12, 535.
- Koch, I., Poljac, E., Müller, H., & Kiesel, A. (2018). Cognitive structure, flexibility, and plasticity in human multitasking—an integrative review of dual-task and task-switching research. *Psychological bulletin*, *144*(6), 557.

- Kushleyeva, Y., Salvucci, D. D., & Lee, F. J. (2005). Deciding when to switch tasks in time-critical multitasking. *Cognitive systems research*, 6(1), 41–49.
- Lehle, C., & Hübner, R. (2009). Strategic capacity sharing between two tasks: Evidence from tasks with the same and with different task sets. *Psychological Research PRPF*, 73(5), 707–726.
- Lehle, C., Steinhauser, M., & Hübner, R. (2009). Serial or parallel processing in dual tasks: What is more effortful? *Psychophysiology*, 46(3), 502–509.
- Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part i. basic mechanisms. *Psychological review*, 104(1), 3.
- Miller, W. D. (2010). The us air force-developed adaptation of the multi-attribute task battery for the assessment of human operator workload and strategic behavior. (Tech. Rep. No. AFRL-RH-WP- TR-2010-0133). Retrieved from the Defense Technical Information Center website: http://apps.dtic.mil/dtic/tr/fulltext/u2/a537547.pdf.
- Miller, W. D., Schmidt, K. D., Estepp, 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.
- Patsenko, E. G., & Altmann, E. M. (2010). How planful is routine behavior? a selective-attention model of performance in the tower of hanoi. *Journal of Experimental Psychology: General*, 139(1), 95.
- Payne, S. J., Duggan, G. B., & Neth, H. (2007). Discretionary task interleaving: heuristics for time allocation in cognitive foraging. *Journal of Experimental Psychology: General*, 136(3), 370.
- Salvucci, D. D., Kushleyeva, Y., & Lee, F. J. (2004). Toward an act-r general executive for human multitasking. In *Iccm* (pp. 267–272).
- Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: an integrated theory of concurrent multitasking. *Psychological review*, 115(1), 101.
- Salvucci, D. D., Taatgen, N. A., & Borst, J. P. (2009). Toward a unified theory of the multitasking continuum: From concurrent performance to task switching, interruption, and resumption. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 1819–1828).
- Smith, M. R., Lewis, R. L., Howes, A., Chu, A., Green, C., & Vera, A. (2008). More than 8,192 ways to skin a cat: Modeling behavior in multidimensional strategy spaces. In *Proceedings of the 30th annual conference of the cognitive science society* (pp. 1441–1446).
- Splawn, J. M., & Miller, M. E. (2013). Prediction of perceived workload from task performance and heart rate measures. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 57, pp. 778–782).

# Specificity of the Jumping-to-Conclusion Bias in Social Anxiety: An Account Using the Bayesian Computational Modelling Approach

Nicole Tan (nicole.tan@anu.edu.au) Research School of Psychology, The Australian National University ACT, 2602 Australia

# **Yiyun Shou**

#### Junwen Chen

#### **Bruce Christensen**

#### Abstract

To date, little is known about the role of social anxiety in the assignment of evidence weights which could contribute to the jumping-to-conclusion bias. The present study used a Bayesian computational method to understand the mechanism of jumping-to-conclusion bias in social anxiety, specifically through the assignment of weights to information sampled. The present study also investigated the specificity of the jumping-to-conclusion bias in social anxiety using three variations of beads tasks that consisted of neutral and socially threatening situations. A sample of 210 participants was recruited from online communities to complete the beads tasks and a set of questionnaires measuring the trait variables including social anxiety and the fears of positive and negative evaluation. The Bayesian model estimations indicated that social anxiety and fears of evaluation significantly biased the assignment of evidence weights to information received in certain conditions of the beads tasks. Our results indicated that social anxiety and fear of evaluation could influence belief updating depending on situations. However, the influences from these trait variables seemed to be insufficient in contributing to the jumping-to-conclusion bias.

**Keywords:** belief updating; jumping to conclusion bias; beads tasks; Bayesian computational modelling; reasoning bias; social anxiety; fears of evaluation

#### Introduction

Biases in information processing are common in psychopathologies including psychosis, anxiety disorders, and depression (Beck & Clark, 1997; Garety et al., 2011; Leppänen, 2006). In recent years, there is an increasing interest in establishing a type of reasoning bias, the Jumping-to-Conclusion bias (JTC), as a transdiagnostic factor underlying mental disorders. The JTC bias refers to the tendency to make hasty decisions (Garety et al., 1991). Whilst the JTC bias is prevalently studied in the context of delusions, some studies have also found that clinically anxious populations exhibit the JTC bias (Bensi & Giusberti, 2007; Giusti et al., 2018; Lincoln et al., 2010). However, several meta-analyses have suggested that the relationships between the JTC bias and delusions as well as anxiety are inconclusive given the heterogeneity in effect sizes (Dudley et al., 2016; Ross et al., 2015; So et al., 2016).

The classic beads task is the gold standard for measuring the JTC bias (Huq et al., 1988). In this task, participants are shown two jars with opposing ratios of beads. Participants are told that one of the jars is randomly chosen and beads are drawn out of the chosen jar (Huq et al., 1988). Participants can request to see as many beads as they wish before deciding the source of beads being drawn. Unbeknownst to participants, the sequence of beads is predetermined. Individuals who reach a decision with fewer than two beads are typically considered jumping to conclusions (Garety et al., 1991; Huq et al., 1988). Although viewing a neutral stimulus such as a bead could result in an extreme responding style in individuals with delusions, the same may not be true for individuals with high anxiety levels. Cognitive models of anxiety postulate that biases in anxiety are only triggered in the presence of a perceived threat that is congruent with the anxiety subtype (Beck & Clark, 1997; Heinrichs & Hofmann, 2001; Rapee & Heimberg, 1997). Thus, individuals with high levels of anxiety may not exhibit the JTC bias when the classic beads task is used to assess this reasoning bias since the task does not involve threat cues.

Schlier et al. (2016) explored the specificity of the JTC bias amongst individuals with social anxiety disorder. They compared the decisional aspects between the classic beads task which involves viewing beads and the social beads task which contains social information about neutral and social situations involving self-relevant and delusion-relevant threats (Westermann et al., 2012). The clinical and healthy samples behaved similarly in the social beads task, but individuals with social anxiety disorder requested significantly more beads than the healthy controls in the classic beads task (Schlier et al., 2016). Whilst this may suggest that social anxiety is not associated with the JTC bias, it is arguable that the social situations presented may not necessarily tap into the cognitive biases in social anxiety. For example, the self-relevant scenarios have specified the outcome of the scenario, such as by asking "Which waitress made a critical comment about you?" or "Which group is bored by your talk?". In these cases, the individuals only need to collect information to determine the source of critical comment, rather than using the

information to interpret whether the scenario is threatening or not. Furthermore, the self-relevant social beads task did not consistently yield acceptable convergent validity with the classic beads task across different levels of this content class (Westermann et al., 2012). Thus, this suggests that the social beads task may not be a sufficient paradigm to capture the JTC bias in social anxiety and may not be comparable to the classic beads task.

Cognitive models of delusion and anxiety have emphasised threat-processing biases, such that individuals with higher levels of delusion ideation and anxiety are prone to using more threat-congruent or belief-confirming information to update their beliefs compared to nonthreatening and contradictory information (Bell et al., 2006; Müller-Pinzler et al., 2019; Speechley et al., 2012). Research has also shown that individuals with social anxiety disorder do not exhibit a tendency to interpret information in a positive light, and thus lack positivity bias compared to healthy controls (Chen et al., 2020; Koban et al., 2017). Following this notion, individuals with higher levels of delusion and anxiety may assign more weight to some types of information than others. A biased assignment of evidence weights could promote a higher rate of belief updating within minimal pieces of information, thereby contributing to the JTC bias. This speculation remains to be tested in the context of anxiety, especially social anxiety.

Recent research has also shown that individuals with higher social anxiety experience not only higher fears of negative evaluation but also positive evaluation (Button et al., 2015; Fredrick & Luebbe, 2020; Weeks & Howell, 2012). Social anxiety is associated with a feeling of apprehension about being evaluated both unfavourably and favourably in social situations due to the tendency to overestimate the probability and stake of social threats and the fear of increasing expectations from others following a good performance (Dryman & Heimberg, 2015; Weeks & Howell, 2012). Studies have found that the fears of evaluation are associated with a perception that social events and outcomes are threatening and negative regardless of the valence of feedback received, although these findings are yet to be consistent (Alden et al., 2008; Button et al., 2015; Dryman & Heimberg, 2015). Considering that fears of evaluation are key cognitive features of social anxiety, the fears of evaluation could be an underlying factor explaining the relationship between social anxiety and the JTC bias, given that they are self-defeating beliefs associated with catastrophic social outcomes (Heinrichs & Hofmann, 2001). Thus, due to the higher fears of evaluation, individuals with higher social anxiety could assign heavier weights to social information in favour of negative social outcomes, regardless of whether the information is positive or negative (Alden et al., 2008). To the best knowledge, the role of fears of evaluation has not been investigated in the context of JTC bias in social anxiety.

Given the existing gaps, the present study aims to introduce two variations of the beads task to investigate the relationship between social anxiety, fears of evaluation, and the JTC bias. One variation involves viewing verbal feedback ("good" and "bad") about one's performance for a hypothetical presentation. The binary social outcomes in this task are either the individual has done a good presentation (positive social outcome) or a poor presentation (negative social outcome). This theme is consistent with the core of social anxiety as it concerns the positive and negative evaluations from the audience (Chen et al., 2020; Heinrichs & Hofmann, 2001; Rapee & Heimberg, 1997). Another variation was designed to assess the JTC bias in a social neutral situation, whereby individuals were required to decide which club in a hypothetical college had been chosen to host an event based on a sequence of gender information presented. This scenario involves social elements but is void of any social threats. These variations retained all original characteristics of the classic beads tasks for better comparisons.

# The Bayesian Model

The present study is the first study to apply the Bayesian computational model developed by Tan et al. (2022) to understand the mechanism underlying the JTC bias in specific situations. This investigation focuses on the role of a trait variable in influencing the assignment of evidence weights as a factor that could contribute to the JTC bias. The model uses the variables measured in the beads tasks including the number of draws to decisions, trial-to-trial certainty about the source of information presented, and the final decision about the source of information to estimate the influence of a psychopathological trait on the assignment of evidence weights to the binary information sampled (see Figure 1). This model also assumes that the Bayesian belief updating is the normative belief updating, similar to early studies that investigated reasoning styles in clinical populations (Garety et al., 1991; Huq et al., 1988).

In this model, the most frequently occurring information from the selected jar is termed *dominant information* whereas the least frequently occurring information from the selected jar is termed *secondary information*. The dominant and secondary information is allowed to have its individual evidence weights,  $W_D$  and  $W_S$ , which are determined by the individual's trait variable. The parameters  $ba_D$  and  $ba_S$  are of interest as these parameters capture the effect of the individual's trait level on the assignment of evidence weights. The evidence weights assigned to the information sampled influence the rate of belief updating. This is reflected in *k*th individual's reported subjective certainty about the source of information on a trial-to-trial basis. The Bayesian model assumes that the prior belief follows a beta distribution with an uninformative prior.

Once the individual's evidence accumulated for a particular hypothesis exceeds the set threshold of log of Bayes factor 3, the model assumes that the individual would stop sampling. This means that the individual is now able to reach a decision about the source of information drawn.





*Note.* The shaded nodes represent observable variables, and the unshaded nodes represent unobservable variables. Square nodes represent discrete variables and round nodes represent continuous variables. The double-edged nodes are deterministic nodes, and the single-edged nodes are stochastic variables.

# Hypotheses

The present study is the first study to systematically investigate the mechanism of JTC bias in social anxiety and fears of evaluation using the Bayesian computational method and a novel variation of beads task that taps into the core of social anxiety.

Based on the existing models of social anxiety, it was hypothesised that individuals with higher levels of social anxiety, fears of negative and positive evaluation will place heavier weights on both negative and positive feedback sampled in the social anxiety beads in favour of a negative social outcome, i.e., performing poorly (Heinrichs & Hofmann, 2001; Rapee & Heimberg, 1997). It was also hypothesised that these traits will not be significantly associated with the evidence weights assigned in the classic beads task and social neutral beads task.

# Method

# **Participants**

A total of 210 participants responded to the study via an online crowdsourcing platform, Prolific. The final data analysis included 169 participants (excluded 39 participants for failing comprehension checks for the beads tasks, one withdrew, and one detected as a potential bot). The final sample consisted of 51% females with a mean age of 40.93 (SD = 13.47), and 23.7% reported having a diagnosis of mental disorder(s).

# Materials

**Beads Tasks** A computerised version of the beads tasks was constructed using the Qualtrics<sup>TM</sup>. The instructions for the beads tasks were similar to the original version of the beads tasks reported by Garety et al. (1991) and Huq et al. (1988). The classic beads task involved viewing two jars of coloured beads with opposing beads ratios whereas the social neutral beads task involved viewing two clubs with opposing ratios of gender information. Meanwhile, the social anxiety beads task consisted of two audiences with opposing ratios of feedback for a presentation.

Each task comprised of two beads ratios; 55:45 represented the highest uncertainty and 90:10 reflected the lowest uncertainty within the task. The sequence of information for the 55:45 ratio was randomly generated once and fixed for all participants whereas the 90:10 sequence was derived from Moritz and Woodward (2005). Each task also consisted of two sequences in which the dominant information was manipulated. For example, in the classic beads task, Sequence 1 may have mostly red beads and Sequence 2 may have mainly blue beads. On the other hand, Sequence 1 in the social neutral beads task may have mostly male as the gender information and Sequence 2 may have mainly female as the gender information. For the social anxiety beads task, Sequence 1 involved mainly positive social feedback whereas Sequence 2 involved mainly negative social feedback. Altogether there were 12 versions of the beads task to complete.

After the first piece of information was presented, participants could either terminate the trial and report their decision about the source of information drawn so far or continue sampling more information until they reached a decision. All previously drawn information was shown on the screen as a memory aid. Participants also had to report their certainty level about the source of information drawn after seeing a new piece of information. Participants could request to see a maximum of 20 pieces of information. If a decision was not reached after the 20<sup>th</sup> draw, they would be prompted to make their decision and the trial would automatically terminate. The presentation of beads ratios, sequences, and types of beads task was randomised, and the "correct" decision for each task was pseudo-randomised as well.

**Trait Measures** There were six measures included in the study, which assessed psychotic-like experiences, social anxiety, positive and negative impression management, as well as the fears of negative and positive evaluation. For the purpose of current aims, only results concerning social anxiety and fears of evaluation would be reported.

The Social Interaction Anxiety Scale and Social Phobia Scale (SIAS-6 & SPS-6) were used to assess trait anxiety associated with social interaction and fear of scrutiny (Peters et al., 2012). The Cronbach's alpha for the combined scales was .94. The Brief Fear of Negative Evaluation (BFNE) consisted of 12 items measuring the fear of negative evaluation (Leary, 1983). Only the straightforward items were included in the present data analysis following the recommendation by Weeks et al. (2005). The Cronbach's alpha for the 8-item BFNE was .96 for this sample. The Fear of Positive Evaluation Scale (FPES) was used to measure the fear of positive evaluation (Weeks et al., 2008). Only straightforward items were included in the present analysis. The Cronbach's alpha of the 8-item FPES was .89 for the present sample.

# Procedure

Participants gave informed consent at the beginning of the study and completed a series of demographic questions about their age, gender, ethnicity, education, English proficiency, and history of mental health. Then, participants completed 12 classic, social neutral, and social anxiety beads tasks. Finally, participants responded to the set of questionnaires measuring trait variables that assessed delusion ideation, social anxiety, fears of evaluation, and impression management. Participants were debriefed at the end of the study and were provided with links to mental health resources. Participants were compensated £3.45 for completing the study which took about 35 minutes.

# Results

The Bayesian model estimations were performed using "R2jags" package (Su & Yajima, 2021) on R version 3.6.2. For each model estimation, Markov Chain Monte Carlo four-chain processing was run with 10000 samples drawn from the posterior distributions and the first 1000 steps being discarded. Each model only included one trait variable at a time. The scores of SIAS-6 & SPS-6 (mean = 13.45, *SD* = 11.78), BFNE (mean = 21.64, *SD* = 8.91), and FPES (mean = 31.07, *SD* = 16.84) were standardised so that a standard deviation increase in these trait variables is associated with a standard deviation change in evidence weights assigned. The decision threshold in the model was fixed at the log of Bayes factor 3 which represented having substantial evidence supporting a particular hypothesis.

Each model was examined for its convergence, such that the model was said to have achieved convergence when the resulting  $\hat{R}$  was less than 1.1 (Su & Yajima, 2021). All model estimations converged for this study with the largest  $\hat{R}$  for a model estimation being 1.008, indicating that the results are reliable and interpretable (Gelman & Rubin, 1992). The more complex models that included a trait variable had lower DIC values compared to the simpler models that did not include a trait variable. This indicates that the more complex models had a better model fit than the simpler models. The summarised results are based on a minimum of 9000 samples averaged over four chains.

When mainly negative feedback was presented in the social anxiety beads task, the estimations indicated that

participants generally placed significantly heavier weights on negative feedback in favour of the source that has a lower ratio of negative feedback (estimate = -1.36, 95%CI[ -1.40, -1.32]) and significantly heavier weights on positive feedback in favour of the source that has a lower ratio of positive feedback (estimate = 1.54, 95% CI[1.50, 1.57]). Meanwhile, when mainly positive feedback was presented, significantly heavier weights were assigned to positive feedback in favour of the source that has a lower ratio of positive feedback (estimate = -.83, 95%CI[-.87, -.79];) and negative feedback was evaluated in favour of the source that has a lower ratio negative feedback (estimate = .93, 95% CI[.89, .98]). Thus, participants were generally slower in updating beliefs compared to a rational Bayesian agent as they placed heavier emphasis on the dominant information in favour of the less probable source of information, regardless of the sequences of feedback presented.

The model estimations further indicated that when mainly negative feedback was presented, higher fear of positive evaluation (FPE) was associated with significantly heavier weights assigned to negative feedback in favour of the source that has a lower ratio of negative feedback; and significantly heavier weights assigned to positive feedback in favour of the source that has a lower ratio of positive feedback. This suggested that higher FPE might promote slower belief updating about poor performance. Social anxiety and fear of negative evaluation (FNE) did not significantly bias the assignment of evidence weights (see Table 1). Meanwhile, when presented with mainly positive feedback, individuals with higher levels of social anxiety and FNE placed significantly heavier weights on positive feedback in favour of the source that has a higher ratio of positive feedback. Thus, higher levels of social anxiety and FNE promoted faster belief updating about performing well. No further significant influences from FPE were observed.

In the neutral beads tasks, participants generally placed significantly heavier weights on the dominant information in favour of the incorrect source which has a lower ratio of the dominant information (estimate = -1.13, 95%CI[-1.16, -1.10] for classic beads task; estimate = -1.26, 95%CI[-1.29, -1.23] for social neutral beads task). Significantly heavier weights were also assigned to secondary information in favour of the correct source which has a lower ratio of the secondary information in both classic beads task (estimate = 1.28, 95%CI[1.25, 1.32]) and social neutral beads task (estimate = 1.28, 95%CI[1.25, 1.32]) and social neutral beads task (estimate = 1.41, 95%CI[1.38, 1.44]). In other words, a similar trend of more cautious belief updating compared to a rational Bayesian agent was observed in neutral beads tasks.

Higher levels of social anxiety reduced the general overcautiousness in belief updating in the neutral beads tasks (see Table 1). Whilst FNE did not significantly influence evidence weighting, higher levels of FPE showed mixed influences on evidence weighting in neutral beads tasks. The results suggested that higher levels of FPE might promote faster belief updating in the classic beads task by assigning heavier weights to dominant information in favour of the correct source of information but did not significantly influence the assignment of evidence weights in the social neutral beads task.

Table 1: Model estimations of biases in weighting evidence

Interestingly, the model estimations indicated that participants were weighing both dominant and secondary information equally when the uncertainty level was the highest at 55:45 across all three variations of the beads tasks (see Figure 2). Social anxiety, FNE, and FPE also did not significantly influence the weighting of dominant and secondary information at this ratio. The biases in weighting information became more prominent when the uncertainty level was the lowest at the 90:10 ratio (see Figure 3). Participants demonstrated slower and more cautious belief updating compared to a rational Bayesian agent for this ratio.

To summarise, the hypothesis about the influences of social anxiety, FNE, and FPE in the social anxiety beads task was not supported. The hypothesis about insignificant associations between these trait variables and evidence weighting in the classic and social neutral beads tasks was partially supported.



Figure 2: A comparison of belief updating rates between human and Bayesian agents for 55:45 ratio across classic, social anxiety, and social neutral beads tasks. The subjective certainty at the point of decision was averaged across 169 human participants. The Bayesian certainty was generated using the basic Bayes theorem assuming no error rates and unbiased evidence weighting.



Figure 3: A comparison of belief updating rates between human and Bayesian agents for 90:10 ratio across classic, social anxiety, and social neutral beads tasks.

Variable	Dominant		Secondary	
	information		information	
	Estimate	95%CI	Estimate	95%CI
CLBT				
SA	.07	[.05, .10]	06	[09,03]
FNE	01	[04, .02]	.03	[.00, .06]
FPE	.04	[.01, .06]	02	[05, .00]
SNBT				
SA	.10	[.07, .12]	07	[10,04]
FNE	.01	[02, .04]	.02	[.00, .05]
FPE	02	[05, .01]	.03	[.00, .06]
SABT+				
SA	.14	[.10, .17]	14	[18,11]
FNE	.11	[.07, .14]	12	[15,08]
FPE	.02	[02, .05]	03	[07, .00]
SABT-				
SA	02	[06, .02]	.00	[03, .04]
FNE	.03	[01, .07]	02	[06, .02]
FPE	10	[14,06]	.05	[.01, .09]

*Note.* CLBT = classic beads task, SNBT = social neutral beads task, SABT+ = social anxiety beads task with mainly positive feedback presented in a given full sequence, SABT- = social anxiety beads task with mainly negative feedback presented in a given full sequence, bold = significant estimations. The biases in weighting evidence are deemed significant if the 95% confidence intervals do not include a 0, which indicates even evidence weights for both sources of information.

### Discussion

The present study sheds new perspectives on how social anxiety, fears of negative evaluation (FNE) and positive evaluation (FPE) could influence decision-making across three variations of the beads tasks using the Bayesian modelling approach. The hypothesised specificity of social anxiety and fears of evaluation in biasing the assignment of evidence weights was partially supported. Higher levels of FNE did not significantly influence the assignment of evidence weights in the absence of threat cues when the classic and social beads tasks were used. This is in line with the theories proposed in cognitive models of anxiety. However, the effects observed in the social anxiety beads task were unexpected. Whilst there were no significant biases in weighting evidence due to social anxiety and FNE when the full sequence of information consisted of mainly negative feedback, higher levels of social anxiety and FNE were associated with significant biases in assigning weights to information when mainly positive feedback was presented. In this condition, higher levels of social anxiety and FNE promoted faster belief updating about the positive social outcome, i.e., performing well.

The effects of social anxiety, FNE, and FPE observed in the present study are inconsistent with previous research that found negative interpretation biases and a lack of positivity bias in social anxiety (Chen et al., 2020; Koban et al., 2017). Several reasons could explain these inconsistencies. Firstly, the social scenario presented was about a hypothetical presentation completed in front of two large audiences. Given the hypothetical nature of the task, participants may not have interpreted the task as particularly threatening. Participants could have made their decisions in this task the same way they would in any other neutral situation. The general cautiousness in belief updating across three variations of the beads tasks provides preliminary evidence supporting this notion.

The nature of beads tasks could also explain the observed positive bias with increasing levels of social anxiety and FNE in this study. The beads tasks only reflected a snapshot of behaviours rather than a sequence of behavioural trends. Individuals with high levels of social anxiety and FNE could be momentarily receptive to positive feedback as indications of good performance (Heinrichs & Hofmann, 2001; Koban et al., 2017). However, the processing of the positive feedback may be impacted by later biased information processing in social anxiety such as rumination after the social event, which triggers a subsequent negative affect (Rapee & Heimberg, 1997). Specifically, individuals with higher social anxiety could fear that they would fail to meet others' high expectations following a good performance, that is exhibiting fear of positive evaluation (Heinrichs & Hofmann, 2001; Weeks et al., 2008). Following this perspective, the present finding suggests that the effects of social anxiety and fears of evaluation may not be immediately evident following positive feedback and may be exacerbated after a period of time in conjunction with rumination. Further investigations are warranted to support this speculation.

The current findings also did not support the hypothesis that social anxiety and FPE would not significantly bias the assignment of evidence weights in the absence of threat cues. Across both classic and social neutral beads tasks, higher levels of social anxiety promoted faster belief updating about the correct source of information. This suggests that individuals with higher social anxiety may have a more efficient way of updating beliefs about the more probable outcome by placing a heavier emphasis on the dominant information to revise their beliefs. These trends of faster belief updating also suggest that higher social anxiety may drive a need to avoid making incorrect decisions, thus, contributing to a different way of evaluating information compared to individuals with lower social anxiety. Overall, the present findings suggest that higher social anxiety may be associated with a reduced overcautiousness in belief updating in both neutral and social situations in which one receives a lot of positive feedback. However, given the large magnitude of general biases in evidence weighting which steers towards overcautiousness, the opposing biases from social anxiety

and fears of evaluation in evidence weighting may be insufficient to outweigh the general overcautiousness. Hence, it is unlikely that social anxiety and fears of evaluations would significantly contribute to the JTC bias.

The present study has also deepened the current understanding of how individuals generally make decisions in the beads task. Based on the Bayesian model, it seems that individuals weigh information equally when the uncertainty level was the highest. Under this circumstance, individuals were updating their beliefs similarly to a rational Bayesian agent. However, when the uncertainty level is low, individuals became more conservative and deviated more from the Bayesian belief updating as they placed significantly heavier emphasis on dominant information in favour of the less probable outcome. This behaviour may reflect a general tendency to seek reassurance by gathering more information when the probable outcome was obvious.

Aside from uncovering the complexities in the relationship between social anxiety, fears of evaluation, and the jumping to conclusion bias, the present findings have important implications. This is the first study that applies the Bayesian model developed by Tan et al. (2022) to model real-life data concerning belief updating across different situations. This study shows promising results in terms of understanding the evidence weighting of binary information, given good model convergence and fitting. Future studies can consider applying this model across a wide variety of settings and populations to explore more factors, beyond social anxiety, that can influence evidence weighting and how they could contribute to the JTC bias. This can be achieved by replacing the trait variable parameter in the model with scores from any measures. Furthermore, longitudinal studies are warranted to test the speculation about the delayed effects of social anxiety and fear of evaluations in appraising positive feedback. Future studies could also consider inducing state social anxiety to investigate the causal effects of heightened social anxiety on the JTC bias.

The current findings are limited by the assumption that individuals perform Bayesian belief updating. The Bayes' theorem is often criticised as individuals generally do not reason like a Bayesian agent. However, the current findings focused on the role of social anxiety and fears of evaluation in exaggerating or reducing the deviations from the optimal Bayesian belief updating. Future studies are warranted to explore other belief updating and non-Bayesian models to investigate alternative explanations for the JTC bias.

To conclude, the present study suggests that social anxiety and fears of evaluation may be associated with the JTC bias given most of their significant influences on the assignment of evidence weights. These biases could occur depending on the situations such as when one receives a lot of positive feedback and in some neutral situations. However, the biases in evidence weighting in social anxiety and fears of evaluation may be too weak to outweigh the general tendency to be cautious in belief updating. Thus, these trait variables seem insufficient to in contributing to the JTC bias.

# References

- Alden, L. E., Taylor, C. T., Mellings, T. M. J. B., & Laposa, J. M. (2008). Social anxiety and the interpretation of positive social events. *Journal of Anxiety Disorders*, 22(4). https://doi.org/10.1016/j.janxdis.2007.05.007
- Beck, A. T., & Clark, D. A. (1997). An information processing model of anxiety: Automatic and strategic processes. *Behaviour Research and Therapy*, 35(1), 49–58. https://doi.org/10.1016/S0005-7967(96)00069-1
- Bell, V., Halligan, P. W., & Ellis, H. D. (2006). Explaining delusions: A cognitive perspective. *Trends in Cognitive Sciences*, 10(5), 219–226. https://doi.org/10.1016/j.tics.2006.03.004
- Bensi, L., & Giusberti, F. (2007). Trait anxiety and reasoning under uncertainty. *Personality and Individual Differences*, 43(4), 827–838. https://doi.org/10.1016/j.paid.2007.02.007
- Button, K. S., Kounali, D., Stapinski, L., Rapee, R. M., Lewis, G., & Munafò, M. R. (2015). Fear of negative evaluation biases social evaluation inference: Evidence from a probabilistic learning task. *PLOS ONE*, 10(4), e0119456. https://doi.org/10.1371/journal.pone.0119456
- Chen, J., Short, M., & Kemps, E. (2020). Interpretation bias in social anxiety: A systematic review and meta-analysis. *Journal of Affective Disorders*, 276, 1119–1130. https://doi.org/10.1016/j.jad.2020.07.121
- Dryman, M. T., & Heimberg, R. G. (2015). Examining the relationships among social anxiety, fears of evaluation, and interpretation bias. *Cognitive Therapy and Research*, 39(5), 646–657. https://doi.org/10.1007/S10608-015-9694-4/TABLES/3
- Dudley, R., Taylor, P., Wickham, S., & Hutton, P. (2016). Psychosis, delusions and the "Jumping to Conclusions" reasoning bias: A systematic review and meta-analysis. *Schizophrenia Bulletin*, 42(3), 652–665. https://doi.org/10.1093/schbul/sbv150
- Fredrick, J. W., & Luebbe, A. M. (2020). Fear of positive evaluation and social anxiety: A systematic review of trait-based findings. *Journal of Affective Disorders*, 265, 157–168. https://doi.org/10.1016/J.JAD.2020.01.042
- Garety, P. A., Hemlsey, D. R., & Wessely, S. (1991). Reasoning in Deluded Schizophrenic and Paranoid Patients. *The Journal of Nervous and Mental Disease*, 179(4), 194–201. https://doi.org/10.1097/00005053-199104000-00003
- Garety, P., Freeman, D., Jolley, S., Ross, K., Waller, H., & Dunn, G. (2011). Jumping to conclusions: the psychology of delusional reasoning. *Advances in Psychiatric Treatment*, 17(5), 332–339. https://doi.org/10.1192/apt.bp.109.007104
- Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4), 457–511. https://doi.org/10.1214/ss/1177011136
- Giusti, L., Ussorio, D., Salza, A., Malavolta, M., Aggio, A.,Bianchini, V., Casacchia, M., & Roncone, R. (2018).Preliminary study of effects on paranoia ideation and jumping to conclusions in the context of group treatment

of anxiety disorders in young people. *Early Intervention in Psychiatry*, 12(6), 1072–1080. https://doi.org/10.1111/eip.12415

- Heinrichs, N., & Hofmann, S. G. (2001). Information processing in social phobia: a critical review. *Clinical Psychology Review*, 21(5), 751–770. https://doi.org/10.1016/S0272-7358(00)00067-2
- Huq, S. F., Garety, P. A., & Hemsley, D. R. (1988). Probabilistic judgements in deluded and non-deluded subjects. *The Quarterly Journal of Experimental Psychology*, 40(4), 801–812. https://doi.org/10.1080/14640748808402300
- Koban, L., Schneider, R., Ashar, Y. K., Andrews-Hanna, J. R., Landy, L., Moscovitch, D. A., Wager, T. D., & Arch, J. J. (2017). Social anxiety is characterized by biased learning about performance and the self. *Emotion*, 17(8), 1144–1155. https://doi.org/10.1037/emo0000296
- Leary, M. R. (1983). A brief version of the fear of negative evaluation scale. *Personality and Social Psychology Bulletin*, 9(3), 371–375. https://doi.org/10.1177/0146167283093007
- Leppänen, J. M. (2006). Emotional information processing in mood disorders: a review of behavioral and neuroimaging findings. *Current Opinion in Psychiatry*, 19(1), 34–39.

https://doi.org/10.1097/01.yco.0000191500.46411.00

- Lincoln, T. M., Lange, J., Burau, J., Exner, C., & Moritz, S. (2010). The effect of state anxiety on paranoid ideation and jumping to conclusions. An experimental investigation. *Schizophrenia Bulletin*, 36(6), 1140–1148. https://doi.org/10.1093/schbul/sbp029
- Moritz, S., & Woodward, T. S. (2005). Jumping to conclusions in delusional and non-delusional schizophrenic patients. *British Journal of Clinical Psychology*, 44(2), 193–207. https://doi.org/10.1348/014466505X35678
- Müller-Pinzler, L., Czekalla, N., Mayer, A. V, Stolz, D. S., Gazzola, V., Keysers, C., Paulus, F. M., & Krach, S. (2019). Negativity-bias in forming beliefs about own abilities. *Scientific Reports*, 9(1), 14416. https://doi.org/10.1038/s41598-019-50821-w
- Peters, L., Sunderland, M., Andrews, G., Rapee, R. M., & Mattick, R. P. (2012). Development of a short form Social Interaction Anxiety (SIAS) and Social Phobia Scale (SPS) using nonparametric item response theory: The SIAS-6 and the SPS-6. *Psychological Assessment*, 24(1), 66–76. https://doi.org/10.1037/a0024544
- Rapee, R. M., & Heimberg, R. G. (1997). A cognitivebehavioral model of anxiety in social phobia. *Behaviour Research and Therapy*, 35(8), 741–756. https://doi.org/10.1016/S0005-7967(97)00022-3
- Ross, R. M., Mckay, R., Coltheart, M., & Langdon, R. (2015). Jumping to conclusions about the beads task? A meta-analysis of delusional ideation and data-gathering. *Schizophrenia Bulletin*, 41(5), 1183–1191. https://doi.org/10.1093/schbul/sbu187

- Schlier, B., Helbig-Lang, S., & Lincoln, T. M. (2016). Anxious but thoroughly informed? No jumping-toconclusions bias in social anxiety disorder. *Cognitive Therapy and Research*, 40(1), 46–56. https://doi.org/10.1007/s10608-015-9724-2
- So, S. H., Siu, N. Y. fan, Wong, H. lam, Chan, W., & Garety, P. A. (2016). 'Jumping to conclusions' datagathering bias in psychosis and other psychiatric disorders — Two meta-analyses of comparisons between patients and healthy individuals. *Clinical Psychology Review*, 46, 151–167. https://doi.org/10.1016/j.cpr.2016.05.001
- Speechley, W. J., Ngan, E. T.-C., Moritz, S., & Woodward, T. S. (2012). Impaired evidence integration and delusions in schizophrenia. *Journal of Experimental Psychopathology*, 3(4), 688–701. https://doi.org/10.5127/jep.018411
- Su, Y., & Yajima, M. (2021). R2jags: Using R to run "JAGS." R Journal.
- Tan, N., Shou, Y., Chen, J., & Christensen, B. (2022). *Jumping to Conclusion Bias through the Bayesian Lens.* Manuscript in preparation.
- Weeks, J. W., Heimberg, R. G., Fresco, D. M., Hart, T. A., Turk, C. L., Schneier, F. R., & Liebowitz, M. R. (2005). Empirical validation and psychometric evaluation of the Brief Fear of Negative Evaluation Scale in patients with social anxiety disorder. *Psychological Assessment*, 17(2), 179–190. https://doi.org/10.1037/1040-3590.17.2.179
- Weeks, J. W., Heimberg, R. G., & Rodebaugh, T. L. (2008). The Fear of Positive Evaluation Scale: Assessing a proposed cognitive component of social anxiety. *Journal* of Anxiety Disorders, 22(1), 44–55. https://doi.org/10.1016/j.janxdis.2007.08.002
- Weeks, J. W., & Howell, A. N. (2012). The bivalent fear of evaluation model of social anxiety: Further integrating findings on fears of positive and negative evaluation. *Cognitive Behaviour Therapy*, 41(2), 83–95. https://doi.org/10.1080/16506073.2012.661452
- Westermann, S., Salzmann, S., Fuchs, X., & Lincoln, T. M. (2012). Introducing a social beads task. *Journal of Experimental Psychopathology*, 3(4), 594–611. https://doi.org/10.5127/jep.017111

# **Predicting Algorithmic Complexity for Individuals**

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

How difficult is it to simulate an algorithm in one's mind and correctly deduce its outcome? In this paper, we present a predictive modeling task in the domain of algorithmic thinking in a railway environment. We present metrics, either based on algorithmic representation complexity (e.g. lines of code) or on the effect on cognitive resources an algorithm simulation can have (e.g. context switching). We implement the metrics within a benchmark and evaluate their predictive performance on an individual level, by assigning a complexity threshold to each individual. We compare these results to a standard statistical correlation analysis and suggest a different perspective for determining the predictive powers of complexity metrics as models.

**Keywords:** Algorithmic thinking; predictive modeling; problem solving; cognitive processes; deduction

#### Introduction

An algorithm is a set of well-defined instructions to be followed in order to solve a specific class of problems. Even though algorithms are most often associated with mathematical and computer sciences, they are present in each human's everyday life. Food recipes, furniture building instructions, getting coffee from a coffee machine... Each one of these examples has a set of rules associated with them that we follow in order to obtain the desired outcome.

Imagine you are building your new bookshelf, and the next step in the visual instructions depicts hammering all the nails in the package, totaling up to 40. What if the instructions were written? One way to express this step would be to write the instruction "Hammer a nail" 40 times, which is obviously unreasonable. Instead, there would be a condensed version represented as a *loop of operations* which would indicate that the hammering action needs to be repeated 40 times, or, while we still have nails in the package. Such loops are very often encountered in computer programming and we distinguish two sorts (Rogers, 1967): *for*-loops where the instructions are repeated *for* a certain amount of times and *while*-loops where the instructions are repeated *while* a specified condition holds.

The comprehension and formulation of algorithms has been researched by psychologists investigating computational thinking (Bucciarelli, Mackiewicz, Khemlani, & Johnson-Laird, 2022). Creating an algorithm for solving a problem requires solving representative instances of the problem class, simulating the process of solution to abduce an algorithm and simulating an algorithm to deduce its consequences in order to determine its correctness (Khemlani, Mackiewicz, Bucciarelli, & Johnson-Laird, 2013).

We narrow down the algorithm domain to a *railway* environment, following Khemlani et al. (2013). Given an ordered sequence of train wagons on a track, rearrangement algorithms of different complexities can be executed, leading to a new order of the wagons on a different track. Focusing on deducing an algorithm's output in this domain, we are interested in the difficulty of such tasks, operationalized by the correctness that humans achieve when trying to solve them. Khemlani et al. (2013) present their finding that for deduction, the difficulty does not depend on the number of moves performed while executing an algorithm, but rather on the Kolmogorov complexity of a corresponding Lisp function containing while-loops for rearranging trains of any length. In this paper, our interest lies in using complexity metrics to model the difficulty an individual has when deducing the correct wagon order after applying a rearranging algorithm.

Though often used as synonyms, the terms 'complexity' and 'difficulty' are in fact used to differentiate between (1) intrinsic characteristics which influence performance but are independent of the context and the people solving the problem, and (2) the direct relationship to the observed performance and subjective individual experience (Effenberger, Cechák, & Pelánek, 2019). There exists a relationship between complexity and difficulty, which, as shown in various settings, is reflected in a human's behavior and performance (Sheard et al., 2013; Campbell, 1988; Liu & Li, 2012).

The complexity of an algorithmic task can be defined using different metrics, like the previously mentioned Kolmogorov complexity (Khemlani et al., 2013), or simply the length of lines in a code describing the algorithm (Nguyen, Deeds-Rubin, Tan, & Boehm, 2007). To further explore the relation between complexity and difficulty, we introduce several metrics, where some are based on algorithmic complexities, while others take the cognitive perspective into consideration, specifically the effect on cognitive resources that the simulation of an algorithm can have.

In order to assess whether our metrics reflect the relationship between complexity and difficulty, we conducted an experiment to establish a data foundation. The experiment was based on the tasks in Khemlani et al.'s (2013) study, where participants were asked to deduce the outcome of applying different rearrangement algorithms to a set of ordered Proceedings of the 20th International Conference on Cognitive Modelling (ICCM 2022)



Figure 1: The Top Left, Top Right and Bottom Right train tracks.

wagons. Following Goodwin and Johnson-Laird (2011), we first evaluated the metrics based on the correlation between their values and the correctness achieved by participants. In a second step, we defined a modeling task where the objective was to predict the correctness of the participants' answers. By implementing our metrics as models within this task, we evaluated their predictive performance and investigated whether the results found based on correlations translate to an individual predictive level.

# Experiment

The goal of the experiment was to test the ability of individuals to correctly deduce the consequence of applying an algorithm. Given a sequence of wagons in a particular order and a rearrangement algorithm, the participants' goal was to simulate the algorithm's execution in their mind and deduce the final order of the wagons, which they provided as an answer.

Our task and algorithm design was inspired by Khemlani et al. (2013). The task scenario consisted of three train tracks (*Top Left, Top Right* and *Bottom Right*), as shown in Figure 1. Initially, the wagons were placed in the *Top Left* track, and the goal was to have the wagons rearranged in the *Top Right* track. The *Bottom Right* track could be used by the algorithms as an intermediate track.

The four algorithms that we used from the railway domain are *Reverse*, *Palindrome*, *Parity Sort* and *Faro Shuffle*. Table 2 shows the initial states of the wagons and their final order after applying the algorithms to four and six wagons. We visualized the algorithms shown in Figure 2 using code blocks (see Figure 3). We distinguish between three types of code blocks - While, Repeat and Move. The While blocks indicate that the commands in their scope will be repeatedly executed as long as a condition holds and they are equivalent to *while*-loops. The Repeat blocks work similarly to While, except the number of repeated executions is explicitly determined with an integer, equivalent to *for*-loops. The Move blocks describe the wagon moving operation that should be performed. A wagon can be moved only between *Top Left* and *Bottom Right*, and *Top Left* and *Top Right*.

The experiment manipulated the type of the rearrangement algorithm and the number of wagons that the algorithm needs to be applied to (four or six). This allows for examining the difficulty of a deduction not only based on the algorithm itself, but also by taking into consideration how an algorithm's complexity changes when the number of instances it needs to be applied to differs. Finally, every participant was presented with eight tasks - four algorithms applied once to four wagons, and once to six.

Reverse
While track Top-Left has wagons: Move wagon from Top-Left to Bottom-Right While track Bottom-Right has wagons: Move wagon from Bottom-Right to Top-Left Move wagon from Top-Left to Top-Right Rolindromo
While first wagon on Top-Left has color blue: Move wagon from Top-Left to Bottom-Right While track Bottom-Right has wagons: Move wagon from Bottom-Right to Top-Left Repeat 2 times: Move wagon from Top-Left to Top-Right
Parity Sort
While track Top-Left has wagons: Move wagon from Top-Left to Top-Right Move wagon from Top-Left to Bottom-Right While track Bottom-Right has wagons: Move wagon from Bottom-Right to Top-Left While track Top-Left has wagons: Move wagon from Top-Left to Top-Right
Faro Shuffle
While not solved: Move wagon from Top-Left to Top-Right While first wagon on Top-Left has color orange:

Move wagon from Top-Left to Top-Right While first wagon on Top-Left has color orange: Move wagon from Top-Left to Bottom-Right Move wagon from Top-Left to Top-Right While track Bottom-Right has wagons: Move wagon from Bottom-Right to Top-Left

Figure 2: Wagon rearrangement algorithms as presented in the experiment. The algorithms were visualized using code blocks, as shown in Figure 3.

Re	epeat while track <b>Top-Left</b> has waggons:
	Move waggon from Top-Left to Bottom-Right
Re	epeat while track Bottom-Right has waggons:
	Move waggon from Bottom-Right to Top-Left
	Move waggon from Top-Left to Top-Right

Figure 3: Example code blocks for the *Reverse* algorithm.

# **Participants**

Thirty-six participants completed the experiment (age 18-35, 72% female). They were recruited on Prolific<sup>1</sup> and the experiment was performed online as a web-experiment. After completing the experiment, participants received a compensation of 10 EUR. 21 of them indicated to have 'some' programming background, 14 had 'none' and 1 participant had 'profound' background. All of them were native English speakers.

# Procedure

Participants were first given an introductory task, which explained the visualization of the train tracks and the code blocks that describe the algorithm for rearranging the wagons. They were presented with the following two rules regarding the execution of the Move block: 1. Only one wagon will be moved at a time; 2. Only the wagon closest to the crossing on a track will be moved (it is referred to as the first wagon of the track). They were informed that their goal is determining the order of the wagons on the Top Right track resulting from executing the algorithm described by the code blocks. They needed to write their answer in a text-field above the Top Right track. Participants were instructed not to use external tools (like pen and paper) to solve the task, but were encouraged to simulate the algorithm in their mind. Afterwards, the participants received their first task. They were presented with either four or six wagons on the Top Left track and code blocks for a rearrangement algorithm. Once they entered their answer, they could proceed to the next task.

# **Observed Data**

The total number of tasks in the experimental data is 288 (36 participants, 8 tasks each). We eliminated 12 tasks, because while solving the task, participants left the page for more than half a minute in total, leaving us with 276 valid data points (*Reverse:* 71, *Palindrome:* 69, *Parity Sort:* 69, *Faro Shuffle:* 67). For 66 of them (23.9%) participants provided correct answers (*Reverse:* 24, *Palindrome:* 11, *Parity Sort:* 18, *Faro Shuffle:* 13). The exact number of correct answers for each one of the eight tasks is provided in Table 1.

# **Modeling Difficulty**

Based on the data obtained from our experiment, we introduce a modeling task for algorithmic thinking within the railway environment. The objective of the modeling task is

Table 1: Total number of valid data points and correct answers for each task.

Algorithm	#Wagons	Total Valid	Correct
Daviana	4	36	10 (27.8%)
Keverse	6	35	14 (40.0%)
Palindrome	4	35	4 (11.4%)
	6	34	7 (20.6%)
Parity Sort	4	35	9 (25.7%)
	6	34	9 (26.5&)
Eano Shuffle	4	34	11 (32.4%)
rato snujjie	6	33	2 (6.0%)

to determine the difficulty, operationalized by the number of errors participants make deducing an algorithm's outcome based on the algorithm and the initial arrangement.

# **Complexity Metrics**

In the following we introduce seven metrics which can be roughly divided into two conceptual groups. The first group of metrics are based on the complexity of the algorithm's structure only, which is common for assessing the complexity of program code. They are suited to represent an individual's ability to understand the algorithm and the underlying concept of what the algorithm is supposed to do and estimate the complexity of the execution based on the complexity of the algorithm itself. Metrics of the second group consider the precise steps performed by the algorithm when executed on a specific output. Therefore, these metrics can better account for the cognitive load that occurs while an individual simulates the steps of the algorithm in their mind, but also requires them to execute the algorithm in order to measure the respective complexity estimate.

*Depth* is based on algorithmic computational complexity which increases when nesting loops. Assuming that when the innermost statements are deeper nested in loops, simulating the algorithm's execution should be more difficult for an individual, the metric describes the depth of an algorithm with respect to (nested) loops, while starting with a top-level depth value of 1. *Reverse* has a depth value of 3, obtained by adding 1 (top-level) + 1 (first level - While commands), + 1 (second level - instructions within While commands). This metric provides the same value for an algorithm, independent of the number of wagons it would be applied to.

*Structure* mimics the relation between the length of a code describing an algorithm and a perceived level of task difficulty (Sheard et al., 2013; Nguyen et al., 2007) by counting the number of code blocks in the algorithm. *Reverse* has a structure value of 5, as it has five blocks. Similarly to the depth metric, structure also provides the same value for an algorithm, for any number of wagons.

*Moves* is the number of wagon moves that the algorithm performs until completion, following Khemlani et al. (2013). When applying *Reverse* to four wagons, 12 moves are

<sup>&</sup>lt;sup>1</sup>https://www.prolific.co/

Algorithm	Initial	Caal	Complexity Metrics						
Algorium	mitiai	Goal	Depth	Structure	Moves	Commands	Contexts	Signature	Entropy
Payarsa	1234	4321	3	5	12	22	8	5.625	0.822
Keverse	123456	654321	5	5	18	32	12	5.906	0.833
	1234	1423			8	18	4	5.25	0.929
Palindrome	123456	162534	4 6	6	12	26	6	6.125	1.022
	1234	1324			0	17	5	6	1.015
Parity Sort	1234	135246	3	7	8 12	24	5 7	0 7	1.013
	120100	100210			12	21	,	,	1.071
Faro Shuffle	1234	1324	4	7	6	15	5	6	0.937
Furo Shujjie	123456	142536	4	1	12	28	9	11	1.050

Table 2: Initial and goal states, and complexity metric values for each one of the eight railway domain tasks.

performed: 4 (first Move instruction on four wagons) +  $2 \times 4$  (two Move instructions on four wagons).

*Commands* takes the structure metric a step further and counts the amount of times that code blocks have been executed, thereby acknowledging possible costs for checking a loop's condition. In the four wagon scenario, *Reverse* has a commands value of 22: 8 (first While and its Move blocks are executed once for each wagon) + 12 (second While and two Move blocks executed for each wagon) + 2 (execution of two While blocks when their condition does not hold).

Contexts represents the people's limitation to attending to only one context in their working memory and the cognitive load increase when context switching is necessary (Garavan, 1998). This metric defines a context as operating on a pair of tracks, i.e. moving wagons from one track to another. When switching between different Move instructions, the relevant pair of tracks changes which leads to a context switch. The metric counts the number of context switch occurrences during an algorithm execution, where a higher number indicates higher cognitive load and therefore a task is deemed more difficult. In the case of Reverse with four wagons, 8 context switches happen: 1 (operating on Top Left and Bottom Right in the first While block and switching to Bottom Right and Top Left in the second While block) + 7 (constant alternating between Bottom Right and Top Left and Top Left and Top *Right* when executing the second While block).

Signature imitates the repetition effect (Bertelson, 1961), which shows that an individual needs less time to perform a repeated task. We transform the effect to a complexity metric in this domain, by assuming that once an individual has processed a command once, its repetitions within a loop should be perceived as easier. The metric assigns a cost of 1 to each executed command in an algorithm, while checking if a command is immediately repeated (within a loop), in which case the cost is halved in each repetition. The signature cost of *Reverse* on four wagons is 5.625: 1.875 (first Move command repeated four times: 1 + 0.5 + 0.25 + 0.125 + 3.75 (other two Move commands repeated four times).

*Entropy* is a measurement of potential knowledge and randomness in information theory (Shannon, 1948). Used as a metric to quantify uncertainty, we apply it in this scenario by taking into consideration the distribution of the wagons over all three tracks. After each move, the entropy of the wagons on the tracks is calculated, as shown in Eq. 1. A higher entropy value indicates a more chaotic distribution of the wagons, potentially increasing the difficulty for individuals to simulate the algorithm and deduce its correct outcome. The final value is the average of all calculated entropies. *Reverse*'s entropy when applied to four wagons is 0.822.

$$E = -\sum_{k} (p_k \cdot \log_2 p_k) \tag{1}$$

The complexity metrics' values for each task are presented in Table 2.

Following approaches in related work (Goodwin & Johnson-Laird, 2011; Khemlani et al., 2013; Khemlani, Goodwin, & Johnson-Laird, 2015), we measure the correlation between the complexity metrics' values and the correctness of the individuals' answers, results shown in Table 3.

Complexity Metric	ρ	p-value
Depth	-0.145	.016
Structure	-0.008	.149
Moves	0.078	.202
Commands	0.019	.751
Contexts	0.086	.152
Signature	-0.133	.027
Entropy	-0.123	.041

Table 3: Correlation (Pearson's  $\rho$ ) between complexity metrics and answer correctness. Significant p-values are marked in bold.

#### **Modeling Individuals**

The significant correlation values (Table 3) indicate that depth, signature and entropy should be the best predictors of answer correctness. We want to determine whether this holds on the individual level - are the metrics with statistically significant correlation good predictors of task difficulty for each individual in our data set?

In our modeling approach models are evaluated based on their ability to account for an individual's capability to correctly deduce the final order of the wagons after applying a rearrangement algorithm. To perform our evaluation, we relied on the Cognitive Computation for Behavioral Reasoning Analysis (CCOBRA) framework<sup>2</sup>, which facilitates model evaluations with a focus on modeling reasoning behavior on the individual level (Riesterer, Brand, & Ragni, 2020). Similar to Riesterer et al. (2020), we performed a coverage analysis, which allows models to fit to each individual participant in the data. This approach allows to assess a models ability to represent a participant's behavior within its parameter space. In our case, the models were created by equipping each metric with a complexity threshold that represented the maximum complexity that an individual participant could "handle", i.e., the complexity value up until which the participant is able to give the correct answer. All models then fitted their thresholds to each individual. When a model is then queried for a prediction for a given task, it determines its prediction by comparing the individual's threshold to the task complexity according to the respective metric: If the complexity is too high, it is predicted that the individual will not solve this task correctly.

Besides models for each metrics, we implemented an additional baseline model which always predicts the participant to give an incorrect answer. This serves as a reasonable lower-bound, as the average correctness for the tasks was below 50%.

#### Results

Each model was judged on its ability to account for an individual's difficulty threshold, on which it depends whether a correct answer is given or not. Table 4 shows the accuracy values for each complexity metric and Figure 4 shows how good the individual participants are predicted. All of them achieve an accuracy above 80% and perform better than the baseline model.

The best performance is achieved by entropy, closely followed by structure, with an accuracy value of 87%. Interestingly, entropy showed a significant correlation to the answer correctness, yet structure did not. The signature metric, with a significant correlation performed very well in our benchmark, reaching an accuracy of 86%. However, even though its mean performance is rather high, Figure 4 shows that it does not manage to fully cover as many individuals as the other metrics. The discrepancy between significant correlations and predictive powers is shown by the depth metric - the worst predictor out of all seven metrics, even though it has a significant correlation. A part of the problem might be the dichotomous nature of the metric, which only assigns the values 3 or 4 to the present algorithms. This restricts the expressiveness of the threshold, impeding its ability to discern between individuals. On the other hand, the structure metric is still the second best predictor without a significant correlation value, although it only distinguishes between 3 possible values for our tasks.

Complexity Metric Model	Accuracy
Entropy	87%
Structure	87%
Signature	86%
Contexts	83%
Commands	83%
Moves	83%
Depth	81%
Baseline	76%

Table 4: Benchmark evaluation results - accuracy values of the complexity metrics as predictive models for a task's difficulty, ordered from best to worst.



Figure 4: Benchmark evaluation results - individual accuracy values of the complexity metrics as models for a task's difficulty. Triangles denote the mean performance.

# **Discussion and Conclusion**

In this paper we presented seven different complexity metrics which we used in our proposed difficulty modeling task in the algorithmic thinking domain with a focus on a railway environment. We implemented them in a benchmark and evaluated their predictive performance by comparing them against a baseline model and also with their correlation values. We used data from an experiment we conducted whose design is inspired by a previous study in the railway environment by Khemlani et al. (2013).

The best performance achieved by entropy is not a surprising result, as it is expected that a more disorganized

<sup>&</sup>lt;sup>2</sup>https://github.com/CognitiveComputationLab/ccobra

distribution of the wagons on the tracks should lead to more difficulties remembering wagons' positions and adjusting them after performing rearrangement operations. Through the signature metric we also learn that in some cases the immediate repetition of instructions helps individuals when simulating an algorithm.

The structure metric is the second best predictor in our benchmark, even though it doesn't have a significant correlation and it only provides 3 possible values. The combination of a high predictive performance with a relatively low degree of freedom for the threshold indicates that its underlying concept is in fact meaningful. That is in line with the found relevance of lines of code to task difficulty (Sheard et al., 2013; Nguyen et al., 2007).

Analyzing the relation between complexity metrics and perceived difficulty of algorithmic tasks is a topic researched for many years from different perspectives, e.g. understanding mental processes in computational thinking (e.g. Khemlani et al. (2013), education and exam creation (e.g. Sheard et al. (2013)) and software maintenance (e.g. Curtis, Sheppar, Milliman, Borst, and Love (1979)). In such studies, usually a significant correlation is always taken as a sign of a good predictor, but we show that our modeling task gives us the possibility to analyze the complexity metrics' capability to act as predictive models of task difficulty beyond statistical analysis. For example, while the depth metric was considered to be a good predictor based on correlation, it failed to translate to an adequate performance when predicting the complexity for individuals and was outperformed by all other metrics.

Our findings open many doors and possibilities for future, exciting research. In the experiment we found a difference in correctness patterns between 4 and 6 trains. An interesting next step would be to perform further analysis whether this influences the complexity metrics as predictive models for these tasks as well. Additionally, it would be useful to conduct a similar experiment, where individuals are exposed to the same tasks multiple times. That would allow for broadening the modeling task to predictions on new, unseen individual data. Further research steps can be taken by looking deeper into relations between the metrics and examining the predictive powers of their combinations. For example, the contexts metric on its own did not perform as well as the best performing models, but its performance might be bettered by analyzing how switching contexts taken into consideration together with the distribution of wagons (entropy) predicts perceived difficulty. Moreover, it would be of great interest to research whether expanding already existing algorithmspecific complexity metrics towards considering the cognitive load an algorithm can have on an individual would lead to better predictions of difficulty thresholds. Finally, the setting allows for an extended version of the modeling task: Instead of predicting the expected correctness based on complexity, the prediction of the exact responses given by participants to a task can serve as a challenging objective. The extension of the modeling task requires a extensive data set, but would in turn open the task for models that go beyond an estimate of complexity. Instead, models that are able account for and simulate the processes underlying human algorithmic thinking would be required to solve the task.

### Acknowledgements

This research was funded by the Saxony State Ministry of Science and Art (SMWK3-7304/35/3-2021/4819) research initiative "Instant Teaming between Humans and Production Systems" on the basis of the budget passed by the deputies of the Saxony state parliament.

# References

- Bertelson, P. (1961). Sequential redundancy and speed in a serial two-choice responding task. *Quarterly Journal of Experimental Psychology*, *13*(2), 90-102.
- Bucciarelli, M., Mackiewicz, R., Khemlani, S., & Johnson-Laird, P. N. (2022). The causes of difficulty in children's creation of informal programs. *International Journal of Child-Computer Interaction*, 31, 100443-100455.
- Campbell, D. J. (1988). Task complexity: A review and analysis. *Academy of management review*, *13*(1), 40-52.
- Curtis, B., Sheppar, S. B., Milliman, P., Borst, M. A., & Love, T. (1979). Measuring the psychological complexity of software maintenance tasks with the halstead and mccabe metrics. *IEEE Transactions on Software Engineering*, 2, 96-104.
- Effenberger, T., Cechák, J., & Pelánek, R. (2019). Difficulty and complexity of introductory programming problems..
- Garavan, H. (1998). Serial attention within working memory. *Memory & Cognition*, 26(2), 263-276.
- Goodwin, G. P., & Johnson-Laird, P. N. (2011). Mental models of boolean concepts. *Cognitive Psychology*, 63(1), 34-59.
- Khemlani, S., Goodwin, G. P., & Johnson-Laird, P. N. (2015). Causal relations from kinematic simulations. In Proceedings of the 37th annual conference of the cognitive science society (p. 1075-1080).
- Khemlani, S., Mackiewicz, R., Bucciarelli, M., & Johnson-Laird, P. N. (2013). Kinematic mental simulations in abduction and deduction. In *Proceedings of the national* academy of sciences (Vol. 110(42), p. 16766-16771).
- Liu, P., & Li, Z. (2012). Task complexity: A review and conceptualization framework. *International Journal of Industrial Ergonomics*, 42(6), 553-568.
- Nguyen, V., Deeds-Rubin, S., Tan, T., & Boehm, B. (2007). A sloc counting standard. *Cocomo II forum*, 1-16.
- Riesterer, N., Brand, D., & Ragni, M. (2020). Do models capture individuals? evaluating parameterized models for syllogistic reasoning. In *Proceedings of the 42nd annual conference of the cognitive science society* (p. 3377-3383).
- Rogers, H. (1967). *Theory of recursive functions and effective computability*.
- Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3), 379-423.
Sheard, J., Simon, B., Carbone, A., Chinn, D., Clear, T., Corney, M., ... Teague, D. (2013). How difficult are exams? a framework for assessing the complexity of introductory programming exams. In *Proceedings of the fifteenth australasian computing education conference* (p. 145-154).

# Learning Linguistic Reference Biases in the PRIMs Cognitive Architecture

# Abigail Toth (a.g.toth@rug.nl)

Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen Nijenborgh 9, 9747 AG Groningen, The Netherlands

## Petra Hendriks (p.hendriks@rug.nl)

Center for Language and Cognition Groningen, University of Groningen Jatstraat 26, 9712 EK Groningen, The Netherlands Nijenborgh 9, 9747 AG Groningen, The Netherlands

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

Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen Nijenborgh 9, 9747 AG Groningen, The Netherlands

#### Jacolien van Rij (j.c.van.rij@rug.nl)

Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen Nijenborgh 9, 9747 AG Groningen, The Netherlands

**Keywords:** cognitive modelling; PRIMs cognitive architecture; language learning; implicit causality

# Background

In order to keep up with the rapid speed of spoken language ( $\sim 2$  words per second in conversational English), language users rely on both linguistic and non-linguistic biases in order to anticipate linguistic input before actually encountering it. One of these biases is known as the *implicit causality bias*, which is illustrated using the examples in 1 below.

- (1) a. Samuel apologized to Noah because...
  - b. Samuel congratulated Noah because...

There is evidence that when language users encounter sentences like these, they expect 1 a to continue about Samuel, the preceding grammatical subject and 1b to continue about Noah, the preceding grammatical object (e.g., Koornneef & Van Berkum, 2006; Pyykkönen & Järvikivi, 2010). This seems to be driven by the assumption that Samuel's behavior more likely caused the apologizing, whereas Noah's behavior more likely caused the congratulating event. As such, 'apologize' is considered a subject-biased implicit causality verb and 'congratulate' is considered an object biased-implicit causality verb.

Despite the important role of biases for predictive language processing, we know very little about how they are acquired and how exactly they get used in real-time. In the present study we report on an updated version of our reference learning model (Toth, Hendriks, Taatgen, & Van Rij, 2021), which was developed in order to investigate whether domain-general mechanisms could explain how language users learn reference biases and to explore how these biases may get used during real-time language processing.

#### Present study: methods and results

We constructed a cognitive model in the PRIMs cognitive architecture (Taatgen, 2013, 2014), which processed

sentences like those in 1. The model then predicted whether the next referent would be the subject referent (e.g., Samuel) or the object referent (e.g., Noah). Subsequently, the model predicted whether the referent would be in the form of a proper name (e.g., 'Samuel'/'Noah') or a pronoun (in both cases, 'he'). The model was then presented the actual continued discourse. In cases where the model's predictions matched the continued discourse the model was issued a reward. Across the 10,000 input items the model was presented with, there were asymmetries with respect to how discourse continued. For example, after subject-biased implicit causality verbs the discourse was more likely to continue about the subject referent, whereas after object-biased implicit causality verbs the discourse was more likely to continue about the object referent. Furthermore, continued subject referents were more likely to take the form of a pronoun, whereas continued object referents were more likely to take the form of a proper name.

We utilized PRIMs' context-operator learning, based on reinforcement learning, such that whenever the model was issued a reward, the associative strengths between the current context and all of the operators (similar to ACT-R production rules) that fired up until that point were increased. This made it more likely for the model to retrieve the same operators in similar contexts in the future.

Crucially, in its initial state, before our reference model processed a certain amount of input items (and updated the associative strengths), it was equally as likely to retrieve subject referent and object referent predicting operators, and likewise name and pronoun predicting operators across the different item types. However, by utilizing context-operator learning the model was able to optimize its predictions, resulting in biased behavior that was in line with the asymmetrical input. The main findings are illustrated in the figures below.



Figure 1: Grand average subject predictions for each implicit causality verb type (green: subject-biased, yellow: neutral and blue: object-biased).

#### Next referent form predictions



B Predicted object referent continuation



Figure 2: Grand average pronoun predictions for each implicit causality verb type (green: subject-biased, yellow: neutral and blue: object-biased) and predicted next referent (A: subject predictions and B: object predictions).

As can be seen in Figure 1, during the initial items the model predicted that the next referent would be the subject referent at chance level for each verb type. However, as the model was presented with an increasing amount of input, the proportion of predicting that the next referent would be the subject referent uniquely changed for each verb type. These results illustrate that the model picked up on the next referent asymmetries in the input, resulting in a learnt implicit causality bias.

As can be seen in Figure 2, in cases where the model predicted the next referent to be the subject, the proportion of pronoun predictions steadily increased for all three verb types, reaching ceiling after  $\sim 1500$  items. In cases where the model predicted the next referent to be the object, pronoun predictions gradually decreased for all three verb types, but with each showing a unique pattern. These results illustrate that the model picked up on the next referent form asymmetries in the input, resulting in a pronoun bias for subject referents and a name bias for object referents, which in the case of object referents seems to interact with verb type.

In order to further evaluate the learning of the model, after the model had already processed the 10,000 input items, we presented it with a series of items that were in some way novel (i.e., either a novel transitive verb or novel subject and object referents). By doing so we were able to conclude that the biases the model learned, generalized to new contexts.

## Conclusions

The present study highlights the advantages of using domain-general cognitive modelling to explain seemingly complex linguistic behavior. Using this method we were able to generate novel predictions that can be tested by future psycholinguistic experiments. The findings have implications for psycholinguistic theories of prediction in language, language learning and reference processing.

#### References

- Koornneef, A. W., & Van Berkum, J. J. A. (2006). On the use of verb-based implicit causality in sentence comprehension: Evidence from self-paced reading and eye tracking. *Journal of Memory and Language*, 54(4), 445-465.
- Pyykkönen, P., & Järvikivi, J. (2010). Activation and persistence of implicit causality information in spoken language comprehension. *Experimental Psychology*.
- Taatgen, N. A. (2013). The nature and transfer of cognitive skill. *Psychological Review*, *120*(3), 439-471.
- Taatgen, N. A. (2014). Between architecture and model: Strategies for cognitive control. *Biologically Inspired Cognitive Architectures*, 8, 132-139.
- Toth, A., Hendriks, P., Taatgen, N. A., & Van Rij, J. (2021). Learning reference biases from language input: A cognitive modelling approach. In *Proceedings of the 19th international conference on cognitive modeling*.

# Using Deep Neural Networks for Modeling Representational Spaces: The Prevalence and Impact of Rarely-Firing Nodes

## Nhut Truong (leminhnhut.truong@unitn.it)

Center for Mind/Brain Sciences (CIMeC), University of Trento

#### Uri Hasson (uri.hasson@unitn.it)

Center for Mind/Brain Sciences (CIMeC), University of Trento

**Keywords:** human similarity judgment, deep neural networks, representational similarity analysis, pruning

## Background

Deep neural networks (DNNs) are increasingly being used as computational models of human vision and higher-level cognition. Many studies have shown that after training these networks to categorize objects, the latent representations they form, quantified via image-similarity in multidimensional space, moderately approximate those produced by human similarity judgments. For example, Peterson, Abbott, and Griffiths (2018) showed that it is possible to improve the prediction of human similarity spaces from DNN embeddings by learning a reweighting of the saliency of each feature (or node). This suggests that DNNs learn relevant features for modeling human knowledge, but those features have the wrong level of saliency, which can be adjusted via reweighting.

Recently, Tarigopula, Fairhall, and Hasson (2021) have shown that it is possible to improve prediction of human similarity judgments not via reweighting, but via supervised pruning of DNN models. Pruning outperforms reweighting in learning human similarity spaces. Because pruning does not alter the original activations of retained features, its success suggests that DNNs may learn a relevant basis function at adequate levels of salience, but that only a subset of features is relevant when modeling human representational space.

## **Current study**

While the work of Tarigopula et al. (2021) used *supervised* pruning, in this work we examine to what extent we can achieve the same goal with an *unsupervised* method. Our work was inspired by a study by Hu, Peng, Tai, and Tang (2016). In their work, they show that in a trained DNN a substantial subset of nodes do not activate for the majority of stimuli, with some nodes not firing for over 90% of all images; moreover, removing such nodes has little impact on the network classification accuracy. In our work, we investigated how the removal of infrequently-activated nodes impacts the representational space of DNNs, and how useful they are for modeling human similarity spaces. For each node we computed the percentage of images in the dataset for which the node's activation was 0. We call this node-wise measure the Percentage of Zeros (PoZ) as in Hu et al. (2016).

In Experiment 1 we trained LeNet5, a small DNN, to classify the CIFAR-10 dataset. The dataset consists of 60000 small images drawn from 10 object categories. We then extracted the representations for 10000 test images from the penultimate layer (containing 84 nodes) to obtain a matrix size of  $10000 \times 84$ . From each matrix we computed a Baseline Representational Similarity Matrix (baseline RSM) from the average representations of each category, and the PoZ of each feature sorted from highest to lowest. We then iteratively removed 10%, 20%, ..., 90% of features according to their PoZ ranking, each time 1) recomputing an RSM from the pruned network, and computing the match between the pruned RSM and baseline RSM (quantified by Pearson correlation  $R^2$  fit between the two RSMs, a.k.a representational similarity analysis); and 2) storing the maximum PoZ value in the remaining features. We repeated the entire process 50 times to start from different initialization positions to obtain means and standard deviations for the two measurements.

As Figure 1 (blue line) shows, keeping the bottom 80% of *PoZ*-rank features had almost no impact on  $R^2$ , with values remaining very close to 1. A sharp drop only occurs once 30% of features and less are retained. It can also be seen (yellow line) that some features have *PoZ* values nearing 100%, and that, e.g., when ranked by *PoZ*, the top-ranked 20% features all had *PoZ* > 60%. The findings show that even for a relatively heterogeneous dataset, there is a substantial subset of features with mostly-zero firing, which contributes minimally to model the similarity space.

In **Experiment 2** we applied PoZ-based pruning to a more realistic dataset, but here we examined whether non-supervised PoZ-based pruning can improve the match between RSMs produced from a DNN and RSMs produced from human similarity judgments. The dataset included images from six different categories (Animals, Automobiles, Fruits, Furniture, Vegetables and Various), each consisting of 120 images. Human similarity matrices were obtained for all image-pairs within each category and provided to us by Peterson et al. (2018). For each set of 120 images we obtained DNN embeddings from the penultimate layer of the Pytorch ImageNet-pretrained VGG-16, computed and ranked the PoZ of each node, and then iteratively removed nodes based on PoZ ranking. After each removal we quantified the fit between the human RSM and the RSM for the DNN



Figure 1: **Experiment 1**. Impact of PoZ-based pruning on representational space. Blue line: match between baseline RSM and RSM of each pruning level. Yellow line: maximal PoZ value remaining in the set at each level of pruning.

pruned at that level.

Figure 2 presents the *PoZ* distribution per category. It shows that for all categories, more than 50% of features had PoZ > 80%. Consistent with this observation, Figure 3 shows that for all categories, the large majority of nodes could be removed with very little impact on the fit between the human and DNN RSMs. A substantial drop only occurred when less than 12% of the features were retained. We also found that for three categories, at least one pruned RSM provided a better fit to human judgements than an RSM computed from the non-pruned network. Significance testing showed that this pattern departed from chance for the Furniture and Vegetables category, where the fit between the DNN and Human RSM improved linearly till 24% and 41% respectively of the features remained.



Figure 2: **Experiment 2**. Cumulative histograms of PoZ values computed for embeddings of each image category.



Figure 3: **Experiment 2**. Fit between RSMs from DNN embeddings and human RSM, for different levels of pruning. Red dots mark point of maximal fit per category. Small vertical lines mark quantiles of PoZ values per category.

Discussion: Experiment 1 showed it was possible to remove all nodes with PoZ > 50% with minimal impact on representational space. This was unexpected: removal of nodes with very high PoZ values should obviously not impact representational space, but the reason for why removal of lower PoZ nodes had a similarly-weak impact requires further study. Experiment 2 generalized the results to a more extensive, realistic dataset and suggested that PoZ-based pruning of DNN embeddings can in some cases improve the fit with human similarity judgments. Overall, our findings suggest that high-PoZ nodes are weakly-informative, and prevalent in image sets of natural categories. We suggest these nodes should be considered as a separate class when constructing encoding or decoding models of human cognition.

#### Reproducibility

The code to produce results and figures is available at github.com/tlmnhut/DNN\_model\_sim\_space

#### References

- Hu, H., Peng, R., Tai, Y.-W., & Tang, C.-K. (2016, Jul). Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv*:1607.03250 [cs].
- Peterson, J. C., Abbott, J. T., & Griffiths, T. L. (2018, Nov). Evaluating (and improving) the correspondence between deep neural networks and human representations. *Cognitive Science*, 42(8), 2648–2669. doi: 10.1111/cogs.12670
- Tarigopula, H. P., Fairhall, S. L., & Hasson, U. (2021). Improved prediction of behavioral and neural similarity spaces using pruned dnns. *bioRxiv*.

# Modelling Expert and Novice Programming Strategies using Python ACT-R

Maria Vorobeva, Robert West, Kasia Muldner

Department of Cognitive Science Carleton University, Ottawa, Canada

#### Abstract

Cognitive architectures have been used to model human problem-solving strategies and behaviours in complex domains – here, we focus on programming. However, to date, models of programming have not included various strategies for generating programs. To address this, the present paper describes two cognitive models that simulate a novice and expert strategy for solving a programming problem in Python. The models were based on theoretical frameworks of expert and novice programming. The SGOMS framework was best for modeling experts and competent novices, because it provided functionality to represent goals and plans that mirrored ones used by these individuals.

**Keywords:** Python ACT-R, cognitive modelling, programming, expertise

# Introduction

Programming is a complex skill that requires time and practice to master. To date, however, the components of this skill and corresponding cognitive mechanisms are not clear. As we describe below, one way to fill this gap involves the construction of a cognitive model.

A cognitive model is a formalization of cognitive mechanisms that are hypothesized to impact problem solving and performance within a particular domain. A common cognitive architecture for building cognitive models is ACT-R (Anderson & Lebiere, 1998). ACT-R uses productions (if/then rules) to model problem solving in a given domain, and declarative memory to store facts about the domain. This architecture has inspired other similar architectures, such as Python ACT-R (Stewart & West, 2007), which is used in the present work. Implementing a cognitive model has a number of benefits. It requires the human author (i.e., the model builder) to formally specify the declarative and procedural knowledge needed to solve problems in a given domain. This formalization step is beneficial as it clarifies the cognitive mechanisms (Frischkorn & Schubert, 2018). Moreover, the model provides an environment for testing theories about the hypothesized cognitive mechanisms.

There is substantial work in the ACT-R community and beyond involving cognitive models for a range of tasks. In this review, we focus on problem-solving tasks in science domains. Two common domains used to implement cognitive models include physics and math (Braithwaite et al., 2017; VanLehn et al., 1991). To illustrate, the model Cascade formalized the mechanisms for self-explanation and analogical transfer used during physics problem solving (VanLehn et al., 1991). Another example is FARRA (Braithwaite et al., 2017), which is a model of fraction problem solving. FARRA simulations demonstrated that the distribution of problems in mathematics textbooks may inadvertently strengthen student misconceptions. Relevant to the present work, some researchers have formalized knowledge representations for program generation (Johnson & Soloway, 1985; Pirolli, 1986; Corbett, 2000). The focus of this work was to parse and/or track students' code generation and provide feedback on program statements. To date, however, there does not exist work on implementing cognitive models of programming that simulate different strategies based on a programmer's knowledge (novice vs. expert).

The present paper takes a step towards filling this gap. Specifically, we identify the knowledge representations needed to program and embed them within computational ACT-R models capable of producing solutions to simple programming problems. Programming was chosen as the domain as it represents a complex problem-solving skill, with competing frameworks providing insight into the process programmers engage in while writing programs. Two models are implemented that simulate a novice and expert approach, respectively, to programming.

## **Novice and Expert Programming Approaches**

Since the models we implemented are influenced by theories of expertise, we begin with a brief overview of these, focusing on work in the programming domain. Programmers' mental representations are pivotal to programming performance and ability. For instance, programmers find it easier to read and understand the output of a program when the language uses functions that align with the programmer's underlying problem-solving strategy (Soloway et al., 1983). Of particular interest for the present work are studies investigating programming expertise (Spohrer et al., 1985; Soloway & Ehrlich, 1984).

Early programming frameworks characterizing novice programming focused on identifying the origins of common bugs in novice programmers' code. Spohrer et al. (1985) used a representational framework called GAP trees (Goal and plan networks) to parse the programs of novice programmers, categorize bugs, and identify the problem-dependent knowledge that led to bugs. The GAP framework decomposes a program using a solution space containing a program's goals, and the set of plans that implement those goals (e.g., through decomposition into smaller goals and plans). Spohrer et al. referred to this solution space as a GAP tree (goal-and-plan tree). There are two types of GAP trees for programs: (1) inferred trees, defined as having goals with multiple executable plans, and (2) solution subtrees, which are branches in the larger inferred GAP tree linking a single execution plan to a goal. Students who were not able to correctly complete programming tasks usually had an error in, or the complete absence of, one or more of the GAP tree components. This suggests that novice errors are caused by missing goal(s), or by incorrect knowledge representation(s).

Rist (1989) also studied novices, by analyzing the program-generation process of 10 novices to identify how they used simple programming plans to compose larger, more complex plans. In this study participants were asked to solve programming problems on paper while thinking out loud during their problem-solving process. Similar to Soloway's (1986) conceptual framework, Rist analyzed novice use of goals and plans, codeding the transcripts according to the plans implemented and their order of implementation. The findings showed that novice programmers used the primary goal of a problem to try and identify a set of known, basic programming plans that could be combined to resolve the goal. Novices first identified a *plan focus*, which is the first expression or line of a programming plan that is implemented; the plan focus served as the anchor for a given programming plan. Once the plan focus was implemented, the remainder of the plan was expanded around it (referred to as program expansion).

Soloway (1986) used the results of prior studies (Soloway and Ehrlich, 1984; Spohrer et al., 1985) to develop a conceptual framework describing expert programmers' problem-solving approaches. Soloway proposed that expert programmers first obtain an understanding of the goal and plan structure of the problem i.e., develop a rough GAP tree. Experts then use stepwise refinement, which is the breakdown of a problem on the basis of simpler problems the programmer has already solve; the solutions for the simpler problems provide the solutions to create the solution to the current problem. Soloway's framework proposed that novices have difficulty identifying the goals needed to solve the problem, as well as face difficulties recalling appropriate plans needed to implement the goals. In contrast, expert programmers use plan composition to combine the fragments of canned solutions into a final solution plan.

Overall, work described thus far suggests that a key difference between expert and novice programmers relates to the ability to generate plans (i.e., algorithms in the programming domain). Novices are unable to generate a plan either because they lack key information or because they are unable to link programming steps together.

## **Computational Models of Programming**

Prior work has used ACT-R to create cognitive models capturing processes related to programming. For instance, the ACT-R Programming Tutor (APT), developed by Corbett (2000), can write small programs. APT engages in both knowledge tracing and model tracing. Knowledge tracing is used to assess the probability that a student has successfully learned a rule based on application of the rule. For model tracing, the tutor uses an underlying production system, called its ideal student model, which contains the full set of rules to solve all of the practice problems. For each student input, once the student has selected their next goal and next step, the model tracer generates all possible correct next steps and compares these to the student's input. If the student input is correct, problem solving proceeds to the next goal-step combination. If the student's input does not match any of the model's steps, the tutor provides feedback and encourages the student to correct the mistake. The model-tracing component can write the small programs as it has the relevant productions, but it does not taken into account programming strategy.

Soloway's conceptual framework of programming plans discussed above does not formalize plans within a computational model. This was partially addressed by PROUST, a model built by Johnson and Soloway (1985), which could identify strategies in programs students wrote. PROUST took as input finished student programs and parsed programs by identifying the strategy/goal these decomposition used in the program. PROUST used its knowledge base of programming plans, strategies, and bugs to map out the solution path. This allowed PROUST to parse a program and identify deviations from the expected programming plan. Thus, PROUST could identify buggy programs and diagnose the source of the bug(s). While this model could identify strategies used to write a program, it was not designed to write programs.

In sum, to the best of our knowledge, there does not exist a computational model that takes into account programmers' strategies to write programs or that models the differences between expert and novice programmers.

# Present Work: Cognitive Models of Programming

We now describe two ACT-R models we implemented, called the *goal expansion* model and the *SGOMS* model. Each model aims to produce a solution to a basic programming problem using the programming language Python – one model simulates a novice approach to solving the problem and the second model an expert approach. Both models solved the rainfall problem, which requires calculating the average of all the positive numbers (including 0) in a list of daily rainfall amounts, and to stop processing the list if a value of -999 is encountered.

We obtained data on the impact of expertise on programming strategies from a case study we conducted (Vorobeva & Muldner, 2022). In the this study, 12 novice and 7 expert programmers were asked to solve the rainfall problem. While they worked on the problem, participants were asked to think-out loud by verbalizing their thoughts, so that data could be obtained on their reasoning and strategies. The data was analyzed using a qualitative approach to identify participants' goals and problem-solving approaches, and subsequently informed the design of the two cognitive models we present here that were implemented to solve the same problem.

The two models were implemented using Python ACT-R. Like ACT-R, Python ACT-R distinguishes two types of memory, declarative and procedural. Declarative memory stores information using *chunks*. In the present context, chunks include both steps (here, lines of Python code) and goals representing higher-level strategies. The declarative memory represents information that is known but not immediately actionable. In contrast, the procedural memory encodes *productions*, which are if/then statements that perform actions when their preconditions are met. The preconditions correspond to chunks in the declarative memory. These productions are used to generate the Python program "steps" (program lines), as will be described shortly.

#### **Model Components: Overview**

As noted above we implemented two models for simulating programming performance. In our framework, a model corresponds to the set of productions that define the expert and novice problem-solving approaches (the nature of the differences between the models will be discussed in the next section). The productions rely on information chunks stored in the declarative memory and module buffers (described below), for their preconditions. While the two models simulate different problem-solving strategies (novice vs. expert), they both rely on the same modules. Modules are specialized components in Python ACT-R, specifying distinct functions of the mind (Stewart & West, 2007). A given model within Python ACT-R may rely on a number modules to help carry out its problem-solving process within the environment. Modules exist outside of the model and are called upon by the model using the appropriate buffer that relays commands from the model to the appropriate module (and may also relay information from the module to the productions, as is the case for the DM module). The modules used by both models include (a) the motor module; (b) the environment; and (c) the declarative memory module (see Figure 1 for a visual of the modules and their relations). We now describe the modules and related buffers.

The motor module writes the Python program to a file and produces a log of the program goals and steps. Thus, the log shows a detailed trace of the problem-solving process. The motor module has a corresponding motor buffer (see Figure 1), which is used by the model's production to control the motor module's behaviour.

The environment module contains the description of the rainfall problem for each model. The environment is the same for the two models. Information from the environment is not mediated through a buffer.

The declarative memory module is a general component of the Python ACT-R architecture. The model uses a buffer to communicate with the declarative memory (see *DM Buffer*, Figure 1), and can use the buffer to add chunks to the declarative memory (e.g., reflecting new goals identified) or retrieve chunks from memory. Sometimes the declarative



Figure 1: Key system components. The arrows represent the direction of information flow. Problem-solving models receive information directly from the environment, send instructions to the motor module to implement actions in the environment (via the motor buffer) and share information bi-directionally with the declarative memory (through the DM buffer) to manage the problem solving process.

memory may make mistakes and fail to retrieve facts, or retrieve an incorrect fact that matches some of the query terms. This reflects that people will not always correctly recall information. During such events the model will be redirected to repeat the retrieval – it will retrieve the correct fact with sufficient attempts. Additionally, in cases where multiple facts match the query terms, retrieval is determined using a probabilistic calculation, where the association strength of the fact (a measure of how often the fact is retrieved) determines its probability of being retrieved. While both models use the same declarative memory module, they are initiated with different information within.

Both models are initialized with a focus buffer, which tracks where the model is in the program-generation process. Unless otherwise stated, the focus buffer holds chunks corresponding to the primary preconditions that must be met for a production to fire.

While both models produce code and corresponding goals associated with the solution process, neither model is capable of *learning* new chunks or productions, i.e., the models do not infer new algorithms or programming syntax through experience. Instead, the models are initialized with this information by the human model builder (and so for the present work, each of the two models' declarative memory was initialized prior to problem solving). Both models do add goals to the declarative memory, but they do not reflect learning of new goals as the goals are generated by the model's productions and thus already exist within productions of the model (though unspecified for the problem at hand).

#### **Two ACT-R Models for Program Generation**

Goal Expansion Model This model is inspired by Rist's (1989) framework characterizing novice programming and uses the goals and steps identified in Vorobeva and Muldners' (2022) study. In this framework, novices first identified a plan focus, and then expanded the plan focus by implementing the program steps (e.g., lines in the Python program). However, unlike the Rist framework, our goal expansion model can identify several goals (more than one) directly from the problem statement, based on keyword - goal associations - this translates the problem text into high-level goals. Goals correspond to the intention to perform a highlevel programming action, needed to solve the problem; for example, calculating the average rainfall. The model expands these goals into related goals based on goal - goal associations. For example, once the model generates the goal to calculate the average rainfall, it generates the related goal to initialize the variables needed for the average calculation.

Like the behavior of novices in prior work, the goal expansion model does not generate a high-level algorithm that orders the goals and steps in advance. Instead, the model identifies and addresses goals in the order it retrieves the relevant goal associations. Once a goal is generated by the model, either from reading the problem statement or after goal expansion from one of the keyword-associated goals, the chunk representing the goal and associated step is retrieved from the declarative memory and stored in the DM buffer. The retrieved chunk stored in the DM buffer satisfies the precondition for the firing of the production that implements the step (i.e., line of Python code) that resolves the goal. The step-implementing production sets the model's focus buffer to contain the precondition used by the production that generates other related goals, i.e., directs the model to engage in goal expansion. If a goal is generated using a goal - goal association, it goes through the same implementation process as described above; it will subsequently be used to check for further goal-goal associations.

**SGOMS Model** SGOMS is a cognitive framework that adds planning units and unit tasks to ACT-R in order to model complex behavior (West & Pronovost, 2009; West & Nagy, 2007; West & MacDougal, 2015). Planning units represent goals, such as *calculating average rainfall* and *initializing variables*, and reflect the goals identified during the coding of the participants' verbal protocol and written program in Vorobeva and Muldner (2022). Planning units are used to initialize the model's declarative memory at the start of problem solving and structure the problem-solving process. Each planning unit is composed of unit tasks that must be completed to resolve the planning unit; collectively. The unit tasks for a given planning unit will be referred to as a subalgorithm. Unit tasks can either define high-level operations or implementational-level operations.

Implementational unit tasks are what the model uses to control implementation of the step, and reflect actions that must be taken to implement the code, as well as to make the written code fit with the rest of the programmed solution. High-level unit tasks are used to implement a planning unit hierarchy by allowing planning units to call upon other planning units as part of the initial planning unit's subalgorithm. This reflects that the resolution of some goals requires the resolution of other goals, and that this creates a sort of goal hierarchy. When a high-level unit task calls another planning unit (when the planning unit goal requires another goal to be resolved), it redirects the model to that new planning unit and this unit must be completed first. Once the called upon planning unit is complete, i.e., the unit tasks that defines its sub-algorithm have all been completed, the model redirects to the next unit task of the calling planning unit. For example, the *calculate average rainfall* planning unit has as its first unit task to call upon the *initialize* variables planning unit. This redirects the model to resolving the initialize\_variables planning unit (by writing the code to initialize the variables) before continuing to the next unit task, namely the *calculate\_average* planning unit.

The model begins program generation by calling on the highest-level planning unit relevant to the problem. For the rainfall problem, this is the calculate average planning unit. This planning unit is considered the highest level as it defines a sub-algorithm for the primary goal stated in the problem statement (to calculate the average rainfall). The subalgorithm includes unit tasks for both high-level (productions requests to other planning units) and implementational-level productions (requesting variables / conditions and implementing steps). The calculate average planning unit will first require the completion of two other planning units (initialize variables and iterate loop). However, as described above, the called upon planning units may themselves call additional planning units, such as the iterate\_loop planning unit calling upon the stop\_loop and track\_variables planning units. When a planning unit is complete, it directs the model to the next unit task in the planning unit that called it. The program is complete when the highest-level planning unit implements its final unit task; for the rainfall problem this corresponds to the expression that calculates the average in the Python program.

By using planning units to organize information, the program-generation process is guided, but without the need to generate the entire algorithm in advance. In this way the SGOMS model more closely mimics the behavior shown by experts and competent novices in our study (Vorobeva and Muldner, 2022). By relying on planning units instead of a pre-canned algorithm, the model has the ability to recombine the planning units to generate different solutions. This reflects the ability of the SGOMS model to be flexible with its treatment of goals. For example, the SGOMS model is currently capable of generating a simple loop function that sums and counts all of the numbers in a list but that does not give an average for the positive numbers.

**SGOMS Model vs. Goal Expansion Model** As is the case with the SGOMS model, the goal expansion model is not given a complete algorithm up front. The goal expansion model relies on associations in its declarative memory to generate the goals, but can not specify the exact relationship between associated goals. Consequently, the goal expansion model has trouble implementing steps in a coherent order. In contrast, the SGOMS model has a concrete structure and hierarchy to the goals that is well defined before problem solving. However, it does not have a pre-existing complete algorithm that defines the implementation of the total solution. For example, the planning unit to calculate average rainfall initializes the planning units for iterating the loop and initializing the variables. However, the calculate average rainfall planning unit does not define which planning units need to be initialized by the other planning units. Therefore, each planning unit functions semi-independently, and can be called upon by any number of other planning units, as long they are defined by the modeler or by learning mechanisms in advance. In this way planning units may be recombined to generate solutions to new problems, something the goalexpansion model would struggle with.

## Simulation of Program Generation via each Model

As described above, both models were implemented using Python ACT-R and initialized with the specification of the rainfall problem. When we ran each model to simulate the problem-solving process by an expert (SGOMS model) and a novice (goal expansion model), the SGOMS model was able to produce a correct solution but the goal expansion model was not. We now describe each model's problem-solving process and output.

Goal Expansion Model Figure 2 shows the output for the goal expansion model. The model was able to form varied solutions to the rainfall problem, because there was no set order of how to address the goals. However, it did not generate a correct program (possibly with sufficient runs it would accomplish it by chance). Specifically, the model had difficulty correctly ordering the program steps (recall that a step corresponds to a single line of Python code). For example, it identified the goal to calculate the average (Figure 2 line 1) and then wrote the line to the top of the Python file to accomplish the goal (Figure 2 line 2). However, the variables that were needed to calculate the average had not yet been initialized or incremented within the loop function (done in Figure 2 lines 4 and 6 respectively). Therefore, the written program would be unable to go through the program at all as it would not have anything assigned to the variables when asked to calculate the average. In general, the goal expansion model currently generates solutions based on the order of keywords it extracts from the problem statement. Thus, adding more refined NLP functionality is needed to appropriately assess its validity as a model of novice programming.

The model produced *some* of the behavior Rist (1989) attributed to novices. It identified goals from the problem statement, and engaged in program expansion to add additional goals and steps. This allowed the model to connect the steps of iterating through the list (a and stopping the loop (Figure 2 lines 7-10). By expanding from a keyword – goal identified plan focus (in this example the keyword – goal plan focus was list - iterating the list), it correctly connected the

1.Goal: [calculate average] I need to: calculate\_average
2.Step: < calculation > average = total/count
3.Goal: [initialize variables count/total] I need to: initialize\_variables
4.Step: < initialize variables > total= 0, count= 0
5.Goal: [track count/total] I need to:track\_variables
6.Step: < condition > if x >= 0:
< increment variables > count+=1, sum+= x
7.Goal: [iterate through list] I need to: loop\_iterate
8.Step: < iterate loop > for x in rains:
9.Goal: [stop loop] I need to: stop loop iterate

10.Step: < stop loop > if x == -999: break

## Figure 2: Log of Goal Expansion Model's Problem-Solving Process

steps together, but was unable to connect both expressions to the broader problem statement of calculating the average. However, the model was also more sporadic in terms of the ordering of its solution goals / steps. We discuss potential reasons for this in the discussion.

**SGOMS Model** Figure 3 shows the output from the SGOMS model. The SGOMS model was able to successfully construct the canonical solution as well as a complete algorithm specific to the problem (note the complete algorithm was not provided to it *a priori*). Additionally, it was able to replicate findings from Vorobeva & Muldner (2022) of experts identifying multiple goals before implementing them (Figure 3 lines 1 and 2), though this ability was restricted to the main goal of calculating the average rainfall. This goal is identified at the start but not implemented until the very end (Figure 3 lines 1 and 10).

## Discussion

The aim of the present work was to leverage earlier work on expert and novice programmers' problem solving to develop models capable of program generation. Specifically, the common goals and steps we identified in both expert and novice solutions (Vorobeva and Muldner, 2022) were used to create the declarative knowledge chunks (goals - step) and productions that implemented the step of the solution. Earlier models such as PROUST and APT were capable of processing programs but were unable to write whole solutions for a programming problem. APT had the knowledge base to write small snippets of code, that is expressions that would address one goal of an overall problem, but was unable to chain them together into a complete final solution. While our models are limited in scope in terms of their capacity to solve a range of programming problems, they are capable of identifying and implementing multiple goals and linking them together to provide an overall solution pathway (albeit not a correct one in the case of the goal expansion model).

We expected novices to be best modelled by the goal expansion model, which reflected Rist's (1989) framework of a novice approach to problem solving. Rist argued that

Figure 3: Log of SGOMS Model's Problem-Solving Process

novices use a *plan focus*, that is a core step that is written first in the program, and *program expansion* to expand the *plan focus* by implementing additional steps which supported the *plan focus* step. The goal expansion model was able to identify multiple plan focuses from the problem statement using the *keyword* – *goal* associations in its declarative memory, and expand around those plan focuses with relevant goals. However, many novice participants in our study (Vorobeva & Muldner, 2022) did not rely on or use a plan focus as predicted by the Rist framework. Hence neither the Rist framework or the goal expansion model accurately modelled all of the novices.

Differences between the performance of some novices and the model of novice behaviors (i.e., the goal expansion model) may be due to the simplicity of the rainfall problem not requiring a more generative problem-solving process by not requiring many goals and steps. Additionally, the model's lack of a revision and reflection mechanism made it difficult to engage in program expansion, as the model could not write steps that would precede other already written steps. For example, the model was not able to initialize variables at the top of the file if it had already implemented the loop that incremented them. Thus, the goal expansion model was limited to only depicting strict forward expansion, where program expansion would follow the same order as the final working solution. In earlier work, Byckling and Sajaniemi (2006) found that strict forward expansion occurred only in more competent novices, and thus the goal expansion model is limited as a model of all novices.

The SGOMS model best represented the performance of the experts and competent novices, as it produced output that showed the greatest degree of similarity to the outputs produced by experts (and some novices) as determined by a qualitative analysis. It replicated some of the expert's behaviours, such as identifying multiple goals in a row (without step implementation), thus demonstrating some preplanning capabilities.

While the models were informative, there are various improvements we are working on. For instance, the models could benefit in terms of validity if they had productions capable of reflecting on and revising the programs written (this would allow the model to insert written code in between or in front of existing lines of already written code, as needed). This problem with the lack of reflection and revision is most apparent in the inability of the models to capture the novice and expert ability to engage in backwards program expansion (the models are only capable of strict forward expansion). This could be implemented through the expanded use of planning units in the future. Moreover, given that the present models have only been tested on a single problem, more work is needed to test and generalize them with a range of problems.

In the future it might also be beneficial to extend the model with the ability to construct a GAP tree of the type described in prior work (Spohrer et al., 1985; Soloway 1986). A GAP tree would allow the model to have a high-level representation of the overall problem and the current state of the problem and would supplement the existing hierarchical structure of the planning units. For problems more complex than the rainfall problem, the current version of the SGOMS model may have difficulties managing more complex arrangements of goals/planning units and determining the best implementation order. One potential solution to this issue could be to add an additional buffer that constructs and tracks a hierarchy tree of goals. Another possibility is the addition of declarative knowledge of how to best arrange multiple subgoals during implementation (such as ensuring that the step for variable tracking is always written within the loop iterating the relevant list) to the declarative memory.

In spite of these limitations, the models presented in this paper matched some of the novice and expert performance from our study (Vorobeva and Muldner, 2022). Additionally, they were capable of capturing various problem-solving strategies from the study conducted and prior research.

## Acknowledgments

This work was supported by an NSERC Discovery Grant (#1507).

## References

- Anderson, J. R., & Lebiere, C. (1998). The atomic components of thought. Lawrence Erlbaum Associates.
- Braithwaite, D. W., Pyke, A. A., & Siegler, R. S. (2017). A computational model of fraction arithmetic. *Psychological Review*, 124(5), 603–625.
- Corbett, A. (2000). Cognitive mastery learning in the act programming tutor. In *Adaptive User Interfaces*. AAAI SS-00-01.
- Frischkorn, G. T., & Schubert, A. L. (2018). Cognitive models in intelligence research: Advantages and recommendations for their application. *Journal of Intelligence*, 6(3), 34-56.
- Johnson, W. L., & Soloway, E. (1985). PROUST: Knowledge-based program understanding. IEEE Transactions on Software Engineering, 3, 267–275.
- Pirolli, P. (1986). A cognitive model and computer tutor for programming recursion. *Human-Computer Interaction*, 2(4), 319-355.
- Rist, R. S. (1989). Schema Creation in Programming. Cognitive Science, 13(3), 389–414.
- Sajaniemi, J. (2002). An empirical analysis of roles of variables in novice-level procedural programs. In *Proceedings IEEE 2002 Symposia on Human Centric Computing Languages and Environments*, 37–39. https://doi.org/10.1109/HCC.2002.1046340
- Soloway, E., Bonar, J., & Ehrlich, K. (1983). Cognitive strategies and looping constructs: An empirical study. *Communications of the ACM*, 26(11), 853–860. https://doi.org/10.1145/182.358436
- Soloway, E., & Ehrlich, K. (1984). Empirical studies of programming knowledge. *IEEE Transactions on software* engineering, (5), 595-609.
- Spohrer, J. C., Soloway, Elliot, & Pope, E. (1985). A goal/plan analysis of buggy Pascal programs. *Human–Computer Interaction*, 1(2), 163–207.
- 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
- VanLehn, K., Randolph, J. M., & Chi, M. T. H. (1991). Modeling the Self-explanation Effect with Cascade 3. In Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society, 132–137.
- Vorobeva, M., & Muldner, K. (2022). Investigating Expert and Novice Programming Problem Solving. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 44.
- West, R. L., & Pronovost, S. (2009). Modeling SGOMS in ACT-R: Linking Macro- and Microcognition. Journal of Cognitive Engineering and Decision Making, 3(2), 194– 207. <u>https://doi.org/10.1518/155534309X441853</u>
- West, R. L., & Nagy, G. (2007). Using GOMS for modeling routine tasks within complex sociotechnical systems: Connecting macrocognitive models to microcognition. *Journal of Cognitive Engineering and Decision Making*, 1(2), 186-211.

West, R. L., & MacDougall, K. (2014). The macroarchitecture hypothesis: Modifying Newell's system levels to include macro-cognition. *Biologically Inspired Cognitive Architectures*, 8, 140-149.

# Beyond Responding Fast or Slow: Improving Cognitive Models of Memory Retrieval using Prosodic Speech Features

Thomas Wilschut (t.j.wilschut@rug.nl) Department of Experimental Psychology, University of Groningen

Florian Sense (florian.sense@infinitetactics.com) InfiniteTactics, LLC

Odette Scharenborg (o.e.scharenborg@tudelft.nl) Multimedia Computing Group, Delft University of Technology

Hedderik van Rijn (d.h.van.rijn@rug.nl) Department of Experimental Psychology, University of Groningen

**Keywords:** Adaptive Learning, ACT-R, Cognitive Modeling, Speech Features, Prosody

#### Background

Technology plays an increasingly important role in education. Digital adaptive learning (AL) systems have successfully improved the efficiency of fact and word learning by tailoring learning procedures to the needs of individual learners (Lindsey, Shroyer, Pashler, & Mozer, 2014; Papousek, Pelánek, & Stanislav, 2014; Van Rijn, Van Maanen, & Van Woudenberg, 2009). AL systems typically track learning performance (measured using typed responses to practice problems) in real time and use this information to provide personalized feedback, select appropriate practice materials or optimize item repetition schedules. The effectiveness of AL systems critically depends on their ability to estimate the extent to which learners have successfully memorized study materials. Some AL systems employ a cognitive model of memory retrieval to estimate the strength of item representations in the learners' memory. For example, the SlimStampen system (Van Rijn et al., 2009) is based on the ACT-R architecture's model of human declarative memory (Anderson, Bothell, Lebiere, & Matessa, 1998) and functions by measuring response times (RTs) and accuracy scores to determine optimal item repetition schedules. The system relies on the assumption that RTs are a good proxy for the strength of fact representations in memory: The quicker the learner produces a correct response, the stronger the memory representation for that item is assumed to be (Anderson & Schooler, 1991; Jescheniak & Levelt, 1994; Levelt, 1999; Van Rijn et al., 2009) The above-described approach uses the limited information available (RTs and accuracy) to estimate memory strength for typed retrieval attempts and uses this information to optimize item repetition schedules.

Recent advances in speech technology have allowed for the transition from typing-based AL systems to speech-based AL systems (Wilschut et al., 2021). In speech-based learning, there is additional information that AL models can use to estimate the strength of item representations in memory: Spoken language contains prosodic speech features (PSFs), which are supra-segmental properties of speech (Xu, 2011). PSFs are commonly used by speakers to convey information beyond the literal meaning of the utterance, and can be roughly divided into three categories: Intonation, the melodic pattern of

an utterance, defined by the dynamics in pitch over the duration of a speech segment; rhythm, the dynamics in timing and speaking speed of a speech segment; and stress, which refers to the intensity that is given to a syllable of speech, resulting in changes in relative loudness.

Here, we aim to examine if spoken retrieval attempts contain information that goes beyond what is already encapsulated in the RT and accuracy scores for that retrieval attempt. We hypothesize (1) that prosodic speech features are associated with retrieval accuracy and (2) that PSFs carry information that can be used - in addition to RTs and accuracy scores to more accurately estimate the extent to which a learner has successfully memorized an item, and predict later retrieval success.



Figure 1: Design and research questions. Participants saw a cue (see Methods) and responded using speech. Using ASR, the accuracy of the response is determined. The first research question examines if PSFs derived from the speech signal are associated to accuracy (ACC) on the same repetition (left question mark). The second research question considers if previous-repetition PSFs can be used to explain current-repetition accuracy (right question mark).

## Methods

A graphical description of the design and research questions is shown in Figure 1. Fifty participants studied Swahili-English vocabulary items using the SlimStampen adaptive scheduling system. Swahili items were presented on a computer screen, and participants were asked to respond by pronouncing the English translation of the item. Participants' utterances were transcribed to text in real time using Google Cloud Speech-to-Text (see https://cloud.google.com/speechto-text) automatic speech recognition (ASR). Speech features were extracted afterwards using Praat 6.2.07 (Boersma, 2006).

#### Results

The results of this study are twofold. The first part concerns the relationship between speech features and retrieval accuracy. As hypothesized, the accuracy of retrieval attempts was associated with specific speech feature characteristics. More specifically, higher retrieval accuracy was associated with falling pitch (negative pitch slope), higher loudness and higher speaking speed (r(7847) = -0.10, p < .001; r(7847) =0.05, p < .001; r(7847) = 0.07, p < .001, respectively). Second, we explored the possibility of using PSFs to explain next-repetition retrieval accuracy. We conducted a logistic mixed-effects regression model to explain current-repetition accuracy using (1) model-based activation estimations, (2) previous-repetition pitch slope, (3) previous-repetition speaking speed and (4) previous-repetition loudness, see Table 1. As expected, model-based activation, estimated using pastrepetition RTs and accuracy scores, significantly explained accuracy (z = 4.60, p < .001, see Figure 2A and Table 1). Importantly, previous-repetition pitch slope and previousrepetition speaking speed also explained variance in currentrepetition retrieval accuracy (z = -2.60, p = .008; z = 3.37, p <.001, see Table 1 and Figure 2B and Figure 2C, respectively). Loudness did not significantly explain next-trial accuracy (z = 0.82, p = .412, see Table 1). Adding the previous-repetition PSFs to a model with only model-estimated activation as independent variable resulted in an 16% increase in explained variance in current-repetition retrieval accuracy ( $R^2 = 0.162$ ) and  $R^2 = 0.135$  for the model with and without PSFs, respectively). Together, our results show that we can use pitch dynamics and speaking speed in addition to RTs and accuracy scores to improve explanations of next-trial retrieval accu-

Table 1: Logistic mixed-effects regression model explaining current trial accuracy from model-estimated activation and previous-trial PSFs.

racy.

	β	SE	z	p
Intercept	1.63	0.12	13.56	< 0.001
Activation <sub>n</sub>	-0.45	0.10	4.60	< 0.001
Pitch slope $_{n-1}$	-0.15	0.06	-2.66	0.008
Loudness <sub>n-1</sub>	0.04	0.05	0.82	0.412
Speaking speed <sub><math>n-1</math></sub>	0.18	0.05	3.37	< 0.001

#### Conclusion

We show that spoken retrieval attempts contain information about the extent to which a learner has memorized an item, and that PSFs can be used to improve model predictions for



Figure 2: Explaining retrieval accuracy on the current trial (n) using (A) memory activation estimated by a response timebased ACT-R model of memory retrieval, and (B-C) high level PSFs on the previous retrieval attempt for the same item (n-1). Dots show empirical accuracy, lines show accuracy estimations.

learner performance on future trials. Our results are important in two ways. First, they have theoretical implications, as they elucidate how speaker accuracy is reflected in speech prosody: to our knowledge, we are the first to demonstrate that inaccurate and slow responses are associated with a rising pitch, low vocal loudness and low speaking speed, suggesting that PSFs can be used as a measure of speaker certainty or confidence. More generally, PSFs may prove to be a valuable new tool in the further exploration of important open research questions (e.g., about speaker certainty/confidence or feeling-of-knowing and a range of other meta-memory judgements). Second, our results have practical implications, as they can contribute to the further development of speechbased AL systems. We show that PSFs can be used to improve AL model accuracy predictions. Importantly, compared to more traditional (deep-learning-based) approaches to automatic speech processing, extracting PSFs from the speech signal is computationally inexpensive, making them especially suitable to be used in real-time AL applications. In short, we show that PSFs are a promising candidate to be used in educationally relevant speech-based learning applications.

#### References

- Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language*, 38(4), 341–380.
- Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological science*, 2(6), 396–408.
- Boersma, P. (2006). Praat: doing phonetics by computer. *http://www.praat.org/*.
- Jescheniak, J. D., & Levelt, W. J. (1994). Word frequency effects in speech production: Retrieval of syntactic information and of phonological form. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20(4),

824.

- Levelt, W. J. (1999). Models of word production. *Trends in cognitive sciences*, *3*(6), 223–232.
- 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.
- Papousek, J., Pelánek, R., & Stanislav, V. (2014). Adaptive practice of facts in domains with varied prior knowledge. In *Educational data mining 2014* (pp. 6–13).
- Van Rijn, H., Van Maanen, L., & Van Woudenberg, M. (2009). Passing the test: Improving learning gains by balancing spacing and testing effects. In *Proceedings of the 9th international conference of cognitive modeling* (Vol. 2, pp. 7–6).
- Wilschut, T., Sense, F., Van der Velde, M., Fountas, Z., Maass, S., & Van Rijn, H. (2021). Benefits of adaptive learning transfer from typing-based learning to speechbased learning. *Frontiers in AI and Big Data*, *4*.
- Xu, Y. (2011). Speech prosody: A methodological review. *Journal of Speech Sciences*, 1(1), 85–115.

# A Model of Motivation and Effort Allocation in the ACT-R Cognitive Architecture

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

Motivation is the driving force that influences people's behaviors and interacts with many cognitive functions. Computationally, motivation is represented as a cost-benefit analysis that weighs efforts and rewards in order to choose the optimal actions. Shenhav and colleagues (2013) proposed an elegant theory, the Expected Value of Control, which describes the relationship between cognitive efforts, costs, and rewards. In this paper, we propose a more fine-grained and detailed motivation framework that incorporates the principles of EVC into the ACT-R cognitive architecture. Specifically, motivation is represented as a specific slot in Goal buffer with a corresponding scalar value, M, that is translated into the reward value  $R_t$  that is delivered when the goal is reached. This implementation is tested in two models. The first model is a high-level model that reproduces the EVC predictions with abstract actions. The second model is an augmented version of an existing ACT-R model of the Simon task, in which the motivation mechanism is shown to permit optimal effort allocation and reproduce known phenomena. Finally, the broader implications of our mechanism are discussed.

Keywords: Motivation, Cognitive Control, Effort, Computational Modeling, Cognitive Architecture

#### Introduction

Observable behavior in cognitive tasks is affected by the degree to which a participant puts effort into the task. The driving force behind this effort allocation is usually called motivation and represents a significant obstacle in properly inferring individual characteristics from observations. For example, a participant performing poorly in an N-back task might be poorly motivated to perform the task, rather than having limited working memory capacity. Despite its importance, motivation is rarely modeled or accounted for in cognitive models. In this paper, we outline a theory of motivation implemented in the ACT-R cognitive architecture and demonstrate its application.

To understand motivation from a cognitive modeling perspective, it is necessary to clarify the definition and relationships between several important constructs. *Motivation* is not directly observable. It is usually described as a driving force or invigorating impact on behavior or cognition that initiates a goal-oriented behavior. That is to say, we can only infer one's motivation from his behavior and cognition. *Effort* refers to how many cognitive resources one would allocate to a particular activity in order to achieve the goal. According to Inzlicht, Shenhav, and Olivola (2018), *Motivation* specifies both direction and intensity of goal-oriented behavior, while effort only indicates the intensity of any possible action, without reference to any goal. *Demand* is different from *Effort* in that it is the descriptive property of the task or environment,

while *Effort* indicates the magnitude of the force that an individual might apply toward the environment. Other cognitive states such as mental fatigue, curiosity, and high arousal may interact with motivation in certain ways to have crucial impacts on learning, memory, and other cognitive control functions therefore, cognitive modeling gives us a unique opportunity to parse apart the specific effect of motivation alone.

# **Expected Value of Control Theory**

Although several attempts have been made to capture motivation within a computational framework (e.g., Niv, 2007), the current dominant theory is the Expectancy Value Theory. It was first proposed by Voom in the 1960s and recently expanded into a formal theory known as the Expected Value of Control (EVC) model by Shehav et al. (2017). The EVC model assumes that individuals would evaluate cost-benefit tradeoffs in order to maximize gains and minimize costs in deciding how much cognitive effort one would allocate to the chosen action, as shown in Fig 1(A, B).

According to the EVC model, the expected value of control is determined by the expected reward and efficacy of the task. The expected reward indicates the expected outcomes of achieving the goal (e.g., monetary incentives) and efficacy refers to how likely the goal will be achieved by allocating a certain amount of control and expending a certain amount of effort (time). Computationally, the EVC model specifies that cognitive effort is allocated based on two dimensions: 1) identify the object (what to attend); 2) intensity (how much effort to allocate, compared to default level). A key assumption of this model is that intrinsic cost would be associated with higher control intensity. At the neural level, the translation between the expected value (i.e., the difference between expected rewards and costs) and corresponding effort allocation is mediated by the dorsal anterior cingulate cortex (dACC), a region that is known to play a critical role in linking adjustments in performance (Botvinik et al., 1999) with task feedback (Holroyd et al., 2004), error learning (Yeung et al., 2004) and with expected rewards (Adcock et al. 2006).

Thus defined, the EVC is an elegant, comprehensive, but highly abstract framework: it does not provide a direct mechanism by which costs and rewards are computed and associated to specific cognitive steps, nor it does make specific predictions about how motivation would precisely shift how an individual performs a task. To do so, we need a more fine-grained and detailed theory of human cognition. One such prominent theory is the ACT-R cognitive architecture (Anderson 2007).

# ACT-R

ACT-R is the most prominent and successful cognitive architecture in psychology and neuroscience (Kotseruba and Tsostsos, 2020). Surprisingly, despite the high relevance of motivation to other cognitive functions and the apparent potential and an ACT-R model of motivation, the interaction between motivation and cognitive control has been largely overlooked in ACT-R literature. Several modeling attempts have been made in order to incorporate motivation-related constructs into ACT-R, such as intrinsic motivation (Nagashima et al., 2020), emotion (Smith et al., 2021), mental fatigue (Herlambang et al., 2021; Halverson et al., 2021), and depression (van Vugt and van der Velde, 2018).

In ACT-R, knowledge is represented in two fundamental formats: chunks and production rules. A Chunk is a vector-like structure that stores semantic or episodic memories. A Production rule (or simply production) is a basic action unit that represents procedural knowledge as an "IF-THEN" conditional statement. Productions and chunks interact through a set of modules which represent different cognitive processes. For example, a Visual module encodes visual information as chunks, and a Motor module transforms chunks into motor outputs. Most critical to this paper are the Goal module (holding current goal information), the Declarative module (storing all declarative memories and managing their availability for retrieval), and the *Procedural* module (maintaining, updating, and selecting productions).

Each chunk is associated with a scalar value, called activation, which represents the odds of a chunk being needed in the future (Anderson, 1998). Similarly, each production has a utility value which represents the expected future rewards associated with the execution of that production. In ACT-R, only one chunk can be retrieved and only one production can be fired at any time; thus, computing chunks are selected on the basis of their activations, and production rules are selected on the basis of their utility. Utilities are learned through experience. At any time point *t*, the utility *U* of production *p* is calculated based on Reinforcement Learning using Eq 1, where  $\alpha$  denotes the learning rate,  $R_t$  denotes the reward the production received at time *t*; *s* denotes the noise parameter.

$$U_{t} = U_{t-1} + \alpha (R_{t} - U_{t-1}) + s \qquad (1)$$

In ACT-R, rewards and costs are represented in time units. For instance, if the model fires a production P1 at  $t_1$  and it receives a reward  $R_{delivered}$  at  $t_2$ . The utility learning discounted the reward by the time it passed: The received amount of reward is:  $R_{received} = R_{delivered} - (t_2 - t_1)$ .

It should be noted that, in ACT-R, the above-mentioned *Goal* module is putatively associated with dACC (Anderson, 2007), but has no relationship to rewards and is, in fact, used only as a way to add additional information to

select between competing productions. This violates established findings in neuroscience and is incompatible with the EVC. It is also a major departure from early versions of the ACT-R architecture (e.g., Anderson & Lebiere, 1998), in which goals were associated with specific values, and values were explicitly used to rank productions on the basis of a cost-benefit analysis. This older framework was, in principle, much more compatible with the EVC theory. One of our objectives is to propose a framework that conserves the current RL-based utility mechanisms but connects it with explicit goal values, re-introducing some of the most desirable features of the previous implementations.

#### **Present study**

The goal of this paper is to outline a general framework of goal-oriented motivation in ACT-R that is consistent with the EVC theory and can be implemented and deployed in any ACT-R model. This framework assumes that the goal module assigns value to chunks representing goals, with this value representing the subjective reward associated with accomplishing the goal. This is implemented by adding to the current Goal chunk a special motivation slot that contains a numeric value M. Once the goal is achieved, M is interpreted as the amount of reward delivered in the end. At that moment, M is automatically translated into the  $R_t$  value that is used in Eq 1., and propagates back to previous productions. Because in ACT-R, rewards are represented in time units, the value M can be interpreted in two ways: as the subjective value associated with reaching a goal, and as the maximum amount of time the model is willing to spend on a particular goal. The first interpretation is consistent with the current interpretation of the reward value R<sub>t</sub>, while the second is consistent with the original interpretation of the goal value G in previous versions of ACT-R. By incorporating this mechanism, the Goal buffer is not only a passive recorder of task status, but an active power behind adaptive behaviors. Crucially, our model also attempts to account for where the intrinsic reward Rt comes from, and how motivation value M alters one's behavior by affecting the calculation of expected reward and effort.

We compare our motivation model to another well-known model of effort allocation and motivation proposed by Shenhav et al. (2013). We argue that ACT-R's procedural system provides an equivalent way of calculating the expected value of control as proposed in the EVC model. To prove that, we develop a simple effort allocation model in ACT-R, showing that ACT-R is capable of selecting the optimal strategy by weighing costs and rewards when making a decision, in line with EVC Theory. Further, we extend this simple model to a more complicated and realistic computational model of a cognitive interference task (the Simon task), augmenting it with the new motivation component. The result demonstrates that the proposed framework is compatible with the EVC model, and it helps us understand why cognitive systems vary widely in making decisions for engaging in effortful activities. Moreover, we propose a modeling approach for future ACT-R modelers that incorporates costs, rewards, and motivational components into cognitive function. All of the model and simulation codes and data are freely available at https://github.com/UWCCDL/ACTR-Motivation

# **Computational Models**

# Motivation and Effort Allocation in an Abstract Model

To demonstrate the relationship between EVC theory and the proposed ACT-R motivational framework, we first present a simple, abstract ACT-R model and simulate the expected value of control predicted by the EVC model. To translate the continuous effort allocated in the EVC model, the abstract model assumes that different amounts of effort correspond to ten possible productions, indicated as P1, P2 ... P10. The pre-conditions of these 10 productions are the same to guarantee that they are competing with each other. When the model starts running, only one of these 10 productions is selected based on the highest utility. Following this, an END production delivers a certain amount of reward at the end.

The 10 productions represent ways to perform the task that is associated with different rewards and costs in terms of mental effort. The reward is represented in terms of the probability of achieving the goal. The cost of production is represented by the time it takes to execute, which is controlled by a production-specific AT parameter (for "Action time") in ACT-R. This parameter represents the effort associated with each production and, in the EVC framework, the associated cost of cognitive control. By default, it takes 0.05 seconds to fire a production, in this simple model, we assign different :AT to 10 productions in ascending order (0.01-0.1). Larger :AT suggests that the model needs to allocate more resources (time) in order to achieve the goal, while smaller :AT suggests that it could quickly finish the task, without spending more time on it. Specifically, production P1 is assigned to the smallest :AT, and P10 is assigned to the largest :AT.

To model expected payoffs, we set various amounts of rewards for 10 productions, in ascending order (0 - 10). P1 is assigned to the lowest reward, while P10 is assigned to the highest reward. Following Musslick, S., Shenhav, A., Botvinick, M. M., & Cohen (2015)'s suggestion, we varied the costs of the different productions according to an exponential function and varied each production's probability of receiving a reward as a sigmoid function. Thus, assigned cost increases from P1 to P10 exponentially, and the delivered rewards increase from P1 to P10 following the sigmoid function. Simply put, P1 spends the least cognitive resources but also has the lowest payoff, while P10 spends the most cognitive resources but has the highest payoff.

Two experimental conditions were simulated, corresponding to the two theoretical cases discussed by Shehav et al (2006). The first is the effect of increased task difficulty. This was simulated by decreasing each

production's probability of obtaining the reward. The second was increasing the payoff. This was done by increasing the absolute value of M and, therefore, of the reward associated with accomplishing the task. In our framework, this is equivalent to simulating higher intrinsic motivation. We simulated 100 times per parameter set and 100 seconds (in ACT-R time) for each trial. During each trial, we recorded the counts of each production being selected to estimate the probability of selecting production. For each selected production, the received reward was also recorded to estimate the payoff.

It was expected that the probability of production being selected would show the same pattern predicted by the EVC model. Specifically, we hypothesized that, under different combinations of rewards and costs, the model would assign the greatest utility (and, therefore, the greatest probability of being selected) to the production that maximizes the difference between rewards and costs. both low-cost low-payoff productions (P1, P2), and high-cost high-payoff productions(P9, P10) are less likely to be selected than optimal cost-reward balanced productions (P6, P7, P8).

# Results

Fig 1(C) and (D) demonstrate the relationship between cost, reward, and expected value of control in the abstract ACT-R model. As expected, our simple mental effort allocation model generated identical patterns of cognitive control allocation as the EVC model does. It selected the optimal production by weighing costs and rewards through utility learning in Reinforcement Learning. At a medium level of difficulty and a medium level of payoff (Fig 1C), ACT-R selects the P7 production most frequently because the utility of P7 is the highest after subtracting costs from payoffs. As the task difficulty increases, ACT-R moves to selecting production P9 most frequently. In terms of the EVC model, ACT-R now allocates more resources (a more costly production) to obtain rewards. If, on the other hand, the task difficulty decreases, ACT-R switches to a less effortful production rule (P6), which guarantees similar rewards but with less costs (shorter times).

We observe similar patterns when the Payoff is manipulated (Fig 1D). In lower payoff, ACT-R chooses P4 most frequently. As payoff increases, ACT-R tends to allocate more resources to gain more rewards by selecting higher-cost higher-reward production P5.



Fig 1. The recreated plot of the EVC Model (A, B); the probability of production in the simple ACT-R model (C, D). In A) and B), x-axis represents control intensity, red curves describe the costs increase exponentially as control intensity, and green curves describe the payoffs curve. The purple curve describes the expected value of control. In C) and D), x-axis represents 10 productions assigned with increasing costs and increasing rewards; Green dots denote the reward each production received; Red dots denote the cost (:AT) of each production; Purple dots denote the probability of each production being selected given the reward and cost parameter. Line types denote two conditions: increase the difficulty (A, C) and increase payoff (B, D). In (A) and (C), dashed curves describe the lowest task difficulty, while solid curves describe the highest task difficulty. In (B) and (D), dashed curves describe the lowest payoff (received rewards), and the solid curves describe the highest payoff (received reward).

#### Motivation and Effort Allocation in a Realistic Task

With the simple ACT-R model of effort allocation, it is safe to say that ACT-R provides a mechanistic implementation of the EVC framework. This case, however, was highly stylized: the ten productions do not represent specific cognitive operations and their costs do not realistically reflect cognitive times; this level of detail is, by contrast, the very strength of ACT-R. To examine whether the motivation framework outlined above could be translated into a realistic ACT-R model of a cognitive task of effort we applied it to Stocco et al.'s (2017) model of the Simon task. The model was chosen because it is freely available (at github.com/UWCCDL/PSS Simon) and is the same task used by Boksem et al (2006) to study motivation. The Simon task requires participants to respond to visual stimuli by pressing a leftward button to one shape (e.g., a circle) and a right button to another (e.g., a square). Congruent trials are where the stimulus is displayed on the same side as the rule dictates, while incongruent trials are on the opposite side. This paradigm was widely used in neuropsychological studies to assess the ability to inhibit cognitive interference that occurs when the processing of a particular visual property hinders the simultaneous processing of a second stimulus property.

Fig 2. provides a complete overview of the model. It is composed of 4 main steps: 1) Encoding visual stimulus 2) Retrieving a Simon rule 3) Responding and 4) Monitoring performance. The model starts by encoding a cue stimulus, and then it selects which dimension of the Simon stimulus to attend to, color or shape. Followed by stimulus processing, it retrieves the corresponding rule. The attended dimension provides spreading activation that facilitates the retrieval of the associated rule (a feature common to other response interference models in ACT-R: Lovett 2001; van Rijn 2009). The equation below describes the activation of chunks calculated with a base-level learning function ( $B_i$ ), which reflects the recency of previous retrievals, as well as a spreading activation component that reflects the degree to which a chunk matches the contextual components, i.e., the values of every slot *j* in every buffer *k* (Eq 2).

$$A_{i} = B_{i} + \sum_{k \neq j} W_{kj} S_{ji} + \varepsilon \qquad (2)$$

$$External Environment$$

$$Wisual Module$$

$$Wotor Module$$

$$Wot$$

Fig 2. The flowchart of the motivation model in ACT-R

Before a response is made, a "check" production executes a final verification step and, if it finds that the response is incorrect, attempts to re-allocate attention and retrieve another answer. The model contains additional assumptions (about the nature of competing processes in attention) that are not relevant to the goals of this paper, and will not be discussed. What is relevant, instead, is the nature of the "check" process. In the original paper, it was constrained to occur only once. In our extended simulation, we removed this limitation and allowed the model to check as many times as possible. Because the number of checks performed corresponds to how much effort is used to control attention, it provides a natural way to model the cognitive control effort in the task. In addition, we incorporated a motivation value M in the *Goal* chunk and added a self-monitor production that assesses whether the response was correct and, if so, triggers a reward of magnitude M. Like the reward and cost parameter in ACT-R, M is also in time units, representing how many seconds the model is willing to continue working on the task. Note that the model will continue checking only if it finds the current response incorrect. Thus, if M is set high, the model would have more opportunities to correct its response. On the contrary, if M is small, it would have less opportunities to refocus attention.

To increase the task difficulty, Boksem et al., (2006)'s paradigm added cues stimulus, where 80% of the cues were valid. They identified an interesting post-error slowing effect in which participants tended to respond more carefully and slowly after they thought they made a mistake. This process is believed to reflect adjustments in the allocation of mental effort, which is key to the EVC and our motivation framework. Critically, we verified that post-error slowing is *not* produced by Stocco's original model; thus, any success in reproducing this effect must be due to our additional changes.

It was predicted that this model would be able to change strategies based on the probability of gaining rewards and costs. If it never checks, the likelihood of gaining rewards will become low because of many errors. If the model checks a lot, the expected payoff will also decline because of the increasing costs. Therefore, the model should weigh costs and rewards to decide the attempts of checking optimally.

We varied Motivation parameter M, the task difficulty parameter VC (which represents the percentage of cues that are valid cues) as well as the initial cognitive control costs through the *action time* (AT) parameter in ACT-R, which determines the time (and, thus the effort) needed to execute each production The parameter space is shown in the table below.

Parameter	Value	Meaning
M	0.5 - 10	Motivation
VC	0 - 1	Task difficulty
AT	0.01 - 0.1	Cost of control at T <sub>0</sub>

#### Results

To test the validity of our model, we first compared simulated data to Boksem et al.'s (2006)'s empirical data. To focus on Simon effects, we fixed the difficulty parameter VC = 0.5, cost parameter AT = 0.05, and limited motivation parameter to a medium-range (0.1 - 2). Fig. 3 confirmed that our model still reproduced the main Simon effects. Incongruent trials were associated with lower accuracy, and longer response time than congruent trials, same for invalid trials.



**Fig 3.** Model simulation results of Simon effect vs. empirical findings of Simon effects (Boksem et al., 2006). Solid lines denote effects of empirical data and dashed lines denote effects simulated by our model.

In addition, our model could reproduce the post-error slow effect observed in empirical data (Fig 4). Note that this effect could not be reproduced by the original model (Stocco et al., 2017) under any combination of parameters; thus, it is a unique feature of the added motivation mechanism. Specifically, post-error slowing is a consequence of the model adjusting control after a mistake is made.



**Fig 4.** The post-error slowing effects in both empirical data and ACT-R model simulation data (across all parameter combinations). The standard error for both empirical data and model data is shown in the plot.

In the Simon model, the degree of cognitive control is determined by how often the CHECK production is employed before a response is made. Additional firings of the CHECK productions result in repeated allocations of attention and, thus, more time spent before making a response. As hypothesized, we found that a model with a lower M (M < 2.5) would check only once or never check, while a model with a high M (M > 7.5) tends to check more. For example, when M < 2.5, the model performs an average of 0.54 checks, when M < 7.5, it performs on average 1.01 checks.

As predicted by the EVC theory, the relationship between motivation, number of checks, and utility of the CHECK production are complex and nonlinear. To examine this relationship, we fixed the parameter ValidCue% to 0.5. Fig. 5 represents the resulting relationship between the costs, rewards, and allocated control. In the figure, the x-axis represents the intensity of control as the number of firings for the Check production, and the y-axis represents rewards, costs, and utilities in time units. The cost curve (red line) is represented by the total response time the model takes as a function of the count of checking. Moreover, in our self-monitoring process, once the model verifies that the response was correct, a reward equal to M is delivered. The utility of the CHECK production (purple line) represents the expected value of control in the EVC model. In line with the EVC model, our ACT-R model predicts that the model will be encouraged to invest more effort if expecting a higher payoff, but as costs increase, the expected reward decreases and the model decides to stop investing more effort by reserving effort. Note that, although the model *could* achieve greater performance through greater control, it naturally sets to an estimated value of one check per trial because, at this level, the payoff is maximal: additional checks have many diminishing returns. Incidentally, this is precisely the number of checks that were determined to yield optimal results in Stocco et al (2017) and Lovett (2005).



**Fig 5.** Expected value of control in the Simon Task model. Control intensity is expressed as the number of firings of the CHECK production. Note that, even when higher rewards would be possible at a higher level of control, the model naturally shifts to the amount of control that maximizes the difference between rewards and costs.

#### Discussion

In this paper, we have proposed a mechanistic interpretation of motivation within the ACT-R cognitive architecture. Specifically, we propose that motivation can be modeled by assigning a value M to the current model's goal and translating this value as the reward  $R_t$  that is triggered when the goal is accomplished. With this mechanism in place, ACT-R's utility learning mechanism then provides a way to adjust the specific combination of productions that are used to perform a task. Because in ACT-R, rewards and time spent on a task are expressed on the same scale (and rewards are adjusted by the time elapsed), the motivation parameter M can be equivalently expressed as the subjective reward associated with accomplishing the goal and the maximum amount of time that the model is willing to spend on the task. We first demonstrated, using a simple abstract

model, that this mechanism is equivalent to the EVC theory. We then showed how this mechanism can be easily implemented in an existing model of a common laboratory task (the Simon task) and used to account for experimental effects that would otherwise go unmodeled, such as post-error slowing, the effect of difficulty, and even fatigue. All of these effects can be understood as ways in which the model flexibly copes with changes in task demands.

A number of limitations must be acknowledged. First, the level control intensity is quantified by the counts of checking attempts, as a discrete variable. Future work will be needed to address these issues and expanding our model to represent the control intensity with a continuous variable could be the next step of research. Moreover, individual variability in motivation could be examined in future modeling work, specifically how motivation affects the response time rather than accuracy for individuals putting different priorities in speed vs. accuracy tradeoffs (Boksem et al., 2006).

These limitations notwithstanding, we believe that our results are noteworthy for three reasons. In addition to providing a way to implement motivation into ACT-R, our framework provides a more complete view of the role of the Goal module in ACT-R. Currently, the model's capabilities make it distinguishable from the Imaging module. By connecting it to the amount of reward that is generated, this framework provides an interpretation that is more in line with neuroscientific data. It also provides a connection to the original interpretation of the goal in previous versions of ACT-R, as well as the original production selection mechanisms. Finally, it provides a way to better fit models at the individual levels, decoupling the effects of individual capacity and motivation.

#### References

- Adcock, R. A., Thangavel, A., Whitfield-Gabrieli, S., Knutson, B., & Gabrieli, J. D. E. (2006). Reward-motivated learning: Mesolimbic activation precedes memory formation. *Neuron*, 50(3), 507–517.
- Anderson, J. R., & Lebiere, C. J. (1998). The atomic components of thought. *Psychology Press*.
- Anderson, J. R. (2007). How can the human mind occur in the physical universe? New York, NY: Oxford University Press. ISBN 0-19-532425-0.
- Boksem, M. A., Meijman, T. F., & Lorist, M. M. (2006). Mental fatigue, motivation and action monitoring. *Biological psychology*, 72(2), 123-132.
- Botvinick, M., Nystrom, L. E., Fissell, K., Carter, C. S., & Cohen, J. D. (1999). Conflict monitoring versus selection-for-action in anterior cingulate cortex. *Nature*, 402(6758), 179-181.
- Halverson, T., Myers, C., Gearhart, J., Linakis, M. W., & Gunzelmann, G. (2021). Physio-cognitive modeling: explaining the effects of caffeine on fatigue. Paper presented at Virtual MathPsych/ICCM 2021. Via mathpsych.org/presentation/619.

- Herlambang, M. B., Cnossen, F., & Taatgen, N. A. (2021). The effects of intrinsic motivation on mental fatigue. *PloS* one, 16(1), e0243754.
- Holroyd, C. B., Nieuwenhuis, S., Yeung, N., Nystrom, L., Mars, R. B., Coles, M. G., & Cohen, J. D. (2004). Dorsal anterior cingulate cortex shows fMRI response to internal and external error signals. *Nature neuroscience*, 7(5), 497-498.
- Inzlicht, M., Shenhav, A., & Olivola, C. Y. (2018). The effort paradox: Effort is both costly and valued. *Trends in Cognitive Sciences*, 22(4), 337–349.
- Kotseruba, I., & Tsotsos, J. K. (2020). 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*, *53*(1), 17-94.
- Kouneiher, F., Charron, S., & Koechlin, E. (2009). Motivation and cognitive control in the human prefrontal cortex. *Nature Neuroscience*, *12*(7), 939-945.Musslick, S., Shenhav, A., Botvinick, M. M., & Cohen, J. D. (2015). A computational model of control allocation based on the expected value of control. *In The 2nd multidisciplinary conference on reinforcement learning and decision making*.
- Nagashima, K., Morita, J., & Takeuchi, Y. (2020). Modeling intrinsic motivation in ACT-R: Focusing on the relation between pattern matching and intellectual curiosity. *Proceedings of the 18th International Conference on Cognitive Modelling*
- Niv, Y., Daw, N., & Dayan, P. (2005). How fast to work: Response vigor, motivation and tonic dopamine. *Advances in neural information processing systems, 18.*
- Shenhav, A., Botvinick, M. M., & Cohen, J. D. (2013). The expected value of control: An integrative theory of anterior cingulate cortex function. *Neuron*, 79(2), 217–240.
- Shenhav, A., Musslick, S., Lieder, F., Kool, W., Griffiths, T. L., Cohen, J. D., & Botvinick, M. M. (2017). Toward a rational and mechanistic account of mental effort. *Annual Review of Neuroscience*, 40(1), 99–124.
- 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.
- Stocco, A., Murray, N. L., Yamasaki, B. L., Renno, T. J., Nguyen, J., & Prat, C. S. (2017). Individual differences in the Simon effect are underpinned by differences in the competitive dynamics in the basal ganglia: An experimental verification and a computational model. *Cognition*, 164, 31-45.
- 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*
- van Vugt, M. K., van der Velde, M., & ESM-MERGE Investigators. (2018). How does rumination impact cognition? A first mechanistic model. *Topics in Cognitive*

*Science,10* (1), https://doi.org/10.1111/tops.12318

Vroom, V. H. (1964). Work and motivation. *New York: Wiley*.

175-191.

Yeung, N., Botvinick, M. M., & Cohen, J. D. (2004). The neural basis of error detection: conflict monitoring and the error-related negativity. *Psychological review*, 111(4), 931.