Title

Reinforcement Learning: A Computational Framework of Cognition

Author

Rmus, Milena

Publication Date

2024

Peer reviewed|Thesis/dissertation

Reinforcement Learning: A Computational Framework of Cognition

by

Milena Rmus

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Psychology

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Anne G. E. Collins, Chair
Professor Steven Piantadosi
Professor Silvia Bunge

Spring 2024

Reinforcement Learning: A Computational Framework of Cognition

Abstract

Reinforcement Learning: A Computational Framework of Cognition

by

Milena Rmus

Doctor of Philosophy in Psychology

University of California, Berkeley

Professor Anne G. E. Collins, Chair

The thesis investigates applications and extensions of reinforcement learning (RL) algorithms to modeling human cognition, and focuses on development of new tools for fitting cognitive models to behavioral data. The first part of the thesis examines the effect of choice abstraction on recruitment of RL mechanisms. This work challenges the basic RL assumption that action space is always finite and defined, and tests the variability in processes that best describe the data when the appropriate choice features are ambiguous (e.g. abstract). Results indicate that when choices of multiple levels of abstraction are plausible, less abstract choices (e.g. simple motor actions) interfere with more abstract choices (e.g. goal selection). Further cognitive modeling and experimental tests showed that working memory (WM) contribution to more abstract choice process was reduced relative to that of RL, potentially due to the use of WM resources for defining the appropriate choice features in the abstract condition. Second project explored the effect of subgoals, the intermediate learning milestones, on learning in the context of hierarchical reinforcement learning (HRL) framework. In this project we operationalized subgoals in a way that removes the features commonly associated with subgoals (novelty, reward associations, frequency) and sought to test whether subgoals contribute to learning hierarchically organized policies and generalization through a pseudoreinforcing effect independent of these features. The results revealed that participants solved the hierarchical task, with data patterns implying the effect of subgoals on behavior; generalization tests showed that generalization of subgoals, under the constraint of our subgoal definition, was possible but predicated on explicit recognition of subgoal features. The third project focused on development of new cognitive model-fitting tool leveraging artificial neural networks (ANN). The results demonstrating ANN efficacy in fitting parameters and identifying models with tractable and intractable likelihoods, with comparable (or better) performance relative to standard methods where standard methods were applicable.

To Roscoe

I dedicate this work to my cat, family, and steadfast companion, Roscoe, whose paw guided me through to the finish line.

Contents

Acknowledgments

I extend my gratitude to my co-authors and collaborators: Amy Zou, Liyu Xia, Ti-Fen Pan, Maria Eckstein, and Anne Collins. Amy Zou's invaluable contributions played a pivotal role in advancing and completing the RLWM-choice project, making her a steadfast collaborator upon whom I could consistently rely. Jimmy Xia, who initiated the artificial neural networks project, generously shared his wisdom during my early years in graduate school, for which I am profoundly grateful. Maria Eckstein's insights into hierarchical learning significantly enriched the subgoal and hierarchy project, from which I gleaned extensive knowledge. Ti-Fen Pan was as an exceptional collaborator, imparting not only profound insights into neural networks but also swiftly propelling project progress upon joining the lab. Anne Collins, my advisor, has consistently provided outstanding scientific counsel, valuable feedback and guidance. I would also like to thank my committee members, Steve Piantadosi and Silvia Bunge, for their invaluable feedback and commitment to ensuring timely research progress. The members of CCN lab (present and past) have also been crucial - including Jing-Jing Li, Daniel Ehrlich, Sarah Oh, Gaia Molinaro, Aspen Yu and Beth Baribault. Their constructive feedback, engaging discussions, and fun lunch/slack chats have made them not only valuable collaborators but also delightful and supportive colleagues (and excellent boba tea trip companions).

I express heartfelt gratitude to my feline companion, Roscoe, whose unwavering support has been a comforting presence during the most challenging moments of graduate school. Additionally, I extend thanks to my family, who, even if not physically present, have provided remote support and encouragement. I am immensely grateful to a multitude of individuals who played important roles in my graduate journey, but I want to highlight the most important ones. To Jennifer, my cherished LOTR fellow and a wellspring of support and life wisdom, your companionship on many Tolkien quests means the world to me - there is truly no one I would have rather done them with. To Sara, Djordje, and Aleksandar — my companions since high school — I am grateful and honored for having known you throughout the years. Your enduring presence has shaped the person I am today (the good parts at least), and for that, I am profoundly thankful.

Chapter 1

Introduction

1.1 Computational cognitive models of reinforcement learning

Reinforcement learning (RL) models occupy a significant role in cognitive science research. The basic premise of RL models is that the learning process can be characterized as a trial-and-error interaction between the learner and the environment (Sutton and Barto, 1990, 2018; Wagner and Rescorla, 1972). In other words, the learner occupies a state in the environment, enacts choices and receives outcome from the environment; the learner then encodes certain actions as more rewarding (at specific states) than others. This simple premise is easily formalized using a set of model equations that can be applied to model the behavioral data from learning experiments, designed to test various properties of learning and decision-making. Model parameters embedded in equations are then used to quantify various features of cognitive processes, such as the rate of learning, decision noise, etc. As such, RL models have been useful for understanding individual variability in learning/decision-making in clinical populations (Adams et al., 2016; Huys et al., 2016; Maia and Frank, 2011; Montague et al., 2012), different developmental groups (Eckstein, Master, Dahl, et al., 2022; Nussenbaum and Hartley, 2019; Palminteri et al., 2016), as well as for shedding light on cognitive mechanisms behind common behavioral patterns (e.g. habitual vs goal-directed behavior: Collins and Cockburn, n.d.; Daw et al., 2011; Decker et al., 2016).

Another reason for the importance of RL models is that it lies at an intersection of computer science (Kaelbling et al., 1996; Qiang and Zhongli, 2011), neuroscience (Dayan and Daw, 2008; Niv, 2019; Schultz et al., 1997) and cognition (Collins and Frank, 2012; Daw et al., 2011; Frank and Badre, 2012). Indeed, direct mapping of RL equations to neural mechanisms, such as the reward prediction error signaling of dopaminergic neurons (Schultz et al., 1997), grants credibility to RL models through linking neural signaling to cognitive mechanisms and behavioral output. In addition many domains in computer science have leveraged RL principles to design complex algorithms in computer vision (object recognition

examples), robotics (training agents) and AI (Abbeel et al., 2006; Konidaris and Barto, 2007; Le et al., 2022; Mnih et al., 2013).

A general property of computational cognitive models that pertains to their capability to define how cognitive mechanisms relate to one another with high precision applies to RL models as well. Therefore, many researchers leverage this specificity to characterize and dissect different components of reward-based learning by applying RL models to data from cognitive experiments, with parameters characterizing different aspects of how agents process information (Eckstein, Master, Xia, et al., 2022; K. Miller et al., 2024), including but not limited to decision noise, rate of learning (from positive and negative feedback), choice perseveration, degree of random lapses.

While RL has indeed been an important framework used to examine learning and decision making, it by no means provides a complete explanation of the underlying cognitive mechanisms. For instance, there are learning patterns that deviate significantly from RL predictions. One example of that is immediate learning based on single exposure, otherwise known as one-shot learning (which is frequently observed in humans). Basic RL alone does not account for such learning behaviors, as learning under RL assumptions is essentially incremental. Modifying RL algorithms to include equations that represent contributions of alternative learning mechanisms (such as various forms of memory) can in part address this issue by enabling immediate storage of information. For instance, previous work showcased that integrating RL models with models of working memory (WM) that assume immediate, but temporary and capacity-limited information storage captures fast learning, especially when the amount of information is within the limits of WM capacity (Collins, 2018; Collins and Frank, 2012; Collins et al., 2014). Similarly, other examples of work have shown that hybrid RL models equipped with episodic memory (Bornstein et al., 2017) and attention (Radulescu et al., 2019) mechanisms provide a better, more robust accounts of learning behavior. The results from cognitive modeling indicate that integrating reinforcement learning (RL) with other learning mechanisms, like memory and attention, often leads to a more comprehensive theory of cognition compared to using RL in isolation. This integration aligns with findings from neural data. Specifically, the basal ganglia, a brain structure known for its role in RL computations (Joel et al., 2002), has strong connections with the prefrontal cortex and the hippocampus, which are involved in working memory, attention and episodic memory functions (Atallah et al., 2004; Hazy et al., 2007; Packard and Knowlton, 2002; Zhao et al., 2018). These connections explain the observed functional association between the basal ganglia and the areas responsible for memory and attention, supporting the idea that a combined approach of RL with memory and attention mechanisms provides a more accurate model of cognitive processes.

Furthermore, RL algorithms have restrictive assumptions. For instance, the key ingredients of baseline RL computations are clearly defined and finite state and action spaces (Rmus et al., 2021). This is not always the case, as the state/action spaces may be continuous, not fully known or highly dimensional. Indeed, basic RL algorithms have been criticized for not scaling up to high-dimensional environments, since the increase in dimensionality of

state and action spaces renders RL computations exponentially inefficient (Botvinick, 2008; McGovern and Barto, 2001; Stolle and Precup, 2002; Xia and Collins, 2021). In addition to this, decisions do not always map onto action for execution, and may involve more complex goal-action contingencies - a variability that is commonly not reflected in RL equations. Furthermore, RL assumes that an outcome is observed after each action, allowing the agents to evaluate their actions immediately as either rewarding or not rewarding. Rewards are, however, sometimes sparsely observed (e.g. with a time delay or after multiple actions have been executed). This set of assumptions indeed limit the range of problems RL can be applied to as a candidate cognitive model.

The following sections provide more detailed background on previous research addressing the mentioned limitations of RL, and introduce the projects in thesis chapters designed to contribute to this ongoing research expanding RL applications to instances which challenge its basic premises.

1.2 Effect of choice abstraction on reinforcement learning and working memory contributions

RL interacts with other cognitive processes that contribute to learning

While a family of RL algorithms offers simple and precise mechanistic descriptions of learning and decision making, it does not provide a full picture. Specifically, in addition to trial-and-error learning humans sometimes require less interaction with the environment, and instead perform immediate learning - through leveraging retrieval of previously acquired useful information (e.g. from episodic memory Bornstein et al., 2017; Gershman and Daw, 2017), or by narrowing down the task space by assigning higher weight to the features of environment most relevant to the task (Niv et al., 2015; Radulescu et al., 2016, 2019). Indeed, the patterns of human behavior (pertaining to learning and decision-making) that depart from RL predictions have been documented in previous work (Bornstein et al., 2017; Collins and Frank, 2012; Collins et al., 2014; Radulescu et al., 2019). One area of this research focuses on the role of cognitive functions referred to by the umbrella term of executive functions (Bunge, 2024). The unifying idea behind this work is that RL does not operate in isolation, and instead interacts with other cognitive processes, such as attention and working memory. For instance, humans might rely on working memory that supports fast, one-shot learning when the amount of information required to be learned is constrained, and arbitrate towards RL when the information exceeds their working memory capacity (Collins, 2018; Collins et al., 2017). Similarly, attention might serve to isolate most relevant features of the task, thus constraining it to the size RL can efficiently compute over (Radulescu et al., 2016, 2019). Models of how these cognitive systems interact are supported by their

neural/biological underpinnings. Specifically, the executive functions have been consistently linked to prefrontal cortex (Badre and Nee, 2018; Friedman and Robbins, 2022; Gilbert and Burgess, 2008; Koechlin and Summerfield, 2007), which has strong projections to the striatal/basal ganglia system (Hazy et al., 2007; Middleton and Strick, 2000; O'Reilly and Frank, 2006) frequently associated with RL functions. For instance, fMRI results have shown that increased PFC activation linked to recruitment of WM resources suppresses the reward prediction error (RPE) in striatum in instances where the information load is small enough to be learned using working memory (Collins et al., 2017).

Working memory contribution to RL in defining the choice space through credit assignment

Defining a choice space is a component of RL computations that may be impacted by the contribution of working memory. Specifically, choice space refers to the set of actions available to the agent for construction of policies, which define how agent operates in environment with a goal of maximizing rewarding outcomes. Choice spaces are sometimes simple (e.g. simple motor actions), but sometimes they are more complex, consisting of multiple features (e.g. position or labels of different options that can be selected via executing different motor actions), where there may be ambiguity as to which one is relevant. For instance, if there are two yogurt cups, a pink and a white one, on left and right side respectfully and pink yogurt is more tasty , should a learning agent credit the good flavor to the color or to the simple act of reaching out to the left side? The effect the variability in abstraction level of choice space may have on RL (and how WM may contribute to this effect) has scarcely been examined in modeling applications. Previous research (Luk and Wallis, 2013) suggests that less (motor action) and more abstract (goal stimulus) choices are encoded by orbitofrontal cortex and anterior cingulate cortex - implying different neural mechanisms recruited for different choice types.

Chapter 2 of the thesis focuses on exploring how working memory contributes to RL in conditions where choice space is either concrete, or more abstract/flexible - contributing to the ongoing work aiming to construct a more complete picture of how different cognitive systems interact, especially in instances that challenge the basic premises of RL. The goal of this project was to examine whether 1) when two different choice types of different levels of abstraction are available these choice processes impact one another other, and 2) what contribution of WM is to learning and the choice process in the instances where there is amibguity as to what defines a correct choice dimension.

1.3 Hierarchy in reinforcement learning

Leveraging hierarchy to explain how learning unfolds efficiently

Basic reinforcement learning has frequently been criticized for not scaling well to large action/state spaces. Under assumptions of RL, agents learn rewarding policies by visiting states in the state space, enacting different actions and encoding the rewarding state-action associations based on the feedback. However, this rapidly becomes infeasible as the number of potential states/actions increases - quickly resulting in a combinatorial explosion of possible state/action associations for the agent to learn. Indeed, outside of simple laboratory experiments, humans learn increasingly more complex policies that simply cannot be captured with basic RL models. As a response to this limitation, researchers have leveraged the concept of hierarchy in learning (Botvinick et al., 2009; Stolle and Precup, 2002; Sutton et al., 1999), which has been an efficient concept for explaining how humans organize information efficiently in a way that permits robust learning (Botvinick, 2008; Lashley et al., 1951; G. A. Miller et al., 2017). Hierarchical Reinforcement Learning (HRL) framework is an extension of basic RL that proposes the primitive actions can be chunked into temporally extended policies, which can be further recombined into more complex policies that are added to an action repertoire an agent can exploit upon encountering novel problems (Solway et al., 2014; Tomov et al., 2021). For instance, knowledge of steps required to prepare ingredients for breakfast can be successfully transferred to the task of preparing lunch. Formally, the temporally-extended policies referred to as options (Konidaris and Barto, 2007; Stolle and Precup, 2002; Xia and Collins, 2021) are initially acquired through trial-and-error exploration that subsequently leads to chunking of simple actions into complex sequences that can more robustly be applied in service of the task.

In addition to providing an account of how people might learn complex solutions to problems, the options/HRL framework offers an important insight into another one of the critical limitations of classic RL - lack of explanation of how learning occurs in environments with sparse rewards. Specifically, RL assumes presence of feedback at each time-step, but fails to account for environments in which feedback is not observed until much later after the action execution. In HRL framework, problems can be decomposed into sets of subgoals - intermediate milestones that can support learning until the final outcome is reached (Diuk et al., 2013; Ribas-Fernandes et al., 2019; Solway et al., 2014). For instance, reaching a hallway on the way from one room to the next can be considered a milestone. Different research lines have focused on different properties of subgoals. Because subgoals are not rewarding in a way primary rewards are (e.g. food, money) (Diuk et al., 2013; Ribas-Fernandes et al., 2019; Sutton et al., 1999), some researchers have conceptualized subgoals as novelty signals that drive curiosity and/or novelty, making it more likely the agent will encode information observed thus far as meaningful (Baldassarre and Mirolli, 2013; Chentanez et al., 2004; Singh et al., 2010). Some researchers have focused on the structural aspect of the state spaces, defining subgoals as the states that are most frequently visited en route to terminal reward

(Diuk et al., 2013; McGovern and Barto, 2001), or states with highest degree of connections with other states (Şimşek and Barto, 2008). However, there is not enough work that looks at specifically whether subgoals affect learning through means independent of external effects such as surprise, structural factors or reliable association with rewarding outcomes.

Subgoals in hierarchical reinforcement learning

Subgoals also serve the function of enabling generalization of policies between different tasks (e.g. reaching a hallway is relevant for both changing rooms and exiting the building), and as such are an important ingredient of robust, generalizable behavior frequently observed in humans (a feature artificial agents frequently fall short of).

Chapter 3 outlines a project in which we designed a hierarchical learning task, where individual actions are combined into simple policies that in turn make up a more complex, final policy. Execution of simple policies was marked by an appearance of a potential subgoal signaling a rewarding sequence. Importantly, we ensured that 1) the subgoals were not entirely predictive of rewards (e.g. could occur in non-rewarded sequences) thus ensuring they don't inherit value from rewards, 2) the subgoals do not occur with higher frequency relative to non-subgoals, 3) subgoals may be observed on each trial and thus are not more surprising. These manipulations permitted us to test the isolated pseudoreinforcing effects of subgoals on learning, independent of associations with external value, intrinsic or structural factors.

1.4 Broadening cognitive modeling methods

Cognitive models, including reinforcement learning, occupy a critical role in computational cognitive science, because they provide a simple way to translate cognitive theories into relatively simple algorithms that can be related to behavioral data and used to 1) quantify different aspects of cognition (such as rate of learning, forgetfulness, etc.), and 2) arbitrate between which theories provide better accounts of observed behavior (Lee and Webb, 2005; Montague et al., 2012; Shultz, 2003). However, the extent to which cognitive models may be useful is determined by the existing statistical tools that enable researchers to relate them to the data.

To fit computational models to the data, we commonly compute the likelihood of participants' behavior (e.g. choices) (D) under the model specification (M), where likelihood is defined in accordance with the Bayes rule:

$$P(M|D) = \frac{P(D|M) \cdot P(M)}{P(D)}$$

Most traditional model-fitting methods (such as Maximum Likelihood Estimation, MLE; Maximum A-posterior Estimation, MAP; Hierarchical Bayesian Modeling) rely on computing

and maximizing likelihood, for the purpose of estimating model parameters and quantifying model fitness (Cousineau and Helie, 2013; Myung, 2003; Rigoux et al., 2014). In other words, in parameter estimation, many model-fitting optimization tools search for parameter values that optimize the likelihood of the data under the given model; in model-identification, likelihood serves as an ingredient for calculating model fitness - how well the model captures the data patterns (e.g. Akaike Information Criterion, AIC (Akaike, 1998; Bayesian Information Criterion, BIC (Schwarz, 1978).

Despite the prevalence of likelihood-based model-fitting methods there is a large body of computational models for which computing the likelihood is not tractable. In other words, computing likelihood for the full data sequence, even for a single participant, is not feasible. For example, models with strong sequential dependencies (e.g. observation on each trial is dependent on all the preceding trials) which assume latent variables that affect behavior on each trial but are not observed in the data (e.g. an unobserved rule or an attention state that governs behavior, Frank and Badre, 2012; Solway et al., 2014) require full marginalization over all the latent variable possibilities across the entire trial history. This quickly becomes intractable beyond the first handful of trials. For instance, if the model assumes presence of a latent state which is not observed in the actual data, computing the likelihood requires the following:

$$\begin{aligned}\mathcal{L}(\theta) &= \sum_{t=1}^{T} \log \mathbb{P}(a_t | h_t, \overline{h}_{t-1}, \theta) \\ &= \sum_{t=1}^{T} \log \Big(\sum_{l} \mathbb{P}(a_t | h_t, ls_t = l; \theta) \mathbb{P}(ls_t = l, \overline{h}_{t-1}; \theta) \Big)\end{aligned}$$

where ls = latent state, l $\in$ { set of possible latent states }, $\overline{h}_{t-1}$ corresponds to the history of all trial observations up to the trial t. Computing likelihood requires summing over latent states in the equation (e.g. all possible latent state and respective trial history trajectories) which is in practice impossible beyond the first few trials.

To ensure that cognitive researchers can test a broader range of theories, including the ones best formalized by models with intractable likelihood, there have been different approaches that aim to substitute traditional model fitting methods in instances where using them is not possible. One group of alternative methods focuses on likelihood approximation (such as inverse binomial sampling, van Opheusden et al., 2020; assumed density estimation, Minka, 2013; particle filtering, Djuric et al., 2003), aiming to approximate rather than compute the exact likelihood. One of the most frequently used approximation methods in cognitive science is Approximate Bayesian Computation (ABC, Palestro et al., 2018; Sunnåker et al., 2013; Turner et al., 2013). ABC involves reducing dimensionality of data sequences, often consisting of hundreds of observations, into summary statistics (e.g. average accuracy, error rates, learning curves). Next, parameter values are sampled to simulate the data

from the specified model, followed by computing the summary statistics. The parameters that generate simulated summary statistics closest to the summary statistics of real data (based on the rejection process) are then estimated as the best parameters. While the ABC does provide a workaround solution it suffers critical limitations - including that insufficient summary statistics (frequently chosen by the researcher) can result in significantly incorrect parameter estimates, and unstable application to models with sequential dependencies.

More recently, there has been a lot of effort dedicated to leveraging the power and flexibility of neural networks for estimating parameters of cognitive models (Boelts et al., 2022; Fengler et al., 2021; Radev, Voss, et al., 2020; Radev et al., 2021; Schmitt et al., 2021). Many of these methods actually build on ABC computations by using the neural networks to automate summary statistics selection process, with neural network architecture designed to compute and optimize summary statistics (Radev, Mertens, et al., 2020; Radev, Voss, et al., 2020). In general, these approaches are examples of simulation-based inference, where large amounts of data are simulated as a training set used to train the neural network to estimate parameter values based on the simulated data sequences. The trained network can then be applied to estimate parameters of the data set it has not observed yet (e.g. human data from experiments). Some of the network approaches, however, are not easily applied to models with strong sequential dependencies (Fengler et al., 2021; van Opheusden et al., 2020).

Chapter 4 offers a discussion of a method that uses artificial neural network approach to estimate parameters and perform model identification using models with sequential dependencies and intractable likelihood. We evaluated this approach by running a comparison to the set of standard, likelihood-based model-fitting methods (MLE, MAP) applied to models with tractable likelihood, and approximation method (ABC) applied to models with intractable likelihood. We evaluated the performance of our method in application to parameter recovery and model identification. Furthermore, we extended the baseline approach to afford uncertainty quantification of parameter estimates. This project contributes to the larger body of work focused on improvement of alternative methods that will enable testing a less restricted range of cognitive theories.

1.5 Aim of the thesis

The aim of this thesis was to formulate projects that examine application of reinforcement learning algorithms (and their extensions) to instances in which basic RL assumptions are challenged, including 1) the task environment with ambiguous choice definition, and 2) the task environment with sparse rewards and temporally extended policies. Specifically, the first project deployed a combination of the experimental design and modeling to probe recruitment of potentially different learning mechanisms for less/more abstract choices in experiment 1; experiment 2 builds on the results of experiment 1 by including working memory manipulation in the task, and hybrid working memory-reinforcement learning model.

The second project aimed to examine the effect of controlled/defined features of subgoals to learning and generalization in a hierarchical setting. Lastly, in project 3 we aimed to develop an alternative model-fitting tool based on artificial neural network implementation that bypasses the likelihood estimation/approximation drawbacks commonly present in standard likelihood-based methods (e.g. inability to fit cognitive models with intractable likelihood).

The key results of the 3 thesis projects can be summarized as follows:

- Project 1: Choice types that are less/more abstract recruit RL mechanisms differently. Specifically, experiment 1 showed that when both types of choices are possible, the less abstract choices (such as motor actions) interfere with more abstract choices (goal selection). This was shown both in behavioral analyses in model comparison favoring the model with policy mixture parameter, responsible for capturing the interference, present only in abstract choice condition. Experiment 2 outlined working memory contribution to reinforcement learning in different choice type conditions: working memory weight, parameter that quantifies the extent to which the working memory contributes to the choice selection relative to RL, was lower in more abstract condition relative to less abstract condition. We reasoned that when action space is abstract, WM resources (e.g. capacity) may be leveraged to define the choice space more concretely and in a way that can be effectively used by a reinforcement learning system - resulting in reduced WM contribution to the actual choice process in abstract choice condition. Furthermore, the policy mixture parameter which captured interference patterns was not specific to either WM or RL.

- Project 2: Participants who passed the screening procedure (based on evidence of task engagement) were able to solve the hierarchical task, and have shown sensitivity to its hierarchical structure (e.g. based on the patterned response times consistent with chunking of simpler sequences into more complex ones). Furthermore, we found that subgoals can exert a pseudo-reinforcing effect on learning independently from their ability to predict rewards, novelty/surprise or frequency. Generalization of subgoals, however, was limited to only a subset of the participants; these participants were able to explicitly identify subgoal features, and have shown bias for subgoals over non-subgoals in a test of preference. These results imply that while the subgoal effect on learning may be isolated from that of novelty, frequency and reward prediction the generalization of subgoals is predicated on explicit recognition of subgoal features.

- Project 3: Artificial neural networks (ANNs) can be leveraged to fit a broad range of cognitive models, including the ones with intractable likelihood. We simulated a set of likelihood-tractable cognitive models from different families (e.g. reinforcement learning and Bayesian inference) on two tasks representative of tasks frequently ran in cognitive experiments. We evaluated our method by comparing its parameter estimation accuracy to that of the standard model likelihood-based fitting methods, and found that in most cases it performed just as well, if not better, compared to the best

case of standard methods. We also evaluated our approach based on the model identification, and found that our ANN method significantly outperformed the standard methods. We also simulated likelihood-intractable models, and evaluated the ANN in comparison to a standard likelihood approximation method commonly applied to fit cognitive models with intractable likelihoods. The ANN performed significantly better, and demonstrated a robust performance in model identification. We also extended the baseline ANN to include evidential learning in order to enable quantification of uncertainty around parameter estimates. In addition, we performed a set of robustness checks (e.g. based on model misspecification where the network is trained to estimate parameters of one model but tested on recovery of parameters simulated from a different model, missing trials, different parameter range) and we found that in some cases our method was quite robust to misspecification, presenting some loss of accuracy which was not catastrophic (e.g. in case of misspecified nested models). In misspecification instances in which our network was impacted (e.g. different classes of models), it was not any more impacted than the standard methods. Our results in total show that our approach represents a contribution to the growing body of work on the application of simulation based inference to computational cognitive models.

Chapter 2

Choice Type Impacts Human Reinforcement Learning

(Previously published: Rmus, M., Zou, A., & Collins, A. G. (2023). Choice Type Impacts Human Reinforcement Learning. Journal of Cognitive Neuroscience, 35(2), 314-330.)

2.1 Abstract

In reinforcement learning (RL) experiments, participants learn to make rewarding choices in response to different stimuli; RL models use outcomes to estimate stimulus–response values that change incrementally. RL models consider any response type indiscriminately, ranging from more concretely defined motor choices (pressing a key with the index finger), to more general choices that can be executed in a number of ways (selecting dinner at the restaurant). However, does the learning process vary as a function of the choice type? In Experiment 1, we show that it does: Participants were slower and less accurate in learning correct choices of a general format compared with learning more concrete motor actions. Using computational modeling, we show that two mechanisms contribute to this. First, there was evidence of irrelevant credit assignment: The values of motor actions interfered with the values of other choice dimensions, resulting in more incorrect choices when the correct response was not defined by a single motor action; second, information integration for relevant general choices was slower. In Experiment 2, we replicated and further extended the findings from Experiment 1 by showing that slowed learning was attributable to weaker working memory use, rather than slowed RL. In both experiments, we ruled out the explanation that the difference in performance between two condition types was driven by difficulty/different levels of complexity. We conclude that defining a more abstract choice space used by multiple learning systems for credit assignment recruits executive resources, limiting how much such processes then contribute to fast learning.

2.2 Introduction

The ability to learn rewarding choices from non-rewarding ones lies at the core of successful goal-directed behavior. However, what counts as a choice? When a child tries a pink yogurt in the left cup and a white yogurt in the right cup, and then prefers the right cup, what choice should they credit this rewarding outcome to? In their next decision, should they repeat their previously rewarding reach to the yogurt on the right, independently of its color, or should they figure out where the white yogurt is before reaching for it? Selecting the type of yogurt is a more abstract choice: It requires subsequently paying attention to the other dimension (Where is the white yogurt?) and applying the appropriate motor program to execute the choice. Thus, making the more abstract choice additionally involves less abstract choices, but in this case, only the abstract choice should be credited for the yogurt's tastiness. Knowing the relevant dimension of choice to assign credit to is essential when learning. How does choice type impact how we learn?

The theoretical framework of reinforcement learning (RL) is highly successful for studying reward-based learning and credit assignment (Sutton and Barto, 2018). However, RL as a computational model of cognition typically assumes a given action space defined by the modeler, which provides the relevant dimensions of the choice space (i.e., either the yogurt color or the cup position)—there is no ambiguity in what choices are (i.e., color such as pink/white, or side such as left/right), and the nature of the choice space does not matter (Rmus et al., 2021). As such, RL experiments in psychology tend to not consider the type of choices (a single motor action such as pressing a key with the index finger; Collins et al., 2017; Tai et al., 2012), or the more general selection of a goal stimulus that is not tied to a specific motor action (Daw et al., 2011; Foerde and Shohamy, 2011; Frank et al., 2007) as important, and researchers use the same models and generalize findings across choice types. Recent research has shed some light on how participants might identify relevant dimensions of the state and choice space (Farashahi et al., 2017; Niv, 2019); however, this research does not address how learning occurs when the learner knows the relevant choice space but multiple dimensions of choice are nonetheless available, such as in our yogurt example.

Examining learning of responses when multiple-choice dimensions may be relevant is important, however, as most of our choices in everyday life are ambiguous: Did I pick the white yogurt or the one of the left? In some cases, these dimensions are hierarchically interdependent: Choices can be represented at multiple levels of abstraction (e.g., have breakfast; have yogurt; have pink yogurt; have the yogurt on the right; reach for the yogurt on the right side). In such cases, a choice along a relevant dimension (yogurt color) requires a subsequent choice on a reward-irrelevant dimension (position/motor action), which then needs to be considered for the choice's execution, but not credited during learning. By contrast, in other cases, some choice dimensions may neither be relevant for learning nor for executing the choice—for example, the child should learn to fully ignore the color of the plate that the yogurt is on for both their choice and their credit assignment.

Different types of choices may recruit different cognitive/neural mechanisms (Rescorla

and Solomon, 1967). For example, previous animal models of decisionmaking suggest that the orbitofrontal cortex and the anterior cingulate cortex index choice outcomes for goal stimulus choices and motor action choices, respectively (Luk and Wallis, 2013). Ventral striatum lesions in monkeys impaired learning to choose between rewarding stimuli, but not between rewarding motor actions (Rothenhoefer et al., 2017). In humans, recent behavioral evidence suggests that the credit assignment process is what differentiates learning more relevant choice dimensions from less relevant (here motor) ones (McDougle et al., 2016), and that there might be a hierarchical gradation of choices in terms of credit assignment. In particular, while people are capable of learning the value of both abstract rule choices and concrete action choices in parallel (Ballard et al., 2018; Eckstein et al., 2019), they also seem to assign credit to more concrete actions by default when making abstract choices that need to be realized through motor actions (Shahar et al., 2019). The brain relies on multiple neurocognitive systems for decision-making, but whether choice format impacts learning similarly across systems remains unexplored. Specifically, although RL models provide a useful formalism of learning, they do not easily relate to underlying processes. Indeed, RL models are known to summarize multiple processes that jointly contribute to learning (Eckstein et al., 2021), such as the brain's RL mechanism, but also episodic memory (Bornstein and Daw, 2013; Bornstein et al., 2017; Poldrack et al., 2001; Vikbladh et al., 2019; Wimmer and Shohamy, 2012), or executive functions (EFs) (Collins and Frank, 2012; Rmus et al., 2021). Here, we focus on working memory (WM), which has also been shown to contribute to learning alongside RL (Collins, 2018; Collins and Frank, 2012; Collins et al., 2017). If choice type matters for learning, does it matter equally for each cognitive system that contributes to learning, or differently so?

In summary, there is a twofold gap in our understanding of how choice format impacts learning. First, when multiple-choice dimensions are available but only one is relevant, does the type of the relevant choice dimension impact learning, and if so, through what computational mechanisms? We consider, in particular, the important case where one relevant choice dimension needs to be executed through a second, irrelevant choice dimension (a motor action), and how this contrasts to learning when one dimension is fully irrelevant to both choice and learning. Second, are the differences rooted in the brain's RL system, WM, or both? To address this gap, we designed a task that directly compares learning to make choices along two orthogonal dimensions, with different levels of generality or interdependence, when there is no ambiguity about which choices are relevant to the learning problem.

In our task, one choice dimension is a spatial position that directly maps onto a consistent motor action, and the other is a more general choice dimension, conceptualized as the selection of stimulus goals that constrain a downstream selection of an overall irrelevant spatial position and corresponding motor action. In a second experiment, we manipulated learning load to separately identify WM and RL contributions to learning, and investigated with computational modeling how choice matters in both systems. Our results across two experiments suggest that choice type strongly impacted learning, resulting in slower learning when the relevant choice dimension was more general and required execution along another

dimension. This was in part driven by an incorrect, asymmetric credit assignment to less general choices when they were irrelevant. Furthermore, WM (rather than RL) mechanisms seemed to drive the deficits in performance in the more general choice format condition, indicating that defining a more general action space, shared by multiple choice systems, recruited limited executive resources. In both experiments, we ruled out the simple explanation that the performance difference was driven by an effect of difficulty by 1) implementing experimental controls that minimize this concern and 2) ruling out predictions of a pure difficulty effect in analyses and modeling.

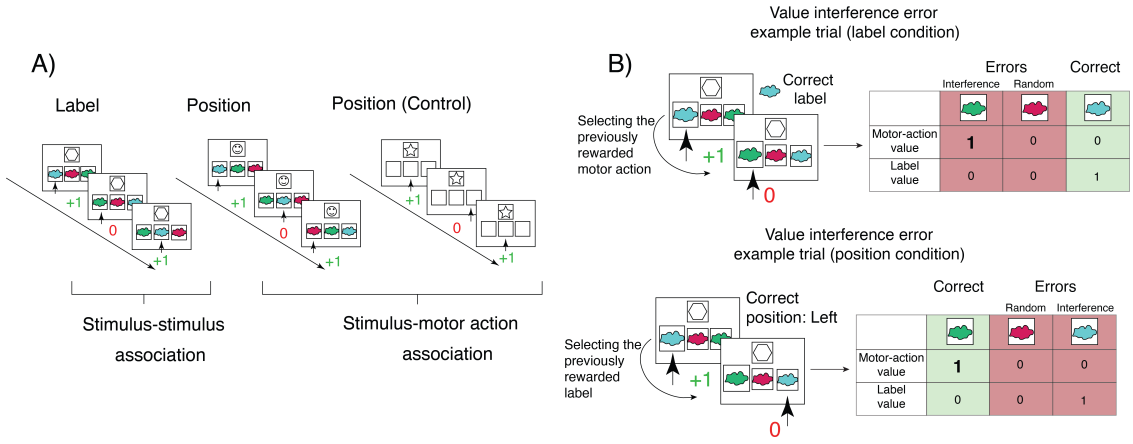


Figure 2.1: Experiment design. (A) Participants played a card-sorting game with three different conditions: label (learning which box color is correct for each card – more general choice), position (learning which motor action/position is correct for each card – less general choice), control (identical sorting rules as position condition, but without labeled boxes). (B) We assumed that participants track card-dependent reward history for both positions and labels, and that both of these contribute to the choice selection process, sometimes resulting in interference errors. Note that the card-dependent reward history is cumulative (tracked across all past trials during which the given card was presented, rather than only one-trial back), but for simplicity of illustration, we only show 1-back trial in (B).

2.3 Results

Experiment 1: Behavioral Results.

We first asked whether participants learned differently across experimental conditions. Learning curves show that participants learned well in all conditions, as their accuracy increased with more exposure to each card (Fig. 2.2A). A repeated-measures one-way ANOVA confirmed that there was a main effect of Condition (label/position/control) on performance,

$F(2, 61) = 97.7$, $p < .001$, $\eta^2 = .62$. We next tested which specific conditions contributed to this significant difference and found a marginal difference between control and position conditions; however, this difference did not reach statistical significance (paired t test: $t(61) = 1.61$, $p = .11$, Cohen's $d = 0.20$). This result suggests that the additional choice feature (the labels) in the position condition did not have a strong impact on the choice process. Performance in the label condition, however, was significantly lower than that in the position and the control conditions (paired t test: position: $t(61) = 11.1$, $p < .001$, Cohen's $d = 1.42$; control: $t(61) = 12.9$, $p < .001$, Cohen's $d = 1.65$). We next examined why label condition performance was worse. We hypothesized that choice was not simply noisier in the label condition, but instead that choice might be contaminated by the reward history of irrelevant motor choices. To test this hypothesis, we computed the cumulative card-dependent label/position reward history (see Methods section) and quantified the proportion of error trials in which participants incorrectly chose a box with high reward history of an incorrect feature (Fig. 2.1). In the position condition, participants did not make more interference errors than expected at chance level (for two possible errors; Fig. 2.2B; $t(61) = 0.13$, $p = .89$, Cohen's $d = 0.01$). This confirms that the presence of labels in the position condition did not impact choice compared with the control condition. By contrast, in the label condition, the proportion of interference errors was significantly higher than chance (Fig. 2.2B; $t(61) = 2.54$, $p = .01$, Cohen's $d = 0.32$). Furthermore, the proportion of interference errors in the label condition was significantly greater than interference errors in the position condition, $t(61) = 2.13$, $p = .03$, Cohen's $d = 0.27$. This result suggests an asymmetry in interference between different choice spaces, in that the values of less general/motor action choices seem to contaminate the more general choice process (but not the other way around). To rule out the possibility that the effect we observed was driven by the block/condition order (i.e., transfer of incorrect strategy from the previous block), we ran a mixed-effects general linear model predicting accuracy with previous versus current block conditions. The result of this analysis showed that participants' performance was affected by the current block condition ($p < .001$), but not the previous block condition ($p = .45$), thus ruling out order effects as a possible explanation of our results. In addition, our results were replicated in the second experiment (as reported later), where we removed the control condition altogether, and counterbalanced the remaining condition blocks such that participants could either experience position or label condition blocks first. This further supports the conclusion that the observed results are unlikely to be explained by the order effects.

Next, we performed a trial-by-trial analysis to examine the effect of card/label values on correct trials' RTs. For each condition, we used a mixed-effects linear model to predict log(RT) from the RHD between chosen and unchosen choices (see Methods section), where choice referred to label in one predictor and position in the other. The rationale behind this analysis is that, if participants are engaging in the appropriate decision strategy, then RTs should decrease with the higher RHD in the conditionrelevant dimension (label or position), because a higher RHD means greater evidence in favor of the correct response. On the other hand, in the event of interference, we expected participants' RTs to be modulated by the

RHD of the incorrect dimension (e.g., position RHD in label condition). We controlled for the trial number in the model.

As predicted, in models for each condition (position condition model $F^2 = .27$; label condition model $F^2 = .154$), participants' RTs decreased with increased respective RHD (Fig. 2.2C; label condition: $\beta_{label} = -0.04$, $p < .001$, position condition: $\beta_{position} = -0.06$, $p < .001$). Label RHD did not affect the RTs in the position condition ($\beta_{label} = -0.004$, $p < .055$). Hence, the mixed-effects model aligned with interference errors, confirming that participants' choices were not affected by the presence of an additional feature (the labels) in the position condition. On the other hand, the position RHD surprisingly increased RTs in the label condition ($\beta_{position} = -0.034$, $p < .001$), suggesting that the interference of motor action values with label values may have resulted in the delay of choices (Fig. 2.2C). We compared the subject-level β estimates of the effect of incorrect dimension RHD on RTs in position and label conditions, and found that the incorrect RHD effect was significantly greater in the label condition (paired t test: $t(61) = 3.87$, $p < .001$, Cohen's $d = 0.49$), confirming the asymmetry between conditions that was revealed in previous analyses.

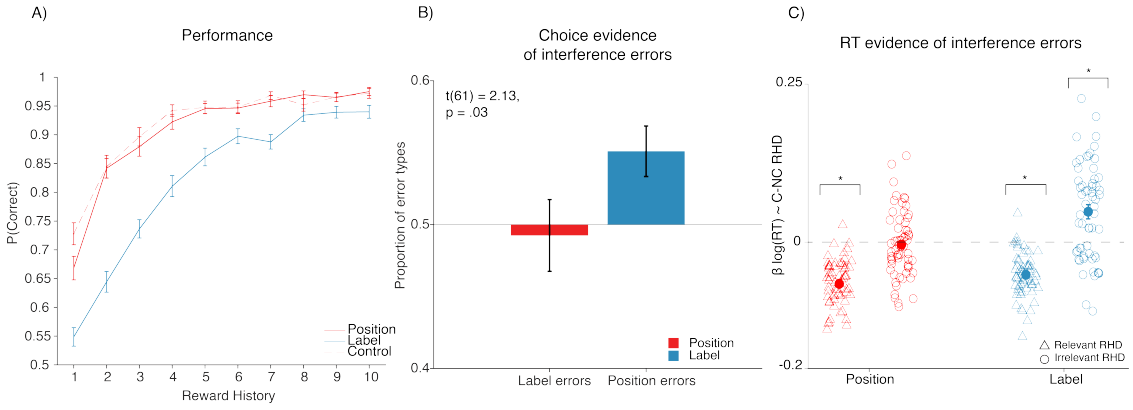
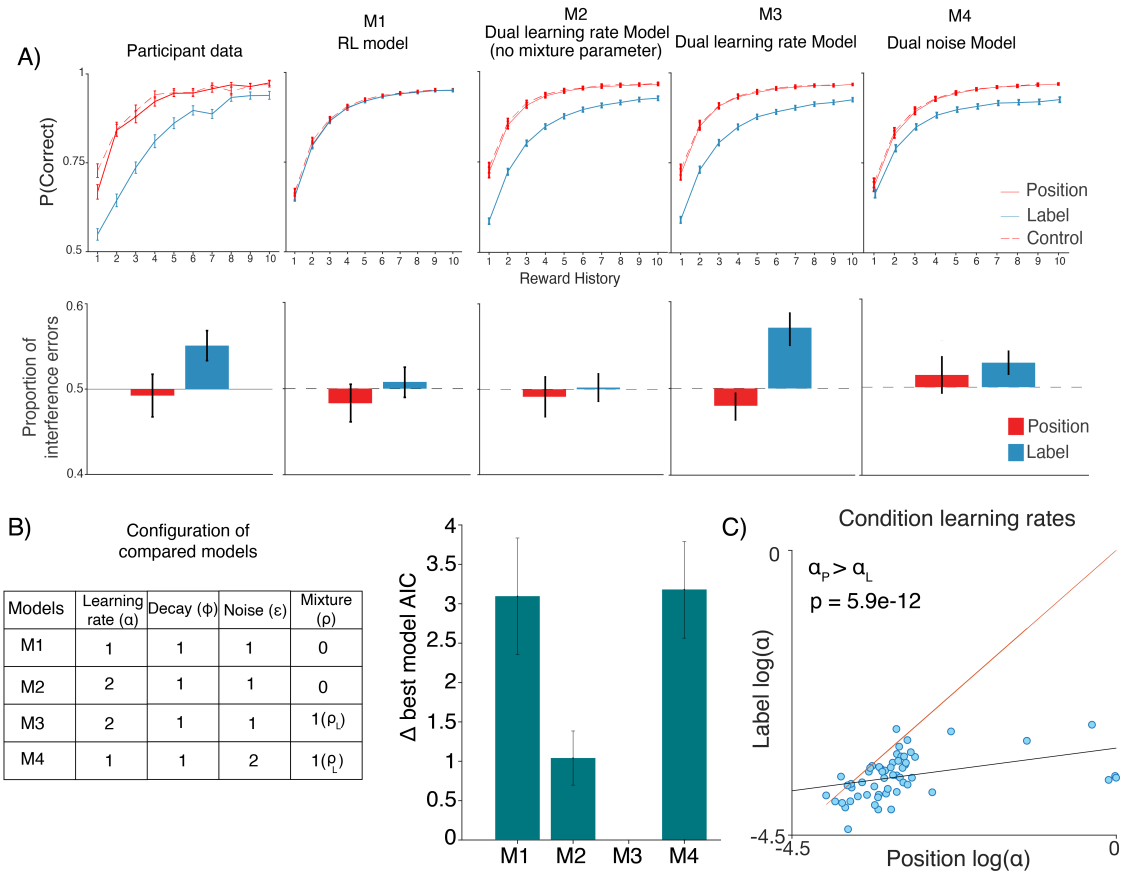


Figure 2.2: Experiment 1 model-independent results. (A) Proportion of correct choices as a function of number of previous rewards obtained for a given stimulus. Participants performed worse in the label condition, compared with the position and control conditions. Performance in the position and control conditions did not differ statistically. (B) Asymmetric value interference: The values of motor actions interfered with values of correct labels in the label condition, thus resulting in the interference errors, but not the other way around. (C) Mixed-effects regression model shows that the interference of motor action reward history/values may have resulted in the longer RTs in the label condition. *Indicates statistical significance at $p < .05$.

Experiment1: Modeling results.

We used computational modeling to tease apart the mechanisms driving condition effects. We fit several variants of RL models and focus here on four models that represent the main different theoretical predictions (Fig. 2.3A,B). The standard RL model (M1) assumes no difference between the conditions and serves as a baseline that cannot capture the empirical effect of condition. RL model M2 lets learning rates depend on condition and tests the prediction that slower learning with labels is driven by different rates of reward integration. Model M3 extends model M2 with an additional mechanism, parameterized by the value mixture (ρ_L), that enables the position value to influence policy in the label condition. Ruling out the difficulty explanation using computational modeling. Model M4, the dual-noise model, is an RL model with a condition-dependent noise parameter (ϵ). M4 captures the hypothesis that the label condition is more difficult, resulting in a noisier choice process. Models M1–4 all assume $\rho_P = 1$, with no influence of labels in position blocks. Other models considered separate decay (ϕ) parameters and a free position condition ρ_P , but did not improve fit.

Model M3 offered the best quantitative fit to the data, as measured by AIC (Fig. 2.3B). Furthermore, only model M3 was able to qualitatively reproduce patterns of behavior. Specifically, for each of the models, we simulated synthetic data sets with fit parameters and tested whether the model predictions matched the empirical results. We focused on two key data features in our model validation: performance averaged over the stimulus iterations (learning curves) and asymmetrical interference errors. Model validation showed that only the model with two learning rates and one ρ parameter (M3) captured both properties of the data (Fig. 2.3 A). These results confirm that the learned value of (irrelevant) motor actions influenced the selection of more general label choices. Furthermore, model comparison results show that slower learning in the label condition was not because of a noisier choice process, but because of a reduced learning rate. Indeed, the position condition was significantly greater than the label condition α (sign test; $z = 6.35$, $p < .001$, effect size: .81; Fig. 2.3C). Interestingly, the learning rates in the two conditions were correlated (Spearman $\rho = .39$, $p = .003$; Fig. 2.3C), suggesting that the learning process in the two conditions was driven by related underlying mechanisms.



Models	Learning rate (α)	Decay (ϕ)	Noise (ε)	Mixture (ρ)
M1	1	1	1	0
M2	2	1	1	0
M3	2	1	1	$1(\rho_L)$
M4	1	1	2	$1(\rho_L)$

Figure 2.3: Experiment 1: Modeling results. (A) Model validation comparing the observed data to predictions of tested models; M3 reproduces behavior best. (B) Parameters used in models M1–4 (left); M3 has best group-average AIC. (C) Comparison of condition-dependent learning rates shows that learning rates are correlated, and that label condition learning rates are significantly lower compared with position condition learning rates.

Experiment 2: Behavioral Results.

The results of the first experiment suggest that the choice type affects learning. However, given the experimental design, our conclusions could not dissociate whether the difference in RL parameters actually reflected a difference in RL mechanisms or in WM mechanisms. Recent work (Collins, 2018; Collins and Frank, 2018), nevertheless, suggest that RL behavior recruits other learning systems, such as WM. Hence, the variations that may appear to be driven by RL mechanisms might conceal what is actually a WM effect. To address the question of whether the choice definition matters for learning at the level of RL or WM,

and whether slowed learning stems from slowed WM or RL, we ran a second experiment. In Experiment 2, we varied the number of cards (set size) to manipulate WM involvement. Furthermore, we fit variants of the RL WM model to test the contribution of WM mechanisms.

Experiment 2 results replicated findings from Experiment 1, showing that there was a main effect of Condition (Fig. 2.4A; repeated-measures one-way ANOVA, $F(1, 56) = 98.95$, $p < .001$, $\eta^2 = .63$). Furthermore, we replicated the pattern of interference errors, suggesting that the value of position choices interferes with that of label choices, but not the other way around (Fig. 2.4B; $t(55) = 2.89$, $p = .006$, Cohen's $d = 0.38$)

We next investigated how set size manipulation affected these results. As predicted, performance decreased with set size in both conditions, position: $F(3, 56) = 11.83$, $p < .001$, $\eta^2 = .38$; label: $F(3, 56) = 23.498$, $p < .001$, $\eta^2 = .55$. There was an interaction between set size and condition, $F(3, 56) = 16.21$, $p < .001$, $\eta^2 = .46$ (Fig. 2.4A). There was a marginal set size effect in interference errors that did not reach significance, $F(3, 56) = 2.17$, $p = .09$, $\eta^2 = .20$ (Fig. 2.4C).

To better understand the source of the set size effect, we ran a general linear mixed-effects model to predict trial-by-trial performance. Our mixed-effects model included predictors indexing WM mechanisms (set size and delay between presentations of the current stimulus and the most recently rewarded stimulus; indexing capacity and susceptibility to decay properties of WM, respectively) and RL effects (dimension relevant, card-dependent reward history, calculated from the cumulative number of earned points for each card, indexing reward-based learning). We also ran a model that tests for an interaction between individual RL/WM factors and the task condition.

A likelihood ratio test provided evidence in favor of the interaction model over a model without interactions (model without interactions $f^2 = .42$; model with interactions $f^2 = .43$; LR $p < .05$). The interaction model showed that, as expected, participants' performance increased as a function of reward history ($\beta = .62$, $p < .001$), and decreased as a function of set size ($\beta = -0.18$, $p = .00011$). There was no effect of Block ($\beta = .04$, $p = .58$) or Delay ($\beta = -0.04$, $p = .37$), suggesting that neither overall task exposure nor delay affected performance over and above reward history and set size. The only significant interaction term was the Condition $\times$ Reward History interaction ($\beta = .16$, $p = .01$), suggesting that the reward history more heavily contributed to an increase in performance in the label condition. To understand our results on a more mechanistic level, we turned to computational modeling.

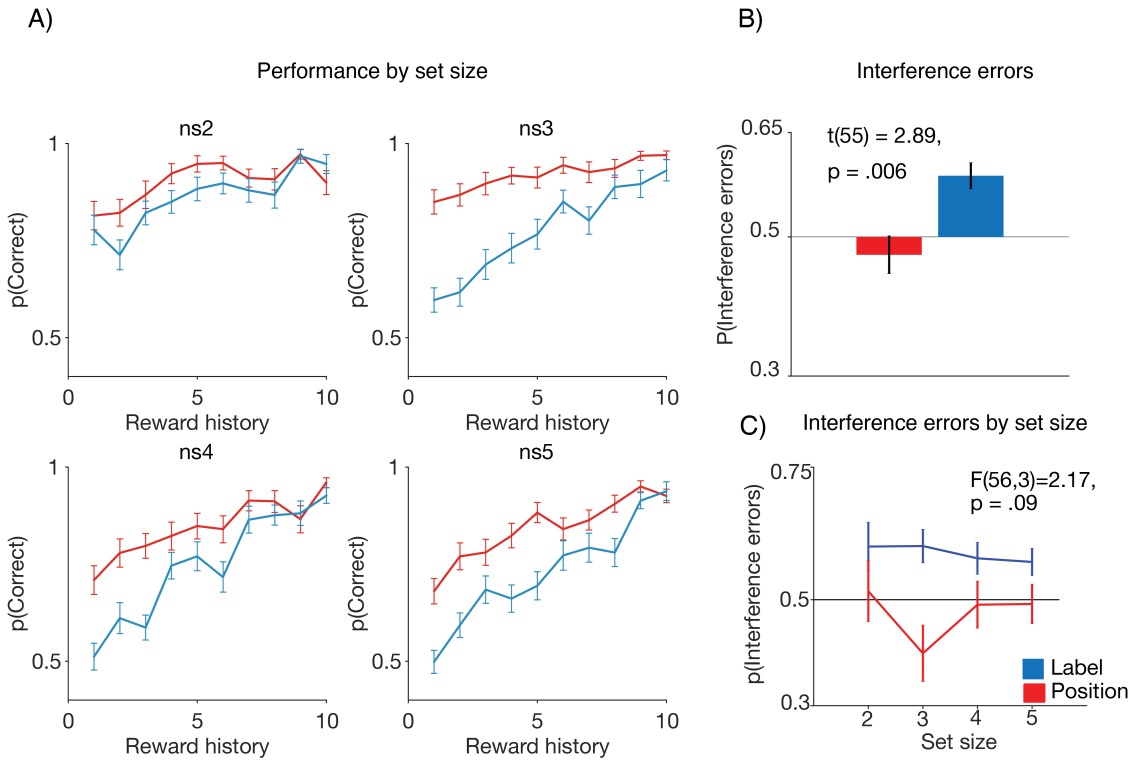


Figure 2.4: Experiment 2 results. (A) Participants' overall performance varied by set size (a marker of WM contribution) and was worse in the label condition. (B) The asymmetry in value interference replicated from Experiment 1, showing that values of position choices interfere with values of label choices, but not the opposite. (C) The interference errors did not vary by set size.

Experiment 2: Modeling Results.

The set size manipulation in Experiment 2 enables us to identify distinct contributions of RL and WM (Collins and Frank, 2012) with the full RL-WM model (see Methods section). Briefly, RL-WM disentangles an incremental, value-learning process (RL), as well as a rapid-learning, but decay-sensitive, short-term, memory-based decision process (WM). Choice policy is a weighted mixture of RL and WM (Fig. 2.6A,B), where the weighting is proportional to one's WM capacity. In other words, the model architecture posits that if one's WM capacity is low, one might be more likely to rely on RL than WM, especially when set size (number of items) is high. We first replicated in Experiment 2 that models including only one of those mechanisms could not adequately capture the set size effect, as has been shown before (Collins and Frank, 2012). We then approached model comparison by systematically varying the complexity of the RL-WM model (Fig. 2.6A), to establish

whether specificity in RL or WM module parameters (or both) is necessary to capture the divergence between behavioral patterns in the two conditions. Because the RL-WM model assumes the policy for choice generation at the level of both RL and WM, we also tested if integrating irrelevant dimension interference with a mixture parameter in the policy of RL module or WM module (or both) could best capture our data. We were interested in the condition-based dissociation between parameters. Exploring all possible parameter combinations was computationally prohibitive. Thus, we explored a subset of the most relevant models (see Methods section; in the main text, we focus only on a subset of models). Using AIC comparison, we identified the simplest model that allowed us to capture the properties of the data (M1, Fig. 2.3A). In M1, the WM weight (ω) and ρ parameters were condition-dependent (with free ρ parameter for label condition, and position condition ρ fixed to 1). Capacity (K), learning rate (α), decay (ϕ), LB, and noise (ϵ) were shared across the two conditions—model comparison showed no benefits to making them independent (Fig. 2.9).

We further consider three other variants of this model: no value interference ρ (M2), ρ in RL policy alone (M3), and ρ in WM policy alone (M4; 2.5A). Last, we consider a control model with condition-dependent ϵ and α, which would primarily attribute the decline in label condition performance to noise/RL system (M5).

Consistent with Experiment 1 results, the AIC comparison revealed that M5 could not capture data well, and that M1 without ρ (M2) fit worse (Fig. 2.5A), providing additional evidence for the necessity of the interference mechanism to capture choice data and, thus, the existence of motor value interference in label blocks. However, the AIC comparison failed to significantly distinguish between the remaining models M1 (ρ in RLWM), M3 (ρ in RL), and M4 (ρ in WM; repeated-measures ANOVA: $F(2, 56) = 2.63$, $p = .07$, $\eta^2 = .08$), although in RL, models fit numerically worse, supporting the idea that we needed to include motor value interference in the WM module to account for the results. Therefore, we henceforth focus on the simplest model, M1 with condition-dependent ω and ρ in RL and WM policy, as this model makes the fewest specific assumptions about RL-WM dissociation between the two conditions. Note that model comparison results were identical (and stronger) when using Bayesian Information Criterion instead of AIC, and that protected exceedance probability supported M1 over other models. The M1 model adequately captured the data patterns in (1) learning curves (Fig. 2.5B), (2) overall interference errors (Fig. 2.5C), and (3) interference errors by set size (Fig. 2.5D). Furthermore, the WM weight ω was significantly reduced in the label condition compared with the position condition in M1 (Fig. 2.5E). Overall, the results suggested that the performance decrease in the label condition was driven primarily by deficits in WM, specifically by a smaller WM weight that indexes the set-size-independent contribution of WM to learning. Therefore, the choice type (more/less general) impacted learning, and it seemed to do so by decreasing participants' ability to use WM for learning. However, the value interference appeared to be present in both RL and WM mechanisms.

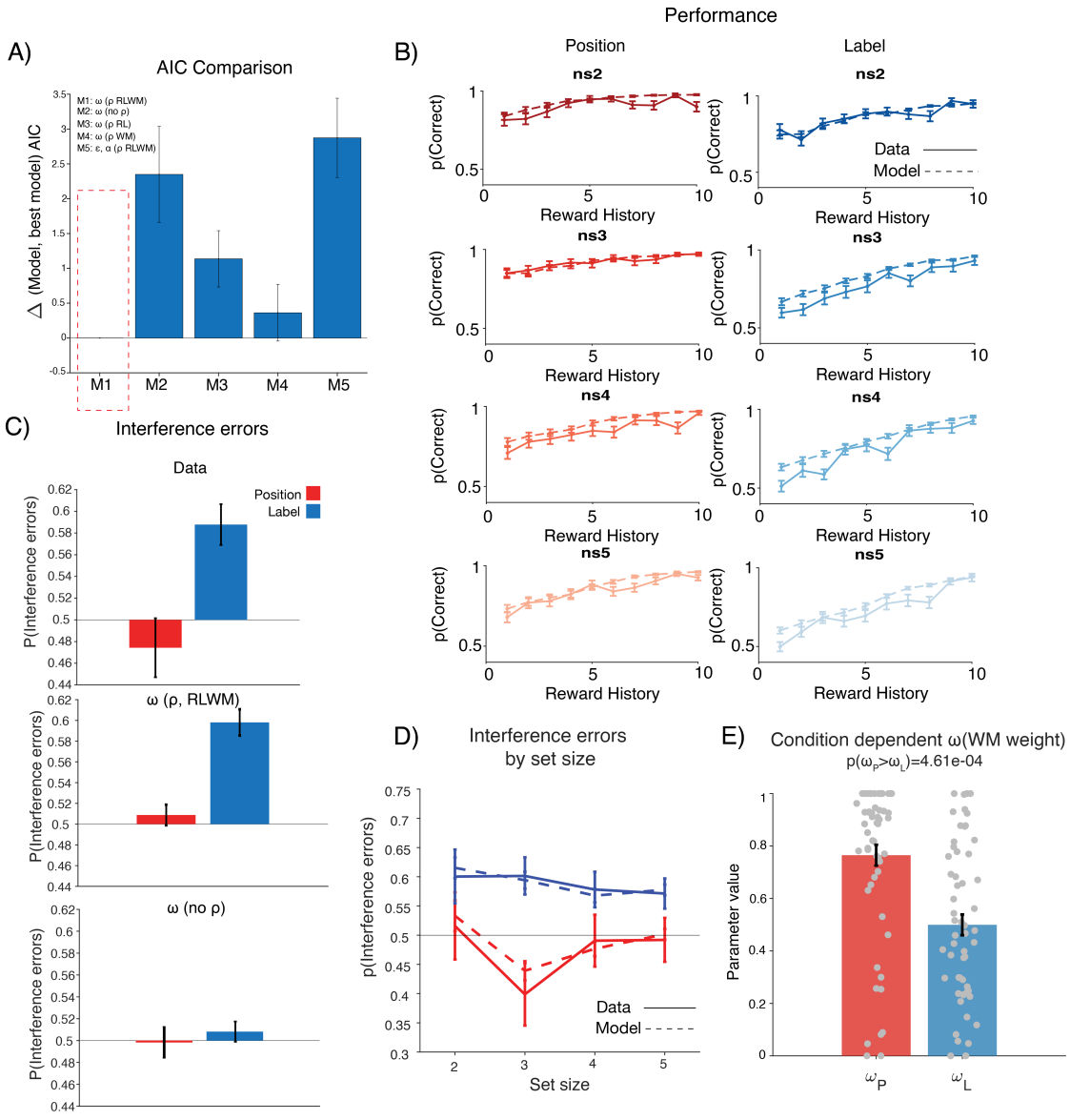


Figure 2.5: (A) AIC comparison allowed us to narrow down the space of models. Models with condition-specific WM weight (ω) fit the best (M1–M4). Removing the mixture parameter (ρ) harmed the model fit (M2). A model assuming impairment in RL did not fit as well (M5). See main text for model specifications. (B) Model simulations of the best model M1 captured the behavioral data patterns. (C) Model validation for M1 (ρ) and M2 (no ρ) confirms the necessity of ρ parameter in capturing the interference error patterns. (D) M1 captured interference errors in different set sizes. We note that the numerical dip in set size 3 is not statistically significant. While it is unclear why the model simulations reproduce it, it is possible that it arises from a pattern in the stimulus sequences, which is used by participants and model simulations. (E) Comparison of condition-dependent parameters shows that ω is lower in the label condition.

2.4 Discussion

Humans and animals make many types of choices, at multiple levels of generality, where some choices are dependent on others. We designed a new experimental protocol to investigate whether and how different choice types impact learning. Across two experiments, behavioral analyses and computational modeling confirmed our prediction that the generality of choice type impacts learning, with worse performance for choices that do not map onto a simple motor action. Computational modeling revealed two separable sources of impairment. First, value learning for relevant choices of a more general type was slower, as revealed by smaller learning rates (α) in Experiment 1. Second, choices were contaminated by irrelevant motor action values. Experiment 2 examined whether this dissociation originated in different neurocognitive systems' contributions to learning, namely, RL and/or WM. Our results revealed that the reduction in learning speed for general-format choices stemmed more from WM than the RL process, with WM weight (ω) reduced but RL (α) unchanged, when controlling for WM contributions. However, the interference of low level values appeared to be present in both mechanisms. The selective reduction in WM weight implies that participants' executive resources might be leveraged to define the choice space that is then used by both the RL and WM system; a more generalized choice space requires a higher degree of such computation, thus leaving reduced resources for actual learning.

In both experiments, we found an asymmetry in interference between choice types. When participants learned to make more general choices (selecting a label) that required a subsequent motor action (pressing the key corresponding to the label's location), their choices were influenced by the irrelevant reward history of motor actions. By contrast, when participants learned to make less general choices (the correct response is defined by pressing the same key corresponding to the box location), they were not influenced by the irrelevant reward history of box labels. This result is consistent with a choice hierarchy interpretation, where participants may be unable to turn off credit assignment to irrelevant choice dimensions when the realization of their (abstract) choice does involve this dimension (Eckstein and Collins, 2020), but are able to do so when the irrelevant choice dimensions are more abstract, as shown here.

Although our results imply that participants exhibit a decision bias toward motor actions, we acknowledge that our protocol cannot disambiguate between the motor actions themselves and the corresponding spatial location of the boxes. That is, we cannot confirm whether the participants track the value of specific motor actions (index/middle/ring finger key press) or of the corresponding box positions (left/middle/right). Hence, a competing interpretation of our results would be that spatial positions, rather than motor actions, are prioritized in tracking value, compared with other visual features such as labels. To completely rule out this possibility, we would need to modify the current task with a condition where the motor actions are not aligned with the specific positions, and inspect whether the interference effect persists in such a condition. However, we think this account is less likely than a choice abstraction account, which explains our results more parsimoniously, without requiring a

"special status" for a "position" visual feature.

Furthermore, animal research supports this interpretation, as it shows differences in the neural code of choices, which are defined primarily as motor actions versus more abstract choices (Luk and Wallis, 2013; Rothenhoefer et al., 2017). Specifically, these studies have utilized recordings from neurons of animals trained to perform a task that contrasted motor action choices with stimulus goal choices, to identify the neural substrates that differentiate between the two. The results seem to implicate pFC, ACC, OFC, and striatal regions (ventral striatum) as areas that differentiate between how choices with different levels of abstraction are coded in the brain. Therefore, it is likely that it truly is dissociation between motor actions, rather than positions, and more abstract choices that led to the interference and the effects we observed in our work. Our results have implications for research on hierarchical representations. Specifically, although simple RL algorithms are useful to capture reward-based learning, they are commonly criticized because they fail to capture the flexibility and richness of human learning. Hierarchical reinforcement learning was developed in part to address limitations of standard RL (Botvinick et al., 2009; Collins and Frank, 2013; Stolle and Precup, 2002; Xia and Collins, 2021). Previous research suggests that the choice space might be hierarchically represented, with the lower level of hierarchy consisting of primitive actions, and the higher level consisting of temporally extended actions (state-dependent, extended policies), also known as options (Stolle and Precup, 2002). Evidence from this research suggests that hierarchical representations are useful for enabling transfer; instead of learning from scratch in the novel context, an agent can leverage higher-level representations to speed up learning (Xia and Collins, 2021). The transfer results also suggest that choices at different levels of hierarchy show an asymmetry in flexibility in novel contexts (lower level choices being less flexible). Our results are consistent with this finding because motor actions seem less flexible and less impacted by competing reward information, providing additional supporting claims for hierarchical representations in choice space.

In addition to this, there is evidence of hierarchical representations at the neural level. In particular, frontal areas (primarily pFC) and BG are also frequently investigated as neural mechanisms that support hierarchical reasoning/learning (Collins and Frank, 2013). Converging insights suggest that the cortico-BG loops support representations of both low-level associations and abstract rules/task sets, giving rise to latent representations that can be used to accelerate learning in novel settings (Collins and Frank, 2013; Eckstein et al., 2019; Stolle and Precup, 2002; Xia and Collins, 2021).

Both experiments implicated overall slowed learning, in addition to value interference, in the worse performance for more general choices. Our first experiment (which allowed us to test RL models only) implicated the learning rate (usually interpreted as a marker of the RL system; Eckstein et al., 2019) as the mechanism driving the difference between conditions with different choice types. However, our second experiment enabled us to test the more holistic hybrid model of RL and WM, and revealed that the impairment in the more general choice condition likely stemmed from the WM system, rather than RL. Previous work has shown that EF, in its different forms (i.e., WM, attention), contributes to RL computations

(Collins, 2018; Niv, 2019). The general summary of this work is that high-dimensional environments/tasks pose difficulty to RL; EF then acts as an information compressor, making the information processing more efficient for RL (Rmus et al., 2021). Operating in a more generalized choice space might more heavily rely on the contribution of EF (in this case WM) relative to operating in the less abstract condition. Therefore, resource-limited WM might be leveraged to define the choice space (i.e., relevant features of the choice space, like labels in label condition). As a result, the WM weight included in the WM + RL hybrid model, which indexes the WM contribution to learning, appears to be reduced in the label condition. Our interpretation of this result is that this reduction in WM contribution may indicate that some of participants' limited WM resources are recruited elsewhere, and specifically that it has already been used to define the choice space over which learning and decision making occurs.

Although we conclude that WM is used for defining the choice space, consistent with prior results on EF contributions to RL computations (Todd et al., 2008), we do not make any particular assumptions about how the use of choice space is divided between RL and WM once it is defined. We tested different model variations, with the parameter mixing label/position values, to explain value interference at the policy level of RL, WM, or both. If there was clear evidence in favor of the mixture parameter in either the RL or WM policy, it would imply that the policy generation based on choice space is primarily driven by that system. However, our model comparison revealed no evidence that the mixture parameter is specific to either RL or WM, suggesting that the choice space is shared between the two. This will be important to further explore in future research.

A competing interpretation for our findings of slowed learning for more abstract choices is that the label condition required more attention and was more difficult. Although this is true, we took steps to mitigate this potential confound on two levels—task design and modeling. In the task design, we constructed the single trial structure such that participants had a chance to see box labels first, before the onset of the card. By doing this, we aimed to eliminate potential advantages of the position condition, where participants do not need to perform an additional process of identifying the label location before executing the response. Furthermore, our modeling enabled us to validate the effects of our task design. Specifically, in both experiments, we tested the model with condition-dependent noise parameters, which predicts that different noise/difficulty levels are what drive the performance difference in our conditions. This model did not fit the data well (Experiment 1: best model AIC $>$ 2 noise model AIC $t(56) = -5.179$, $p = 3.13e-06$, Cohen's $d = 0.69$; Experiment 2: best model AIC $>$ 2 noise model AIC $t(56) = -5.05$, $p = 4.98e-06$, Cohen's $d = 0.67$), making it unlikely that difficulty-induced lack of attention/motivation could explain our condition effect.

A competing interpretation of our results might be that participants simply did not pay attention to the labels in the position condition, accounting for the observed asymmetry. That is, because the labels are not informative for selecting a correct response in the position condition, participants might simply not be attending to them at all, as opposed to encoding

them, with the choice process remaining unaffected by the interfering information from labels. However, we think this competing account is unlikely, for multiple reasons. First, the labels were very salient (colors, and presented before the stimulus); thus, participants would need to actively avoid them to not perceive them. While we have no direct measure of participants' attention to the labels, it is unlikely that they did not process them at all. Second, there is evidence from previous work that participants encode and use information from unattended stimuli, especially when the unattended stimuli might be relevant for the reward structure in the task (Gutnisky et al., 2009; Sasaki et al., 2010). Therefore, the labels (even if not strongly attended to in the position condition) would be a part of the input in the choice process that, according to the results, does not strongly impact the choice of the position, which is consistent with our interpretation. We thus consider the more probable interpretation to be that the participants do perceive and attend to the irrelevant labels, but successfully avoid learning their values. However, future work should investigate more directly how much attention participants pay to irrelevant labels.

Another limitation is that our design did not manipulate the degree of value interference between the choice dimensions, because we equally counterbalanced the position of labels. Instead, introducing a systematic bias such that, in a label block, for example, some positions had higher value because of overlapping with correct labels more frequently, would provide an opportunity to induce and measure different magnitudes of interference. This would be an interesting question to explore in the future.

Surprisingly, we found that participants' RTs on correct trials increased as a function of position RHD in the label condition. This implies that when both label and position sorting rules were in agreement on the best choice to make (i.e., the blue box was the correct box and was in the position that had been most rewarded so far), RTs tend to be longer (the corresponding effect was not observed in the position condition, where label RHD had no effect on RTs). This is, therefore, a counterintuitive effect, as we would expect the congruent information to accelerate response execution, rather than slow it, as observed here. One possibility might be that participants do engage in a form of arbitration between selection of different response types. Specifically, they might be biased to execute the motor action based on the RHD, as it seems to present itself as a default option based on our results. However, because they are informed that the response based on label selection is correct for the given block, they might delay the response execution, to override the default. Nevertheless, this is a speculation—careful modeling of RTs is required to further explain this effect, which is beyond the scope of this article. This account would also predict the highest degree of conflict in this congruent situation, rather than in situations where both rules disagree. It will be an important question to solve in future research. Our results highlight the importance of correct credit assignment and investigation of mechanisms, which might lead to errors in the credit assignment process.

Our results are consistent with the previous research suggesting that motor actions might have a stronger effect on the choice selection process than is usually considered (Shahar et al., 2019). Our modeling approach allowed us to show that the mixture of Q values at the policy

level is what may lead to the interference effect/incorrect credit assignment. However, as of now, we cannot conclusively say whether the mixture happens selectively at the policy level of RL, WM, or both.

Identification of correct rewarding responses is a critical building block of adaptive/goal-directed behavior. Impairments in one's ability to identify the appropriate choice space, which is then used for one's inference process, may consequently result in maladaptive/suboptimal behavioral patterns. Our interference effect results suggest that some aspects of the choice space might be incorrectly overvalued, thus resulting in choice patterns that reflect repeated erroneous selection of incorrect choice types or an inability to utilize flexible stimulus–response mappings. These kinds of perseverative responses are reminiscent of the inability to disengage from certain actions, observed in conditions such as obsessive–compulsive disorder (Rosa-Alcázar et al., 2020). It would be interesting to use our task and computational modeling approach to investigate whether the mixture/interference of values at the policy level could also explain the behavior of such populations.

Conclusion

In conclusion, our findings provide evidence that the choice type and how we define a choice have important implications for the learning process. The behavioral patterns (i.e., value interference from less abstract choices) are consistent with the premises of hierarchy in learning and behavior (i.e., lower levels in hierarchy impacting processing in higher levels), which has become an increasingly promising topic of research (Collins and Frank, 2013; Eckstein and Collins, 2020; Stolle and Precup, 2002). We also demonstrate additional evidence, relevant to the definition of the choice space, that EF (specifically WM) contributes to RL in reward-driven behaviors (Rmus et al., 2021), further demonstrating the complex interplay between various neurocognitive systems.

2.5 Methods

Participants

Experiment 1. Our sample for Experiment 1 consisted of 82 participants (40 women, age mean = 20.5 years, SD = 1.93 years, age range = 18–30 years) recruited from the University of California, Berkeley, Psychology Department's Research Participation Program. We based our sample size on samples from previous similar behavioral experiments (Collins, 2018: 91 participants; Collins et al., 2014: 85 participants; Collins and Frank, 2012: 78 participants). In accordance with the University of California, Berkeley, institutional review board policy, participants provided written informed consent before taking part in the study. They received course credit for their participation. To ensure that the participants included in analyses were engaged with the task, we set up an exclusion criterion of or greater average accuracy

across all task conditions. This cutoff was determined based on an elbow point in the group's overall accuracy in the task (Fig. 2.12). We excluded 20 participants based on this criterion, resulting in a total sample of 62 participants for the reported analyses.

Experiment 2. For the second experiment, we recruited 75 participants(54 women, 1 preferred not to answer; age mean = 20.34 years, SD = 2.4 years, age range = 18–34 years) from the University of California, Berkeley, Research Participation Program. One of the prerequisites for participating in Experiment 2 was that participants had not previously taken part in Experiment 1. We also relied on previous research to decide on the sample size, as in Experiment 1. Participants completed the experiment online (De Leeuw, 2015) and received course credit for their participation. Using the same exclusion criteria as the previous experiment (based on the distribution of average accuracy), we excluded 18 participants, resulting in the total sample of 57 participants.

Experimental Protocol

Experiment 1

Learning blocks. Participants were instructed that they would be playing a card sorting game, and that on each trial, they would sort a card into one of three boxes. Their goal was to use reward feedback to learn which box to sort each card into. The boxes were labeled with three different colors (green, blue, and red), and participants chose one of the boxes by pressing one of three contiguous keyboard keys (corresponding to the box position) with their index, middle, and ring finger. Importantly, the color of the boxes changed positions on different trials (i.e., the blue box could appear on the right side on trial n, and in the middle on trial n + 1). Participants received deterministic feedback after each selection (+1 if they selected the correct box for the current card, 0 otherwise).

Before the experiment, participants read detailed instructions and practiced each task condition. The task then consisted of eight blocks, divided into three conditions. Each of the three conditions was defined by its distinct sorting rule. In the label condition, the correct box for a given card was defined deterministically by the box's color label (2.1A). For instance, if the blue box was the correct choice for a given card, participants were always supposed to select the blue box in response to that card, regardless of which key mapped onto the blue box on a given trial. In the position condition, the correct box was defined deterministically by the box's position (left/middle/right). For example, the correct response of a given card would always be achieved by pressing the leftmost key with the index finger, regardless of the box color occupying the left position (2.1B). The sorting rule in the position control condition was identical to the sorting rule in the position condition, but the boxes were not tagged with color labels. This condition allowed us to assess participants' baseline performance when only one response type (e.g., position, but not the label) was available. Importantly,

participants were explicitly told the sorting rule (position or label) at the beginning of each block, to avoid any performance variability that may arise as a function of rule inference and uncertainty. Following the eight learning blocks, participants performed two additional tasks; these are not the focus of the current article and are not analyzed here. Out of eight blocks in total, two were control condition blocks, three were position conditions, and three were label conditions. Block order was pseudorandomized: Participants completed a control block first and last, whereas the conditions of Blocks 2–7 were randomly chosen within participants, but counterbalanced across participants. In each block, participants learned how to sort six different cards; we used a different set of images to represent cards in each block. The boxes were labeled with the same three colors across all blocks, except the position control blocks, where the boxes were not labeled. Participants experienced 15 repetitions of each card, resulting in 90 trials per block; trial order was pseudorandomized to ensure a uniform distribution of delays between repetitions of the same card in a block. We controlled for the card-dependent position–label combinations across trials. Specifically, each label occurred in each position an equal number of times (i.e., the blue label occurred 5 times on the left, right, and middle box for each card). We also ensured that the position–label combinations were evenly distributed across the task (i.e., the blue–middle combination did not occur only during the first quarter of block trials).

Single trial structure. On each trial, participants first saw the three boxes with their color labels underneath a fixation cross at the center of the screen. After 1 sec, the card appeared in the center of the screen, replacing the fixation cross. Participants were allowed to press a key only when the card appeared, with a 1-sec deadline. Following their response, participants received feedback (+1 or 0) that remained on the screen for 1 sec, followed by a 1-sec intertrial interval (fixation cross). This trial structure was designed to mitigate the concern that condition-based differences in performance might stem from the label condition being more difficult, by giving participants time to identify where each color label was positioned. This minimizes a potential advantage of the position condition, where participants did not need to know where colors were on a trial-by-trial basis to make a correct response. Giving participants time to identify where each color is positioned before card presentation decreases the difference between the conditions in terms of difficulty, making this confound less likely.

We designed the label and position conditions to engage choice processes with different degrees of generality. The position condition should capture the less general choice process in which the rewarding response is defined by a single motor action, and the label is irrelevant to both choice and learning. The label condition, on the other hand, captures a more general choice process in which the rewarding response (i.e., choice of the correct label) can be made by identifying one of three positions and executing any of the three motor actions, depending on where the correct box label is positioned on the given trial, such that the other dimension (position) remains irrelevant for learning but becomes relevant for choice.

Experiment 2. The task design for Experiment 2 was the same as the task design for Experiment 1, with one important exception - we varied the number of cards per block between 2 and 5, for both position and label conditions. This manipulation has previously been shown to enable computational modeling to disentangle WM and RL processes (Collins and Frank, 2012).The order of blocks was counterbalanced across participants; they completed either label or position blocks first, with the order of set sizes randomized for the first completed condition, and then repeated for the second. In addition, we removed the control condition, given that we previously observed no difference between position and control. Participants completed four blocks of position and label each, where each block within each condition had a different set size.

Analyses

Model-independent analyses. In addition to general diagnostics and standard statistical analyses (see Results), we sought to analyze participants' choices and RTs as a function of how often each motor action and each label had been rewarded for each card. Specifically, we computed card-dependent cumulative reward history (CRH) for both positions P and labels L on each trial for a card C, in each condition:

$$CRH_k^P(C,P) = \sum_{k=1}^{t}(r_k * 1(Card_k = C, Choice_k = P))$$

$$CRH_k^L(C,L) = \sum_{k=1}^{t}(r_k * 1(Card_k = C, Choice_k = L))$$

where r_k is the outcome at trial k in the block, and 1 is the indicator function that takes a value of 1 if the card and position/label match C and P/L, and 0 otherwise. We used this metric to analyze how the integration of two value sources shaped choices when choice format was less/more general. In particular, in the example of the position condition, the position CRH for a card and its associated correct position indicated the past number of correct choices, whereas the CRH for other positions was 0. By contrast, in the same position condition, the label CRH for a card reflected how often each label had been rewarded because of this label being in the correct position. All label CRH values in the position condition were expected to be close to each other because label positions were counterbalanced, but slight differences because of past choice randomness could be predictive of biases in future choices. The opposite was true in the label condition.

To analyze how the value integration for each type of choice shaped decisions, we focused on the error trials and computed the proportion of errors driven by the other irrelevant choice dimension. We reasoned that if participants were randomly lapsing, any of the two possible errors should be equally likely. However, if participants experienced value interference, they should be more likely to select the error with the higher CRH in the irrelevant

dimension. In the label condition, such an interference error would look like selecting the position/motor action that was rewarded on the previous trial, although the correct label had switched positions since (2.1B). In the position condition, an interference error would occur when participants selected the previously rewarded label that had switched positions, instead of the label currently corresponding to the position/motor action that is always correct for the given card (2.1B). We ran a trial-by-trial analysis using a mixed effects general linear model to characterize choices. We used trialby-trial reward history difference $(RHD) = CRH(chosen) - mean(CRH(unchosen))$ between chosen and unchosen boxes, for both positions and labels, and tested whether this discrepancy modulated accuracy and RTs. If participants implemented an optimal decision strategy, their accuracy and RTs should increase and decrease, respectively, with an increased RHD in the relevant choice dimension (i.e., label RHD in label condition, position RHD in position condition). Alternatively, contribution by the irrelevant dimension RHD (i.e., position RHD in label condition or vice versa) would serve as evidence of value interference. Our mixed-effects models had the following general structure:

$$\begin{aligned} Performance = 1 + \beta_1 pRHD + \beta_2 lRHD + \beta_3 t \\ +\beta_4 block + (1 + \beta_1 pRHD \\ +\beta_2 lRHD + \beta_3 t + \beta_4 block|Subject) \end{aligned}$$

where $pRHD$ is RHD based on position reward history and $lRHD$ is RHD based on label reward history. Performance can refer to either accuracy (coded as correct/incorrect) or RTs. In the analysis of Experiment 2 data, we also ran mixedeffects models including predictors that indexed WM mechanisms (set size and delay between presentations of the current stimulus and the most recently rewarded stimulus, which, respectively, correspond to indexing capacity and susceptibility to decay properties of WM) and RL effects (dimension-relevant, card-dependent reward history, calculated from the cumulative number of earned points for each card, indexing reward-based learning):

$$\begin{aligned} Performance = 1 + \beta_{RL} RL + \beta_{WM} WM + \beta_t t \\ +\beta_b block + (1 + \beta_{RL} RL \\ +\beta_{WM} WM + \beta_t t + \beta_b block|Subject) \end{aligned}$$

where RL corresponds to RL factors such as reward history, and WM corresponds to WM factors such as decay and set size. Note that this is a general structure to demonstrate how we structured the mixed-effects model, but set size and decay were entered as separate predictors. In other words, we explored the effects of interest on a group level, as well as how the estimates of these effects vary across individual participants. We included a predictor for trial number in this model, to ensure that reduction in RTs is not simply conflated

with practice effects/task progression. In addition, we added block number as one of the regressors, to capture overall improvement in performance across the task.

Computational Modeling

RL-WM. To computationally quantify the differences in learning processes between the motor choice/general choice conditions, we used a set of hybrid RL and WM models. Our baseline assumption was that, in the RL process, participants track and update two independent sets of stimulus-action value tables, corresponding to the two possible choice spaces: a card-position value table and a card label value table. We also assumed that the choice policy may reflect a mixture of both the relevant and the irrelevant value tables, potentially leading to interference errors when the value of irrelevant choice dimension (position/label) contributes to the choice process (Fig. 2.6A). In addition to the RL module, a WM module allows us to capture the contribution of WM to performance. The WM memory module learns fast, but is sensitive to short-term forgetting and cognitive load, and is thus particularly identifiable in the second experiment where the set size varies between 2 and 5 (Collins, 2018; Collins and Frank, 2012, 2018). WM also potentially tracks associations between cards and two choice types, and like RL, its policy may reflect a mixture of both relevant and irrelevant associations. We investigated a range of models to pinpoint the computational mechanisms of divergence between the learning processes in the two conditions, by varying the extent to which the models allowed for condition-dependent specificity/model-parameters.

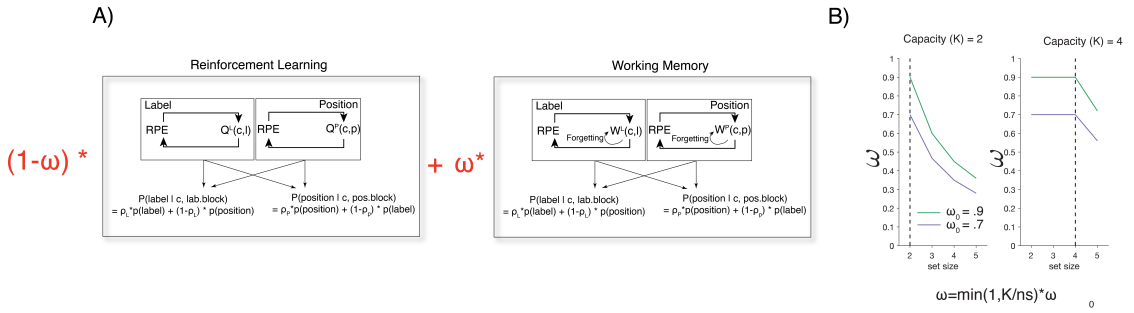


Figure 2.6: (A) In Experiment 1, we used RL model variants, which assume incremental, feedback-driven learning. In Experiment 2, we combined RL and WM modules, under the assumption that learning is a weighted interaction between RL and WM systems. (B) The extent to which participants relied on WM was determined by the WM weight parameter (ω), proportional to participants' WM capacity (K), and inversely proportional to set size.

RL Rule. The RL module assumes incremental learning through a simple delta rule (Sutton and Barto, 2018). Specifically, on each trial t, the values of labels $Q_L(c, l)$ and positions

$Q_P(c,p)$ for the trial's card c and chosen labels and positions l and p are updated in proportion to the reward prediction error:

$$Q^P_{t+1}(c,p) = Q^P_t(c,p) + \alpha * (r - Q^P_t(c,p))$$
$$Q^L_{t+1}(c,l) = Q^L_t(c,l) + \alpha * (r - Q^L_t(c,l))$$

where α is the learning rate and r = 0/1 is the outcome for incorrect and correct trials. Q-tables are initialized at 1/3 (3 = total number of positions/labels) at the start of each block to reflect initial reward expectation in the absence of information about new cards.

WM Rule. Unlike RL, WM processes can encode and retain the previous trial's information perfectly, thus enabling one-shot learning. Note that other cognitive processes (such as episodic memory) could also support one-shot learning and contribute to learning behavior in this experiment; however, here, we focus on RL and WM processes only, as our protocol does not allow us to disentangle other contributions (Yoo and Collins, 2022). Following previous works (Collins, 2018; Collins and Frank, 2012; Collins et al., 2014), we model the one-shot learning in WM by storing the immediate outcome as the stimulus–response weight:

$$W^P_{t+1}(c_t, p_t) = r_t$$
$$W^L_{t+1}(c_t, l_t) = r_t$$

Prior work in similar tasks (Frank et al., 2007; Gershman, 2015; Katahira, 2018; Niv et al., 2012) has shown an asymmetry in learning based on positive/negative feedback, such that individuals are less likely to integrate negative feedback while learning rewarding responses. Thus, we included a learning bias (LB) parameter ($0 \leq LB \leq 1$), which scales the learning rate α by LB when participants observe the negative feedback. We applied LB to both RL and WM (for both position and label dimensions, showing only an example for position here):

$$Q^P_{t+1}(c,p) = Q^P_t(c,p) + LB * \alpha(0 - Q^P_t(c,p))$$
$$W^P_{t+1}(c,p) = W^P_t(c,p) + LB * \alpha(0 - W^P_t(c,p))$$

To capture the phenomenon that maintenance of information in WM is short term and subject to interference, the weights stored in WM are susceptible to decay (ϕ) at each trial, which pulls all position and label weights to their initial values (W^{P_0}, W^{L_0}) following the application of the WM forgetting rule:

$$W^P_{t+1} = W^P_t + \phi * (W^{P_0} - W^P_t)$$
$$W^L_{t+1} = W^L_t + \phi * (W^{L_0} - W^L_t)$$

Whereas information stored in WM decays over time, reflecting the well-documented short time-scale of WM maintenance, RL is assumed to be a more robust system that is less susceptible to forgetting. Therefore, it is theoretically less justified to include a decay mechanism for Q-values. Nevertheless, for completeness, we fit the version of the model with a separate decay process in the RL module as well and confirmed that it does not improve the model fit. Thus, in further implementations of the RL-WM model, we limited decay implementation to the WM module only.

Policy. We used the softmax function to transform WM weights and RL Q-values into choice probabilities to produce position choice policies P^P_{RL} and P^P_{WM}:

$$P^P_{RL}(p|c) = \frac{exp(\beta * Q^P_t(c,p))}{\sum_{i=1}^{3} exp(\beta * Q^P_t(c,p_i)}$$
$$P^P_{WM}(p|c) = \frac{exp(\beta * W^P_t(c,p))}{\sum_{i=1}^{3} exp(\beta * W^P_t(c,p_i)}$$

We applied the same softmax transformation to the label Q and W-tables to obtain the label and choice policies P^P_{RL} and P^P_{WM}. This policy permits the selection of choices with higher Q-values/weights with higher probability. The softmax β is the inverse temperature parameter, which controls how deterministic the choice process is.

For each module, the overall choice policy is a mixture of both policies, determined by mixture parameters, ρ:

$$P_{RL}(p_i|pos.block) = \rho_P * P^P_{RL}(p_i) + (1-\rho_P) * P^L_{RL}(label(p_i))$$

$$P_{WM}(p_i|pos.block) = \rho_P * P^P_{WM}(p_i) + (1-\rho_P) * P^L_{WM}(label(p_i))$$

We apply the same mixture process with mixture weight ρ_L for the label dimension blocks:

$$P_{RL}(l_i|lab.block) = \rho_L * P^L_{RL}(l_i) + (1-\rho_L) * P^P_{RL}(position(l_i))$$

$$P_{WM}(l_i|lab.block) = \rho_L * P^L_{WM}(l_i) + (1-\rho_L) * P^P_{WM}(position(l_i))$$

The RL-WM model posits that choice comes from a weighted mixture of RL and WM, where one's reliance on WM is determined by the WM weight (ω) parameter:

$$P(p|c) = \omega * P_{WM}(p|c) + (1 - \omega) * P_{RL}(p|c)$$
$$P(l|c) = \omega * P_{WM}(l|c) + (1 - \omega) * P_{RL}(l|c)$$

where ω reflects the likelihood of an item being stored in WM and is proportional to the ratio of capacity parameter (K) and block set size (or number of stimuli; ns), scaled by the baseline propensity to rely on WM (ω_0; Fig. 2.6):

$$\omega = min(1, \frac{K}{ns}) * \omega_0$$

We further modified the policy to parameterize additional processes. For instance, individuals often make value-independent, random lapses in choice while doing the task. To capture this property of behavior, we derived a secondary policy by adding a random noise parameter in choice selection (Nassar and Frank, 2016):

$$P' = (1 - \epsilon) * P + \epsilon * \frac{1}{n_A}$$

where n_A is the total number of possible actions and $1/nA$ is the uniform random policy and is the noise parameter capturing the degree of random lapses. We fit the different configurations of the full RL-WM model to the data from Experiment 2, where we varied set size, which permitted us to modulate WM involvement. Note that previous research with experiments including multiple set sizes has shown that single process models (such as RL with decay or interference) are insufficient to capture set-size effects; indeed, these processes can be decomposed into both pure cognitive load and increased forgetting with longer delays between stimuli across set sizes. Thus, in Experiment 2, we do not consider RL-only models. In the absence of a set-size manipulation, it is not possible to separately identify the WM module from the RL module. Thus, in the first experiment, where set size is fixed, we only consider the RL module as approximating the joint contributions of both, and do not include a WM module. Because the RL module summarizes both RL and WM contributions, we add to it a short-term forgetting feature of the RL-WM's WM module: Specifically, we implemented decay in Q-values for all cards and all choices at each trial:

$$Q^P_{t+1} = Q^P_t + \phi * (Q_0 - Q^P_t)$$
$$Q^L_{t+1} = Q^L_t + \phi * (Q_0 - Q^L_t)$$

whereas in the RL-WM model, the forgetting parameter is limited to the WM module only. The list of baseline parameters for RL-WM model (Experiment 2) includes learning rate (α), inverse temperature (β), lapse (ϵ), LB, decay (ϕ), capacity (K), WM weight (ω), and value mixture (ρ). The baseline RL model (Experiment 1) include learning rate (α), inverse temperature (β), lapse (ϵ), LB, decay (ϕ), and value mixture (ρ). We explored different model variants by making different parameters fixed/varied across conditions. In the RL-WM (Experiment 2) model, the parameters did not vary as a function of set size (i.e., same label/position parameter values for all set sizes).

Model Fitting and Comparison

Fitting procedure. In both Experiment 1 and Experiment 2 modeling, we used maximum likelihood estimation to fit participants' individual parameters to their full sequence of choices. All parameters were bound between 0 and 1, with the exception of the β parameter, which was fixed to 100 (found to improve parameter identifiability here and in previous similar tasks; Master et al., 2020), and the capacity parameter (K) of Experiment 2 models, which could take on one of the discrete values between 2 and 5. To find the best fitting parameters, we used 20 random starting points with MATLAB'sfmincon optimization function (Wilson and Collins, 2019).

Model validation. To validate whether our models could indeed capture the behavioral properties we set out to model, we simulated performance from the best parameter estimates for each participant 100 times per participant. We then compared whether the model predictions from the simulated data captured the patterns we observed in the actual data set.

These simulations also allowed us to ensure that our fitting procedure could adequately recover parameters in our experimental context, by fitting the model to the simulated data and evaluating the match between the true simulation parameters and recovered parameters fit on simulated data.

Model comparison. Exploring the full model space would lead to a combinatorial explosion of models, given the possible variations along all parameters. Thus, to explore the model space, we took a systematic approach by starting with the most complex model (all parameters varied across conditions), and gradually decreasing model complexity, while also monitoring the goodness of model fit. Specifically, we reduced the model complexity only if we found that removing a parameter improved the model fit. We chose this approach to conduct model comparison systematically, testing out plausible parameter configurations with varying complexity. We compared the models using the Akaike Information Criterion (AIC; Wagenmakers and Farrell, 2004), which evaluates model fit using likelihood values and applies a complexity penalty based on the number of parameters. To ensure that our models

were identifiable with AIC, we computed a confusion matrix (Wilson and Collins, 2019) by creating synthetic data sets from each model, fitting each model to the simulated data sets, and performing AIC based comparison where the ground truth was known. This confirmed that AIC was adequately penalizing for model complexity in our situation.

2.6 Supplementary Materials

Experiment 1 additional model comparisons. We tested whether an additional decay parameter, an additional mixture parameter, a mixture parameter shared across the two conditions and free softmax temperature parameter improved the fit to the data. These models did not improve the fit compared to M3 (our winning model).

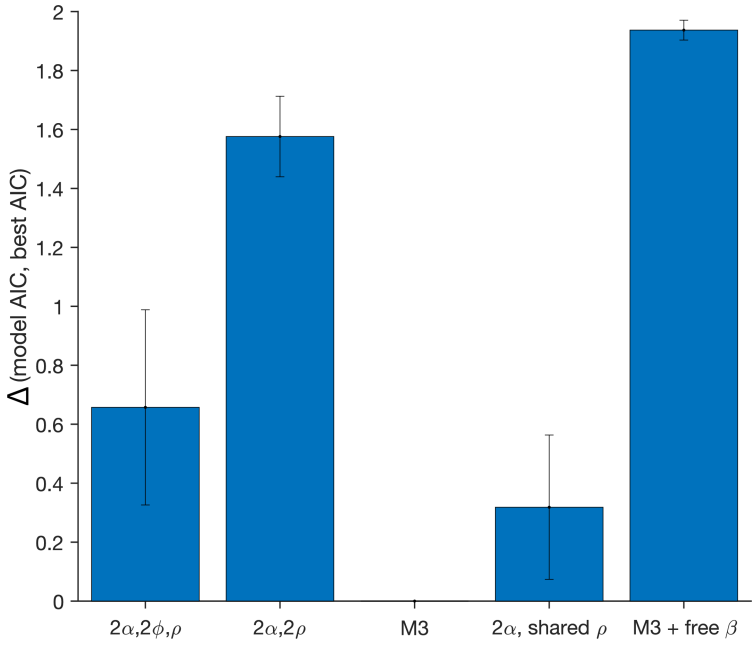


Figure 2.7: Additional models tested in Experiment 1.

Experiment 1 confusion matrix. To demonstrate the identifiability of our models (i.e. models are meaningfully different from one another), we simulated the data from each model on 62 iterations (number of participants). We used best parameter estimates for each participant to create a synthetic data set on each iteration. We then fitted each of the models to each simulated data set with 20 random starting points, to match the fitting procedure to participants' data. Next, we computed the proportion of the times each model fit the best.

If the models are identifiable, the model the data was simulated from should fit the best on most iterations (i.e. the matrix should have the highest proportion of best fit values on its diagonal). The confusion matrix showed that our models are identifiable.

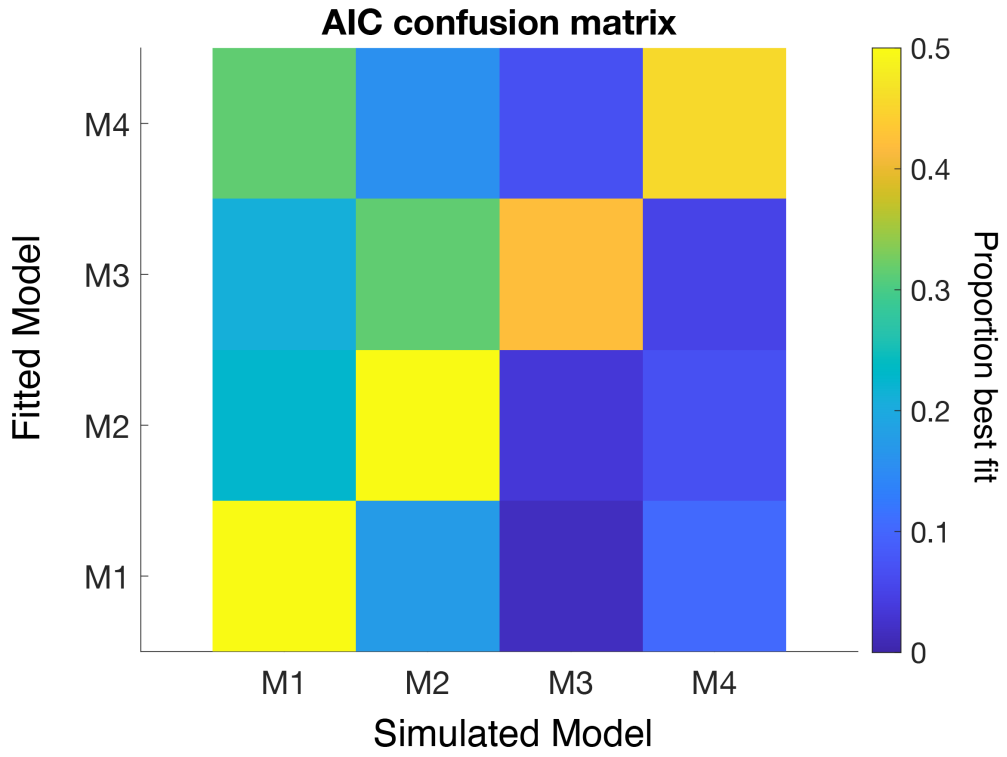


Figure 2.8: Confusion matrix of the main models tested in Experiment 1.

In our second experiment, we fit a considerable range of models, starting with the most complex (all RL + WM parameters condition-dependent), to the simplest (all parameters shared across conditions). We systematically varied the complexity of the model, while monitoring the model fit/complexity tradeoff using AIC scores, in order to test which parameters are necessary for capturing the difference between the conditions while also making sure our models are not overfitting (Fig. 2.9).

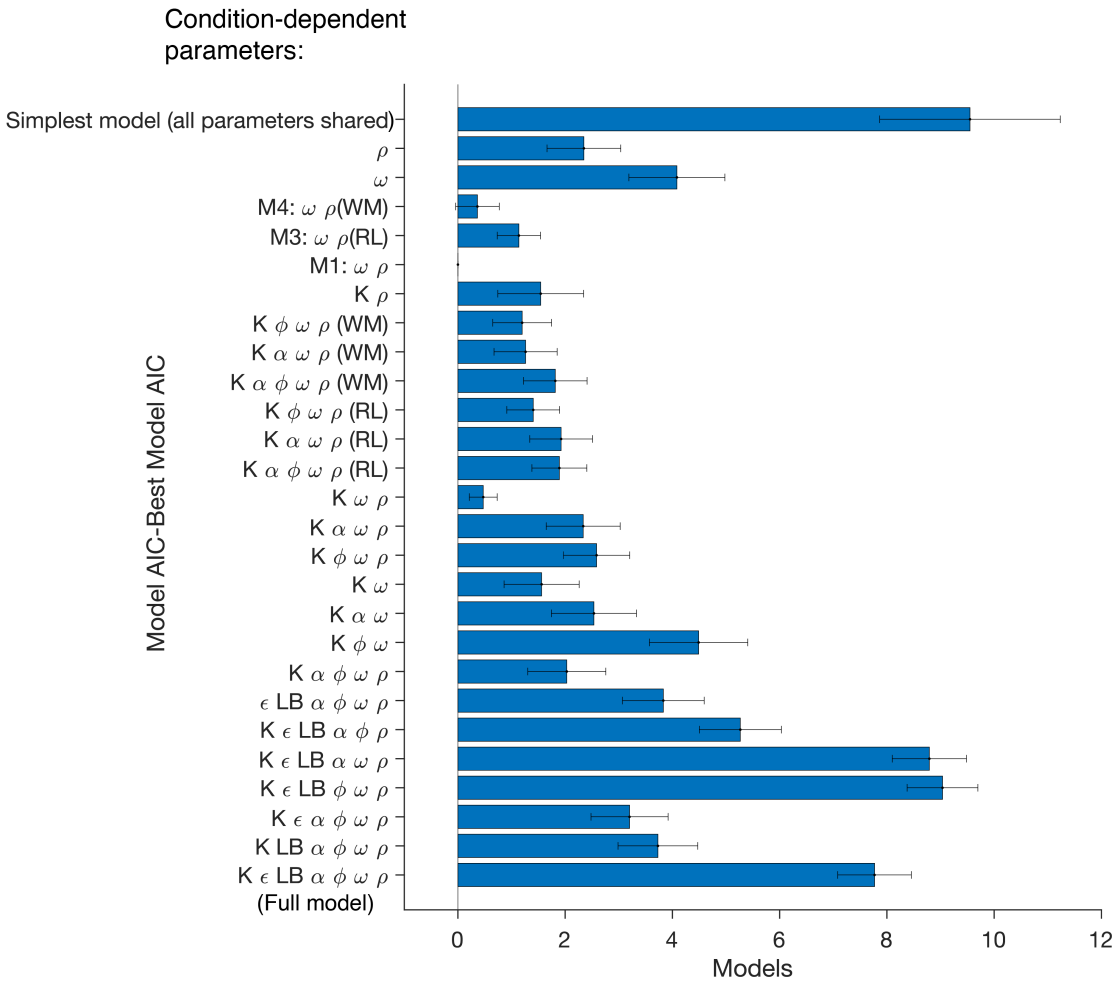


Figure 2.9: AIC comparison of models tested in Experiment 2. Here we show the difference in individual AIC scores between M3, and all other models that were tested.

Experiment 2 Confusion Matrix. We tested the identifiability of our models in Experiment 2 by creating a confusion matrix, similarly to Experiment 1 (Wilson and Collins, 2019). We constructed two different confusion matrices, which test for identifiability of our model along 2 different dimensions. Our first confusion matrix allowed us to test whether the models with different placements of the ρ parameter (i.e. with wrong choice dimension policy mixture in RL, WM or both) are meaningfully dissociable. The confusion matrix shows that the models with mixture ρ in RL and WM policy can be dissociated (Fig. 2.10). The data simulated from the model with ρ parameter in both WM and RL policy was fit equally well

by that model and the model with ρ in WM policy alone. This is consistent with our results, as model comparison revealed that AIC scores did not meaningfully differ between these two models. Note that the models included in the confusion matrix are nested models (differing by at most 1 parameter), or in the case of the second confusion matrix, identical models in terms of number of parameters, but with different *rho* parameter placements. Therefore, we did not expect the AIC scores to be considerably different for paired model fits paired with data simulated across different models.

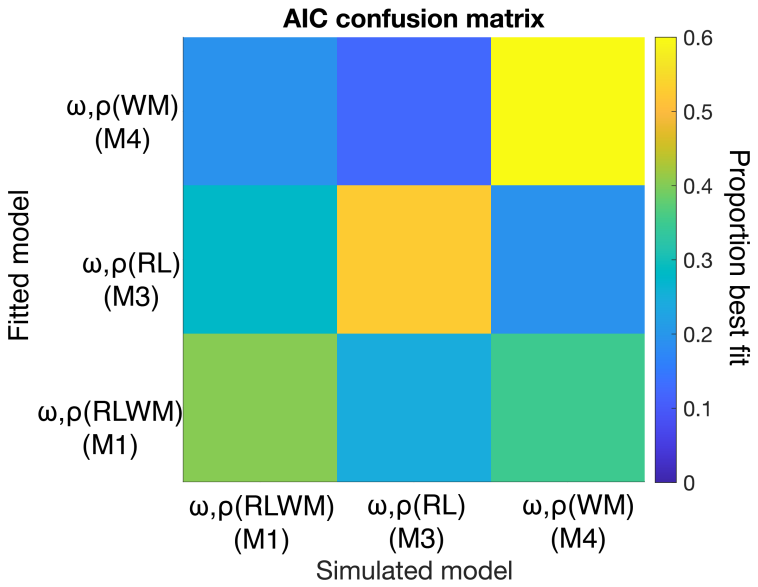


Figure 2.10: Confusion Matrix 1. Proportion of times the models fitted different simulated data sets best, based on cross-fit AIC scores for models with different placement of ρ paramater.

Our second confusion matrix tested whether we can dissociate the model we converged on in the main text (M1, ω with RL-WM ρ) from variations of model with 1) no ρ parameter, and 2) shared WM weight ω. Our results showed that our models are mostly identifiable, with an exception of M2 (Fig. 2.11). However, M2 cannot produce the observed qualitative error patterns, providing another method to rule out this model.

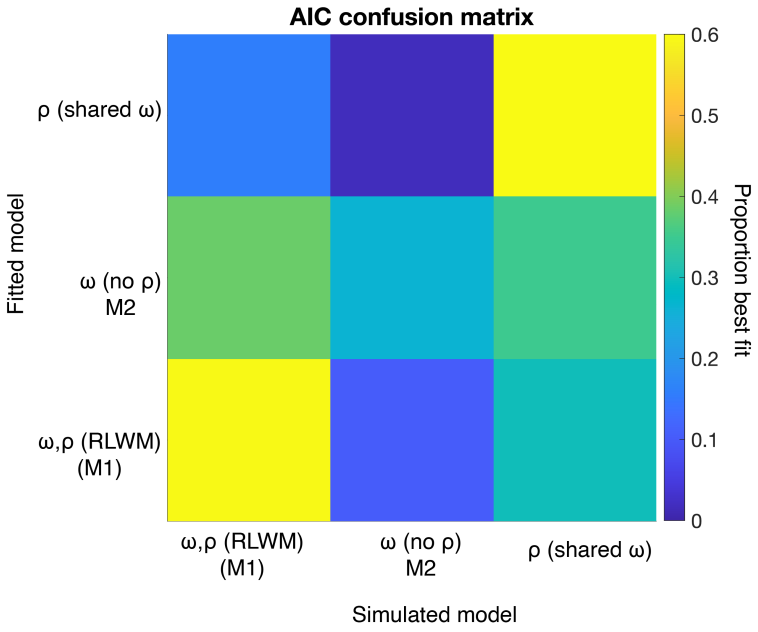


Figure 2.11: Confusion Matrix 2. Proportion of times the models fitted different simulated data sets best, based on cross-fit AIC scores for models with condition dependent ρ and ω parameters (M1), condition dependent ω (M2), and condition dependent ρ.

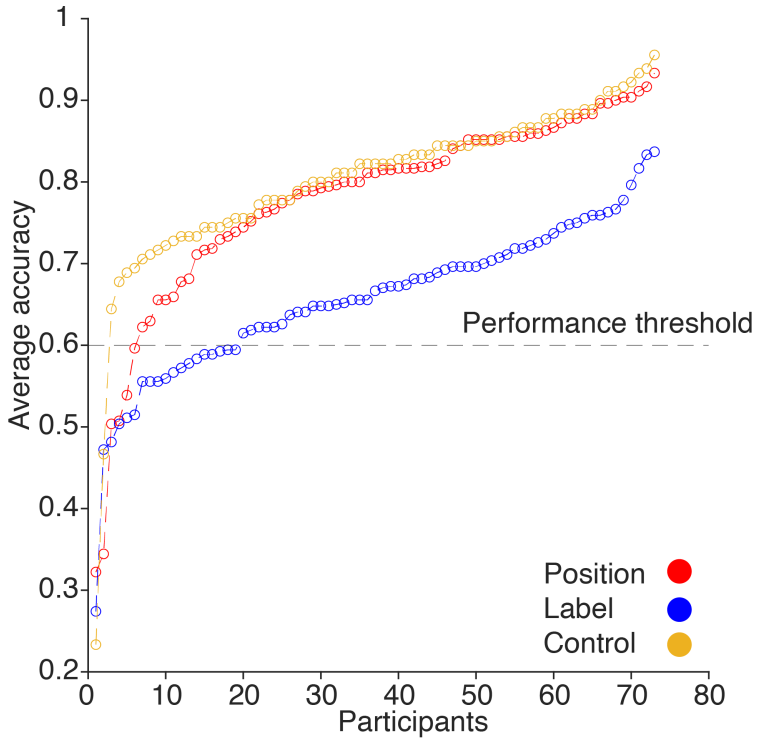


Figure 2.12: Exclusion criteria based on the task performance. We averaged accuracy across all conditions. Based on the "elbow point", most participants' performance is above .60, so we used .60 as criteria for exclusion.

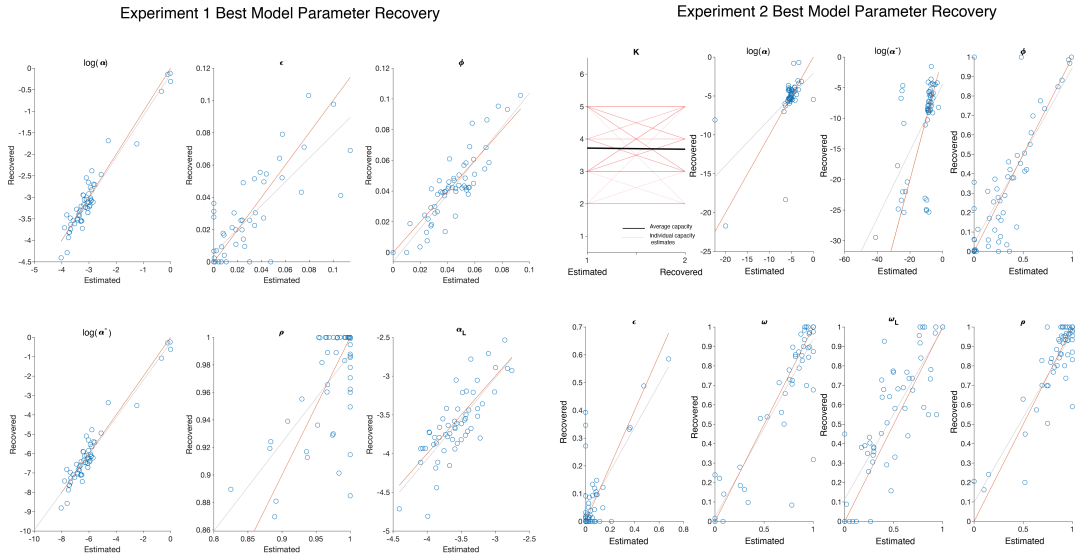


Figure 2.13: Parameter recovery for the best models in Experiment 1 and Experiment 2.

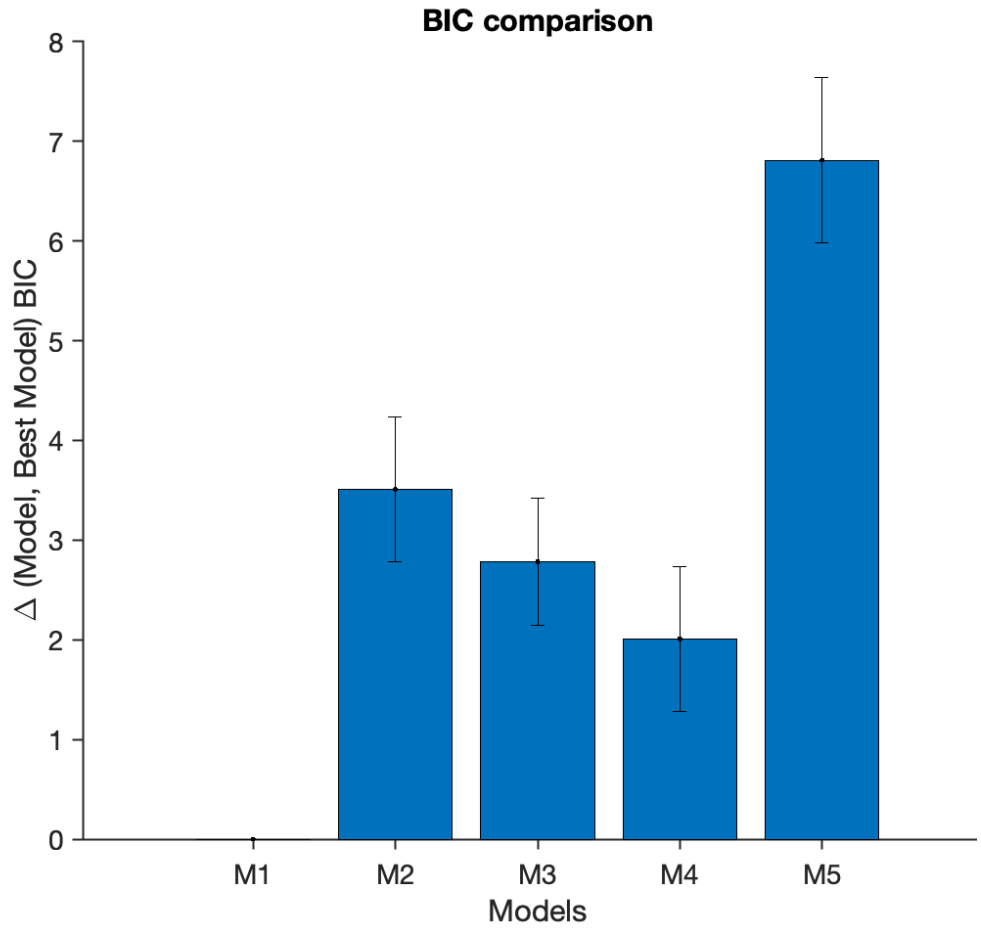


Figure 2.14: Parameter recovery for the best models in Experiment 1 and Experiment 2.

M1	M2	M3	M4	M5
0.20	0.18	0.19	0.22	0.18

Table 1. Protected Exceedance Probability of tested models in Experiment 2, computed based on AIC evidence. Bayes Omnibus Risk BOR (indexing the probability that model frequencies are equal) = 0.94, which suggests that frequency is not strongly differentiable between models.

M1	M2	M3	M4	M5
1	0	0	0	0

Table 2. Since BIC provided stronger differentiation between models, we computed the protected exceedance probability based on BIC evidence. Bayes Omnibus Risk (BOR) = $1.29e-12$, with $PXP(M1) = 1$,suggests that M1 has the highest frequency.

Chapter 3

Subgoals in Hierarchical Reinforcement Learning

3.1 Abstract

In the hierarchical reinforcement learning framework, complex learning problems are decomposed into sub-components that lead to subgoals, and that can flexibly be combined and used to solve a problem in an absence of immediate rewards. Subgoals are hypothesized to be reinforcing. Little is known about how subgoals are used to support learning, with different theories proposing different factors subgoals depend on to affect learning. Here, we show that subgoals reinforce choices even when controlling for other factors often associated with subgoals (such as novelty, bottleneck or external rewards), and that people strategically search the subgoal space. Pseudo-reinforcing effect of subgoals can also transfer to an independent task, with the caveat that its generalizability seems to be possible only with explicit recognition of subgoal features.

3.2 Introduction

Representational hierarchy of information, which assumes simple, concrete bits of information are organized into increasingly more complex and abstract levels is a concept which has been critical in characterizing how humans efficiently process large amounts of information (Botvinick, 2008; Lashley et al., 1951; G. A. Miller et al., 2017). For instance hierarchical organization can be leveraged to understand how humans solve complex tasks: high-level strategies can be used to inform individual decisions (Badre and D'esposito, 2009; Collins and Frank, 2013), final goal of the task can be represented using a set of intermediate goals that can be accessed more readily (Diuk et al., 2013; Ribas-Fernandes et al., 2011; Solway et al., 2014), perceived input is processed from a coarse to increasingly more fine level (Fleuret and Geman, 2001; Hegdé, 2008). Beyond offering a theoretical framework for understanding

a range of cognitive processes, an extensive array of neuroscience research supports the concept of hierarchical organization within neural systems (Badre and D'esposito, 2009; Badre and Nee, 2018; Koechlin and Summerfield, 2007; Koechlin et al., 2003; E. K. Miller and Cohen, 2001). This empirical evidence closely aligns with the theoretical models that posit a structured, hierarchical organization of cognition.

A computational framework that formalizes how hierarchy is leveraged for problem solving is hierarchical reinforcement learning (HRL,Botvinick et al., 2009; McGovern and Barto, 2001). The HRL framework posits that agents interact with their environment through primitive actions, and over time acquire more complex policies (Sutton et al., 1999). These temporally-extended policies, also referred to as options (Stolle and Precup, 2002), can be flexibly applied to accelerate learning when encountering new problems (Solway et al., 2014; Tomov et al., 2021). This is a departure from the flat reinforcement learning which is constrained to learning of individual state-action associations through a trial-and-error process (Sutton and Barto, 2018), where an agent forms individual state-action associations, based on the observed feedback at each time-step. The benefit of HRL is that it (through temporally extended policies/options) accounts for learning in highly dimensional tasks that would quickly lead to a combinatorial explosion of individual state-action associations an agent would need to explore under the premise of basic RL. In addition, environments with sparse rewards (e.g. where feedback is not observed at each time-step) represent a critical limitation of RL algorithms, that HRL addresses through subgoals - intermediate task milestones that can support learning in the absence of feedback.

Specifically, subgoals serve the function of decomposing problems into policies that terminate when the subgoal is reached (Dayan and Hinton, 1992; Karlsson, 1994; Vezhnevets et al., 2017). Reaching a subgoal produces a pseudo-reward signal that strengthens formation of local policies by reinforcing preceding action sequences, even in the absence of external reward (Diuk et al., 2013; Ribas-Fernandes et al., 2011; Ribas-Fernandes et al., 2019). The pseudo-reinforcing effect distinguishes subgoals from rewards: subgoals do not reinforce the global policy of the agent toward an overall goal (presumably a standard reward), only the local policy leading to the subgoal. Various research studies have explored and proposed mechanisms through which the pseudo-reinforcing effect of subgoals manifests, but many questions remain. Some accounts indicate that subgoals rely on shared neural substrates as standard rewards for their reinforcing effects (McDougle et al., 2022), others have shown distinct neural prediction error signals associated with rewards and subgoals (Ribas-Fernandes et al., 2011). Account of intrinsic motivation and curiosity argues that subgoals depend on effects of novelty and surprisal to encode preceding action sequences as meaningful (Baldassarre and Mirolli, 2013; Chentanez et al., 2004; Singh et al., 2010). For instance, when feedback is not immediately delivered, encountering a novel or surprising event can provide a signal that actions that led to that event are meaningful and should be repeated (Eckstein and Collins, 2021; Schmidhuber, 1991). Indeed, there is evidence that neural data aligns with this computational theory, showing that novelty/surprise recruit neural signals (e.g. dopaminergic signaling) that are typically associated with rewards (Bromberg-Martin et al.,

2010). An alternative explanation defines subgoals as bottleneck states that are most frequently visited in successful attempts to reach the final reward (Diuk et al., 2013; McGovern and Barto, 2001); they have also been defined using concepts of graph theory, such as centrality, as states that have the highest degree of connectedness in the state space (Şimşek and Barto, 2008).

It is noteworthy that some of the accounts of how subgoals support learning might be in conflict with the premise that subgoals are not reinforcing in a way rewards are. Specifically, if subgoals are defined as states that most frequently precede rewards, then the effect of subgoals might entirely be dependent on the value inherited from the terminal reward through TD-like mechanisms (O'Doherty et al., 2003; Schultz et al., 1997; Sutton, 1988; Sutton and Barto, 1990). Relatedly, intrinsic motivation accounts might conflate subgoals with novelty and surprise, when there is no evidence that subgoals necessitate either of these. In other words, it is unclear whether pseudo-reinforcing effect of subgoals can be isolated beyond that of reward association, novelty/surprise or structure.

To explore this question, we developed a hierarchical learning task that involves combining simple actions sequences, followed by subgoals, into more complex ones to secure rewards. We carefully decoupled subgoals from rewards by incorporating subgoals into both rewarding and non-rewarding trial sequences. In addition, we separated subgoals from elements of novelty and surprise by ensuring participants could anticipate observing them on each trial. This allowed us to behaviorally test the more isolated pseudo-reinforcing effect of subgoals. Moreover, given the critical characteristic of subgoals in HRL is their ability to be generalized to enhance performance in new tasks, we investigated whether these subgoals could inform decision-making across a set of separate tasks.

Our findings indicate that participants generally utilized subgoals to hierarchically solve the task, strategically navigating through potential subgoal options to organize their actions in the absence of immediate rewards. While behavioral patterns of the majority of participants indicated sensitivity to subgoals, only a subset displayed generalization of subgoal to separate tasks - potentially restricted only to participants who explicitly recognized features that defined the subgoals. These outcomes suggest that subgoals can influence behavior beyond motivations based on surprise or direct prediction of rewards, though their broad application, under this restrictive definition, across different task contexts is not assured.

3.3 Methods

Experiment

Participants.

We collected data from a total of 107 participants (70 women, age mean = 22.2 years, SD = 4.25 years, age range = 18–38 years) from the University of California, Berkeley, Psychol-

ogy Department's Research Participation Program and from online research platform Prolific (www.prolific.com). All participants provided written informed consent before beginning the experiment, in accordance with the University of California, Berkeley, Institutional Review Board (IRB) policy. Participants from RPP pool received course credit for their participation; participants who took part in the study through Prolific received monetary compensation ($10.63/hour, the Prolific recommended rate). We excluded 13 participants who did not meet participation criteria required by our IRB protocol.

Performance exclusion criteria. We further excluded 31 participants based on their performance. Specifically, we excluded participants who failed to discover at least one correct action sequence throughout the entire task, if their response patterns suggested that they were not compliant with the task (e.g. repeating the same keys or same sequences across blocks). This resulted in the final sample of 63 participants whose data was analyzed.

Learning phase.

Participants played a game where their goal was to obtain golden coins by inputting 4 keys sequentially into a machine (Fig. 3.1). The machine generated a coin if the input 4-key sequence (composed of two simpler 2-key subsequences) was correct for the current block. Each trial was divided into two subtrials, marked by an execution of the first and the second 2-key sub-sequence. At the completion of each 2-key sub-sequence, a token appeared on the machine screen, followed by the final outcome of the trial. To obtain a coin, participants were required to learn a valid 4-key sequence composed of 2 valid sub-sequences and signaled by 2 subgoals represented by the tokens. Participants had 2500ms to press each key; if they failed to do so within specified time frame they received a warning to respond faster and the trial terminated. If the trial was correct participants received a golden coin; if not they observed a puff of smoke. Feedback was presented for 500ms, after which the next trial commenced.

Tokens were defined by 2 dimensions: shape (balloon, boat, car) and color (red, blue, yellow) - resulting in 9 possible token combinations. For each participant only one token dimension (either shape or color) was relevant, with specific features of the relevant dimension reliably signaling reaching a subgoal toward the final goal of producing a gold coin. The tokens of relevant dimension were consistently observed on correct trials, and reliably signaled occurrence of a subgoal. There were 6 possible sub-sequences of two different individual key presses, 3 of which were valid (reliably generating a specific feature of a relevant dimension) and three of which were invalid (reliably generating a feature of the irrelevant dimension).

For example, if the relevant token dimension was shape, and [01] was a correct partial sequence that produced the balloon shape, then every time participants entered [01] they observed a balloon-shaped token. This token shape could appear in any one of three colors, sampled with equal probability. On the other hand, if participants pressed the keys in the reverse order [10], they would always observe a specific color of a token (e.g. yellow) that

could take on any of the 3 shapes (again with equal probability) (Fig. 3.1A). Each of the 3 valid sub-sequences was a part of a valid 4-key sequence in some experimental blocks; invalid sub-sequences never contributed to a valid 4-key sequence (Fig. 3.1B). Therefore, executing a valid 2-key sequence and observing a relevant token dimension (e.g. balloon) could be considered as reaching a subgoal towards final reward at the end of the trial.

Only one 4-key sequence was correct and lead to coins in a single block. Each of the 3 valid 4-key sequences were repeated 6 times across the task, resulting in the total of 18 blocks. Blocks had a maximum of 40 trials, but if participants reached a criterion of a minimum of 10 trials, and at least 8 of the last 10 trials were correct, they proceeded to the next block. Relevant dimension and valid sub-sequences were counterbalanced across participants.

The rationale for the complex nature of the subgoal definition rested on 2 major factors. First, we needed subgoals and non-subgoals to appear equally often, in order to equate the surprise/novelty/bottleneck features for subgoals and non-subgoals . Second, we needed to subgoals to appear as a part of both rewarded and non-rewarded trials, in order to equate extrinsic value for subgoals and non-subgoals, thus ruling out the possibility that subogals impact behavior simply because they are more extrinsically valuable than their counterparts because they always predict rewards.

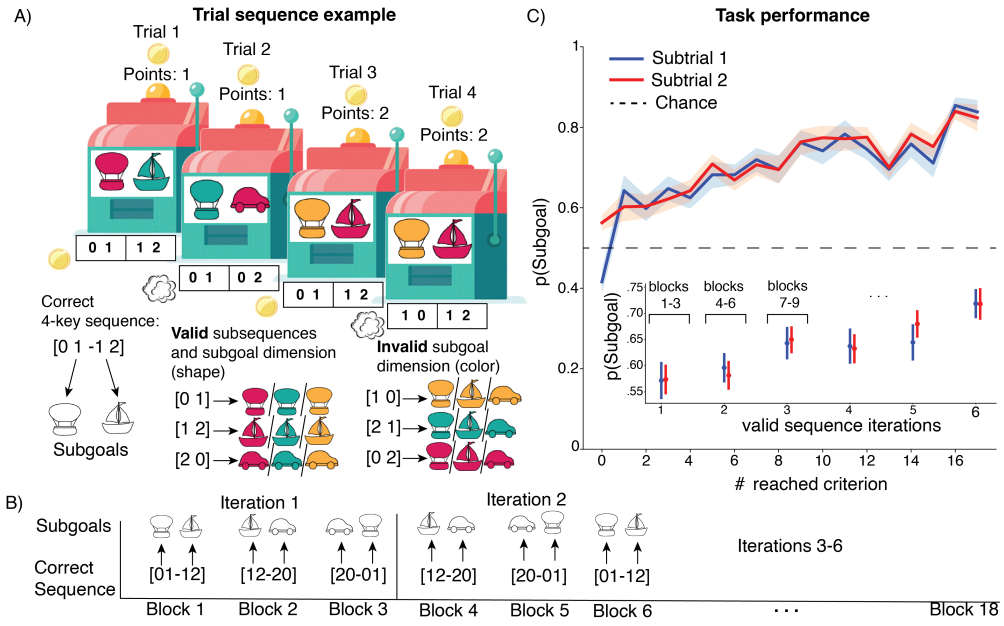


Figure 3.1: A) Participants sequentially pressed 4 keys, observed two tokens after each 2-key presses, and received the final outcome at the end of the trial. Subgoal tokens appeared on both correct and incorrect trials. B) Each of the 3 possible 4-key sequence was repeated 6 times for a total of 18 blocks. C) Participants learned to press subgoal-generating two-key sequences increasingly across the correct sequence iterations (inset), and as a function of the number of times they successfully discovered valid 4-key sequences (top), demonstrating learning of the subgoal structure.

Test

A critical property of subgoals we were interested in is that they can be generalized to novel tasks to accelerate learning (Solway et al., 2014; Tomov et al., 2021). We wanted to test whether our participants generalized subgoals and their pseudo-reinforcing properties to a separate set of tasks they were administered after the learning phase (assuming they inferred the correct subgoal structure in the first place).

First, participants completed a 2-arm bandit reversal task (Fig. 3.2A), which served as an implicit test of whether subgoals' impact on behavior transfers to a separate learning task. This test phase consisted of 4 blocks. In the first block, participants saw 2 novel images they hadn't encountered in the task before (e.g. a star and a diamond). Their task was to choose between the two images to collect points. One of the images was correct: selecting this image by pressing the key corresponding to the position of that image on the screen resulted in gaining 1 point. Selection of the rewarding image resulted in a reward until the

reversal, following which the previously incorrect image became the rewarding one. In the remaining three blocks, participants performed the same type of tasks, with an important difference: outcomes in these blocks were actual token dimensions instead of the points. That is, selection of the rewarding image resulted in a correct subgoal dimension. For example, if previously correct dimension was shape, then participant should always select an image that leads to one of the 3 shapes, instead of an image that leads to one of the colors. In each of the 3 subgoal blocks, we used 3 different pairwise shape-color combinations as outcomes. Each block had 24 trials, with at least 8 trials required for a reversal to occur (3 possible reversals per block). Participants had 1000ms to make their selection; trial feedback was also presented for 1000ms.

Next, participants performed a preference task (Fig. 3.2B). In this task, they observed isolated components of each dimension as stimuli (3 colors and 3 shapes). On each trial, they were presented with a pair of stimuli (e.g. a shape and a color) and asked to select their preferred option. The aim of this task was to test if participants showed a stronger preference for the dimension associated with subgoal tokens from earlier tasks. Participants performed 60 trials of this task - presented in a single block, with 48 trials of cross-dimension pairs (e.g. shape-color), and 12 trials of within-dimension pairs (e.g. shape-shape, color-color). We added within-dimension pair trials to check whether participants showed a bias for particular colors/shapes within dimension. On each trial participants had 1500 ms to respond, and they were not given any feedback. Trials were separated by a 500ms presentation of a fixation cross.

Finally, at the very end of the experiment we asked participants which dimension they thought was the correct one in the task, as well as to rate their confidence in their answer (on the scale of 1-5) (3.2C).

We constructed different tests with an aim to gauge 1) whether participants inferred the subgoals in the training phase, and 2) if so whether subgoals generalized to impact choices in separate tasks in

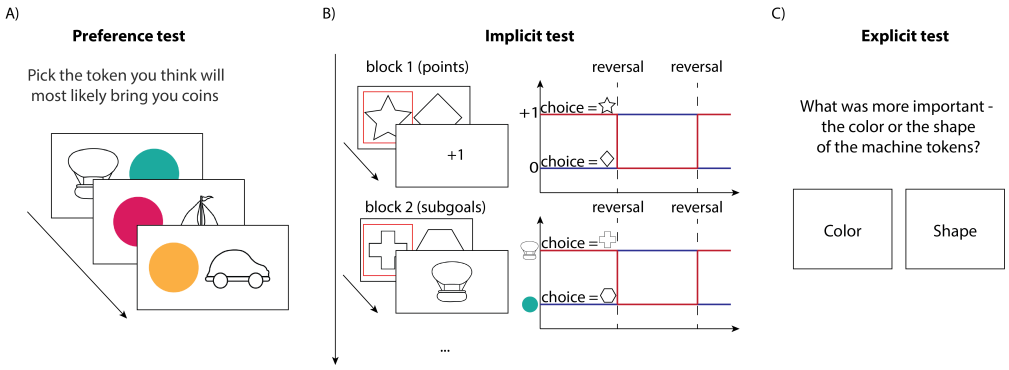


Figure 3.2: A) An implicit test of subgoal effect on performance in a new learning task, where subgoals are treated as trial outcomes. B) Preference test probing whether participants showed preference for the subgoal dimension, independent of feedback. C) Final question probing participants' explicit knowledge of the correct subgoal dimension.

Modeling

To get a better understanding of the extent and manner in which subgoals are generalized from the training phase to the test phase, we analyzed how test phase performance was predicated on individual color and shape values learned in the training phase by applying different computational models. These models were based on different underlying assumptions regarding the generalization of subgoals (Fig. 3.3).

Baseline models

The baseline assumption of the models we tested was that the values (V) associated with each of the shapes and colors are independent, and are thus learned and updated individually. Initially, we assigned uniform values to each color and shape at $\frac{1}{ND}$, with ND=6 representing the total count of distinct colors and shapes. The 6 values were continuously tracked and updated across trials throughout the 18 learning blocks. The baseline model assumption was that these values then served as the primary criteria for decision-making in the test phase.

On each trial of the 18 training blocks, the color and the shape of two sub-trial tokens observed were updated according to the delta rule (Sutton and Barto, 1999):

$$\delta_{shape1} = r - V_{shape1}$$
$$\delta_{color1} = r - V_{color1}$$

where δ represents the reward prediction error (RPE, the discrepancy between the expected and the observed outcome), the variable r denotes the final outcome of the trial, the

number 1 indicates the index of the sub-trial (either 1 or 2), and V_{shape}/V_{color} refer to the values of the token's shape/color, respectively, observed during the sub-trial. The reward prediction error for the second sub-trial (indexed as 2) was calculated in the same way.

The values of observed token shape and color were updated at the individual learning rate α:

$$V_{shape1} = V_{shape1} + \alpha \cdot \delta_{shape1}$$
$$V_{color1} = V_{color1} + \alpha \cdot \delta_{color1}$$

Following the last training block, we modeled participants' choices during the implicit test phase, specifically within the subgoal blocks. Since participants encountered stimuli they haven't observed before, we set their values (Q) uniformly to $\frac{1}{nS}$, where $nS = 2$. Each outcome within the subgoal blocks corresponded to one of the previously encountered colors or shapes, with their respective learned values inherited from the training blocks. The assumption was that if subgoals generalized from the training phase, the correct dimension would be treated as a rewarding outcome (e.g. if shape was a correct dimension, then participants should be more likely to select one of the two images that produces the shape outcome).

To estimate the likelihood of participants' choices, we used the softmax function to transform Q values to stimulus choice probabilities:

$$P(s) = \frac{exp(\beta Q_t(s))}{\sum_{i=1}^{nS} exp(\beta Q_t(s_i))}$$

where the β parameter corresponds to the decision noise, with higher values indicating more deterministic choices (i.e. choosing the stimulus with higher Q value with higher probability).

The accuracy of participants' choices was assessed based on whether the selected stimulus matched the correct stimulus for the current block, where the correct stimulus was defined as the stimulus with the correct subgoal dimension as an outcome.

We next assessed the subjective reward, noting a key distinction from the points block, where rewards are objectively defined (+1 or 0 points). In the subgoal blocks, the reward was based on the participant's learned value of the color and shape outcomes. The subjective reward (r) was determined as follows: if the value of the observed outcome ($V_{observed}$) - be it color or shape—exceeded that of the alternative ($V_{alternative}$) then r was assigned 1; otherwise, it was assigned 0:

$$r(t) = \begin{cases} 1 \text{ if } V_{observed} > V_{alternative} \\ 0 \text{ if } V_{observed} \leq V_{alternative} \end{cases}$$

This allowed us to decouple the subjective value of outcomes from their accuracy. Specifically, a participant might correctly identify a stimulus that results in a subgoal outcome, but experience a lower subjective reward. This scenario is plausible because subgoals might be part of both rewarded and non-rewarded trials in the training phase, potentially leading to situations where a subgoal outcome has a marginally lower value than a non-subgoal outcome.

We then used the subjective reward and stimulus Q values to compute the reward prediction error, and update the Q value of the selected stimulus according to the learning rate (different from training phase learning rate):

$$\delta = r - Q(s)$$
$$Q(s) = Q(s) + \alpha_{test} \cdot \delta$$

Therefore, our baseline model has 3 parameters: training phase learning rate (α), test phase softmax beta parameter (β) and test phase learning rate (α_test).

Next, we expanded this model to account for participants' tendency to repeat the stimulus selection from the previous trial (common behavioral strategy observed in this type of task):

$$P(s) \propto exp(\beta Q + \kappa \text{ same}(s, s_{t-1})) \tag{3.1}$$

with κ parameter referring to stickiness (higher κ values quantifying higher tendency to repeat the stimulus selection from the previous trial).

The expanded baseline model had the following parameters: training phase learning rate (α), test phase softmax beta parameter (β), test phase learning rate (α_test) and stickiness parameter (κ).

Subgoal-consistent boost models

We aimed to explore the hypothesis that subgoals influence choices via a mechanism distinct from the straightforward reward-driven value acquired during the training phase. To investigate this, we introduced a subgoal boost parameter (η) to the value of the correct dimension for the current block:

$$V(subgoal) = V(subgoal) + \eta$$

In this way, the model enforces qualitative differentiation between subgoal and non-subgoal outcomes, beyond the differences in baseline values that should in average be null, as per the design of our experiment, and thus should only reflect variability.

This model had the following parameters: training phase learning rate (α), test phase softmax beta parameter (β), test phase learning rate (α_test) and subgoal boost (η). Like in

the baseline model, we also considered the variation of the subgoal boost model with added stickiness.

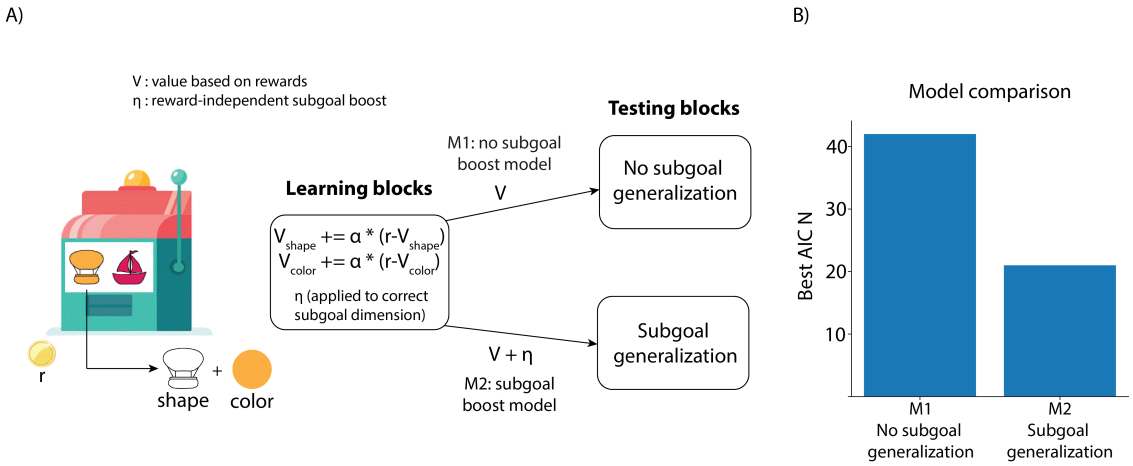


Figure 3.3: Modeling approach. A) Each token was defined by a shape and a color, and the values of each shape/color was updated as a function of the observed feedback at the end of the trial (with values increasing if followed by a reward, and decreasing if followed by no reward). Values of each color/shape are treated as independent and are learned and updated individually across learning blocks. Because subgoals can occur in non-rewarded trials the correct subgoal token dimension does not necessarily have a higher value. B) Results of AIC comparison show that majority of participants' data was better fit by the models in M1 group - the set of models assuming no subgoal generalization beyond the value acquired through reward prediction.

3.4 Results

Performance

We started by examining whether participants were sensitive to the subgoal structure in the task by examining the probability with which they executed 2-key sequence actions that reliably led to the correct subgoal dimension. We examined this proportion both as a function of correct sequence iteration (each of the 3 correct 4-key sequences was repeated 6 times), and the number of times participants finished the block early by showing sufficient evidence that they have succesfully inferred the correct sequence (minimum of 10 trials, with at least 8 being correct, Fig. 3.4A). We used the latter due to the fact that occasionally participants do not discover a correct sequence until later blocks due to the high number of possible key combinations they need to explore in order to discover a correct one; therefore,

aligning their performance to the block number might not be an informative indication of the learning progress. We found that participants overall tended to select 2-key sequences that led to subgoal dimension and reliably resulted in a rewarding outcome more than expected by chance (subtrial 1: $t(62) = 9.58$, $p = 3.7e\text{-}14$; subtrial 2: $t(62) = 8.74$, $p = 1.02e\text{-}12$, and they did so more consistently with more acquired evidence. This suggests that participants were able to discover the correct 4-key sequences throughout the task.

Next, we were interested in exploring whether participants treated the 4-key sequences as a hierarchical composition of simpler sub-sequences. We examined participants response times associated with each of the 4 keys in the trial (Fig. 3.4C). We found that for each of the sub-sequence, participants' first responses were faster at the completion of the sequence (key presses 2 and 4), compared to the initiation of the sequences (key presses 1 and 3) (**correct trials** K1 >K2 : $t(62) = 8.74^*$, K3 >K4 : $t(62) = 6.03^*$; **incorrect trials** K1 >K2 : $t(62) = 9.31^*$, K3 >K4 : $t(62) = 9.08^*$; *all results were at the $p < .001$ level of significance) - replicating the previous results of patterned RTs indicative of hierarchical composition structure (Eckstein and Collins, 2021). This was true for both correct and incorrect trials. Response time associated with the first key of the sub-sequence was faster for the first sub-sequence compared to the second sub-sequence, with an exception of second key comparison in correct trials (**correct trials** K1 >K3 : $t(62) = 3.05^*$, K2 >K4 : $t(62) = 1.40$, $p = .26$; **incorrect trials** K1 >K3 : $t(62) = 4.90^*$, K3 >K2 >K4 : $t(62) = 3.25^*$; * all results were at the $p < .05$ level of significance). These results imply that 1) participants represented 4-key sequences as a hierarchical composition of simpler 2-key sub-sequences, and 2) that second sub-sequence was contingent on the first, thus leading to the faster initiation through a potential frontloading mechanism (Eckstein and Collins, 2021).

Since participants showed evidence of hierarchical representation (e.g. composition of sub-sequences rather than individual keys), and there were many possibilities participants could explore to combine 3 possible keys into a 4-key sequence, we tested whether participants leverage the compositional representation of the 2 sub-sequences to construct a patterned exploration of the subgoal space. We examined whether participants were more likely to repeat the 2-key sequence from the previous trial on the first or second subtrial. To do so, we only computed the probability of repetition based on the trials before participants observe their first reward in the block. We restricted the trial window to only trials before the first reward occurrence because we did not want potential response strategies to be biased by the reward discovery. The results showed that participants were more likely to repeat the first subtrial sequence compared to second subtrial sequence (Wilcoxon Z score = 263, $p = 3.39e\text{-}07$, Fig. 3.4B). This implies that participants potentially picked a sub-sequence, and explored different combinations on the second subtrial before backtracking to switch to a different first subtrial sub-sequence. This strategy potentially aided participants in exploring as much of the subgoal space as possible in a systematic way.

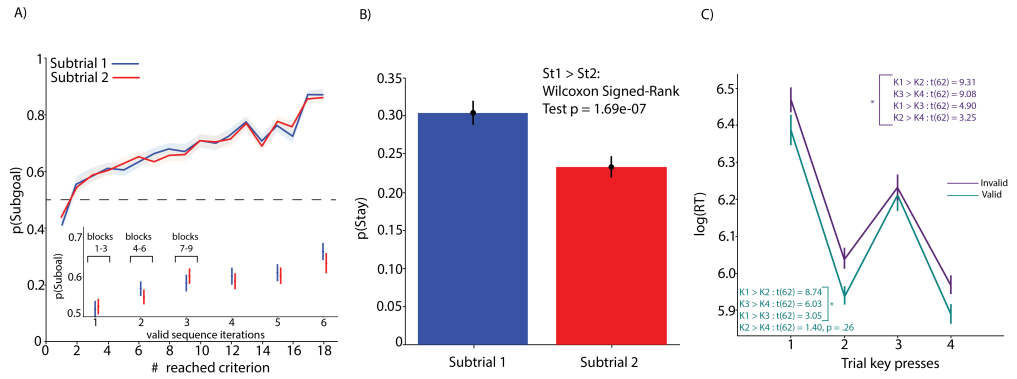


Figure 3.4: Pariticipants tend to press 2 key sequences that lead to subgoals increasingly over the task. B) Prior to observing rewards, participants were more likely to repeat the first than second sub-sequence, suggesting that they implement a systematic search of the subgoal space. C) Response times suggest participants represent 4-key sequences as hierarchically chunked sets of sub-sequences: key presses were faster at sub-sequence initiation, and for the second sub-sequence, replicating previous findings(Eckstein and Collins, 2021)

Do subgoal dimensions affect responses?

Results so far indicated that participants were able to discover valid sequences in the task. However, in theory, they could discover 4-key sequences without necessarily paying close attention to the tokens, or using the tokens to guide their action selection at all. We performed a trial-by-trial logistic regression analysis, which enabled us to assess the effect of observed token dimensions on subsequent action choice and accuracy.

We constructed different regressors representing different types of subgoal representation through the trial history of 1) previously chosen actions, 2) observed tokens and 3) final trial outcomes (Fig. 3.5A). We then used these regressors to predict either accuracy or repetition of the sequence on the subsequent trial. The regressors mapped on to: correct subgoal model (participants replicated the same 2-key action sequences on previous two trials, observed the relevant token dimension and received a reward), correct subgoal model rewarded only on the previous trial (participants executed different 2-key action sequences on previous 2 trials, observed the relevant token dimension on both trials, but recieved reward only on the trial $t-1$), correct subgoal model rewarded two trials ago (same as the previous regressor, except the correct trial occurred on $t-2$), irrelevant subgoal model (participants replicated the same 2-key sequence on the two previous 2 trials, observed same incorrect token dimension and received no reward). In addition to this, we controlled for the final outcome on the previous trial $t-1$, as well as the pattern of choices that suggest absence of a subgoal model.

We found that having a correct subgoal model significantly contributed to higher accuracy

(median β coefficient = 0.55, Wilcoxon = 38, p = 3.12e-11, Fig. 3.3B). On the other hand, having an irrelevant subgoal model impacted accuracy negatively (mean β coefficient = -.30, t(62) = -8.90, p = 1.06e-12). In sequence repetition regression, we found that both relevant (median β coefficient = .47, Wilcoxon = 89, p = 3.14e-10) and irrelevant (mean β coefficient = 0.41, t(62) = 9.82, p = 2.93e-14) subgoal model increased the likelihood of in repeating the actions from trial t-1 to trial t (Fig. 3.5C). This suggests that subgoal tokens did guide choice selection, potentially beyond the simple effect of being predictive of rewards (e.g. since participants were more likely to repeat actions to reproduce the tokens with dimension that did not represent a subgoal, suggesting a potential incorrect subgoal model that had a pervasive effect on behavior). We used one sample t-test on coefficients which were normally distributed, and otherwise used a non-parametric version (Wilcoxon test).

To verify that participants were actively engaging with the task and paying attention to the tokens, we interspersed probe trials at random intervals throughout the training blocks. During these trials, participants were prompted to identify two out of nine possible tokens that they had observed in the most recent trial. The results from these probe trials indicate that participants were generally successful in identifying the correct tokens, confirming their attentiveness. Participants were able to identify the token they previously observed both by relevant (subtrial 1: relevant dimension t(62) = 15.96, p = 1.21e-23 subtrial 2: t(62) = 14.16, p = 4.23e-21) and irrelevant dimension (subtrial 1: t(62) = 15.39, p = 7.45e-23 , subtrial 2: t(62) = 11.85 , p = 1.39e-17) more accurately than expected by chance. There was no significant difference in accuracy for relevant and irrelevant dimension (subtrial 1: t(62) = 0.31 , p = 0.75; subtrial 2: t(62) = 0.26 , p = 0.78, Supplementary Fig. 3.7).

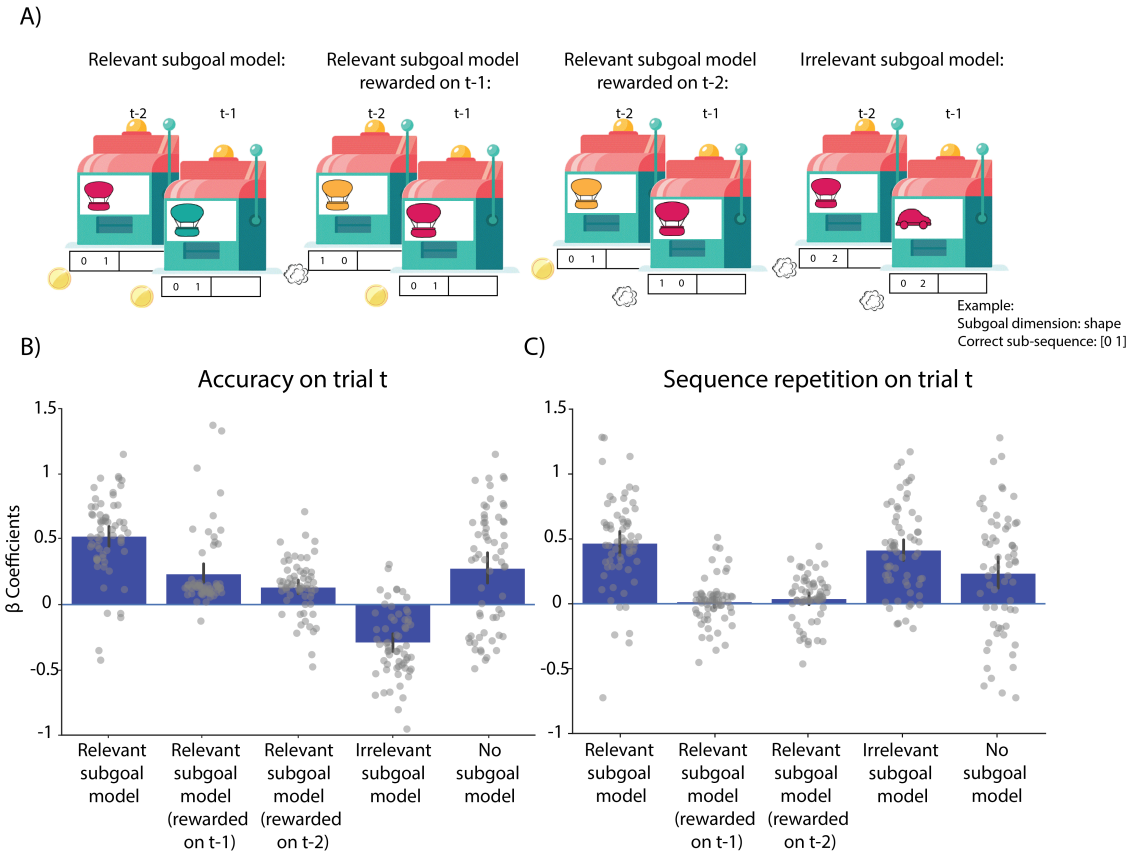


Figure 3.5: A) Trial-by-trial regression model predictors used to to test the effect of different subgoal representations inferred by participants on 1) accuracy and 2) sub-sequence repetition through trial history of observed tokens, outcomes and selected actions. Note that we only show first sub-trial for the clarity of explanation of how the regressors were constructed but we have done the same for the second sub-trial. B) Subgoal outcomes impact accuracy and C) sequence repetition.

Identifying subgoal generalization

The preliminary findings suggest that, on the whole, most participants were able to complete the task; we also found evidence of the effect of subgoal model on sequence choice and accuracy. Next, we sought to determine if the subgoals could extend their influence to action selection and learning behavior in a separate task, thereby providing more robust evidence of subgoals effects beyond mere associations with rewards or intrinsic motivational elements driven by surprise and novelty.

We designed three distinct assessments to explore participants' understanding of subgoals, as detailed in the Methods section. In both the preference task and the final question, we inquired whether participants could explicitly discern the specific dimension (either shape or color) that delineated the subgoals - through preference for the subgoal dimension and explicit recognition of the difference between correct/incorrect dimension respectively. For the implicit assessment, we utilized token dimensions as learning outcomes to test whether participants were more biased towards choices leading to outcomes that were represented by a relevant subgoal dimension.

Overall we observed a lot of variability in performance across 3 tests (Supplementary Fig. 3.8). To conduct an analysis in a systematic way, we opted to first identify if participants showed sensitivity to subgoals beyond what can be explained by the value inherited through association with reward. We did so by fitting different models to participants' choices in the implicit test phase, based on the value of each of the three colors and shapes learned during 18 blocks of training (Fig. 3.3). In other words, we assumed that participants learn the value (V) of each color and shape, based on the observed feedback, at a rate unique to each participant (α). Because each of the token dimensions can occur during rewarded and non-rewarded trials, the expected values for each of the 3 colors and shapes should, by design, be roughly the same; we confirmed this (Supplementary Fig. 3.10). Consequently, decisions in the implicit test based solely on these values should reflect no bias for outcomes associated with previously identified subgoals.

Further, we explored models incorporating a hypothesis that participants add a subgoal boost (η) to the value of the correctly identified subgoal dimension during the test phase, thus valuing choices leading to subgoal outcomes more highly. In total, we examined four models (including variations with and without a stickiness parameter κ), two of which relied solely on value (V), and two that incorporated an additional subgoal bonus (η).

After fitting the models to the data from the implicit test phase and conducting an AIC (Akaike Information Criterion, Akaike, 1998) comparison, we found that the model assuming addition of subgoal bonus to the reward-driven value of the relevant dimension had the best average AIC score, and that stickiness improved fit in both models (Supplementary Fig. 3.9). However, a closer look at individual participants' AIC scores revealed that the data of majority was better fit by the model without the η parameter (Supplementary Fig. 3.9). We divided our sample into two groups: M1, consisting of participants whose data was better fit by the models without the subgoal bonus, and M2, comprising those whose data was better fit by the model with the subgoal bonus. Out of 63 participants, 42 were in the M1 group and 21 in the M2 group, indicating that approximately one-third of the sample showed signs of subgoal generalization, as evidenced by a reward-independent subgoal boost (Fig. 3.3).

In our subsequent analysis, we assessed the performance of both M1 and M2 groups during each phase of the three tests. For the M1 group during the implicit test phase, there was no discernible influence of subgoals on their decision-making. This was evident from their lack of preference for choices leading to subgoals, with their selection probabilities not exceeding what would be predicted by random chance (t(41) - .86, p = .39, Fig. 3.6B).

They did not show preference for subgoal dimension during the preference task, performing at levels comparable to guessing (t(41) = -2.16, p = 0.98). Additionally, only 24 out of 42 (57%) participants were able to identify the correct token dimension when asked explicitly (Fig. 3.6A).

In contrast, participants in the M2 group displayed a clear pattern of subgoal generalization (post-reversal accuracy difference from chance t(20) = 6.03, p = 6.76e-06; post-reversal accuracy difference from points condition: t(20) = 1.78, p =0.9, Fig. 3.6C). They favored options that led to subgoal outcomes and were able to adapt their decisions following changes in which options were correct. Furthermore, their performance in preference test was significantly better than chance level (t(20) = 4.79, p = .0001, Fig. 3.6B), and compared to M1 group (t(60) = 4.69, p = 1.57e-05). Furthermore, 17 out of 21 (80%)were able to explicitly identify the correct subgoal dimension based on the final question (Fig. 3.6A).

Finally, we compared two groups of participants based on the number of correct responses in the final question. We found that at the 5% level of significance, there is sufficient evidence to conclude that a larger proportion of M2 group participants identified a correct subgoal compared to M1 group ($z = 1.86, p = .03$, Fig. 3.6A). This result implies that generalization of subgoals between tasks might be contingent on explicit recognition of subgoal features.

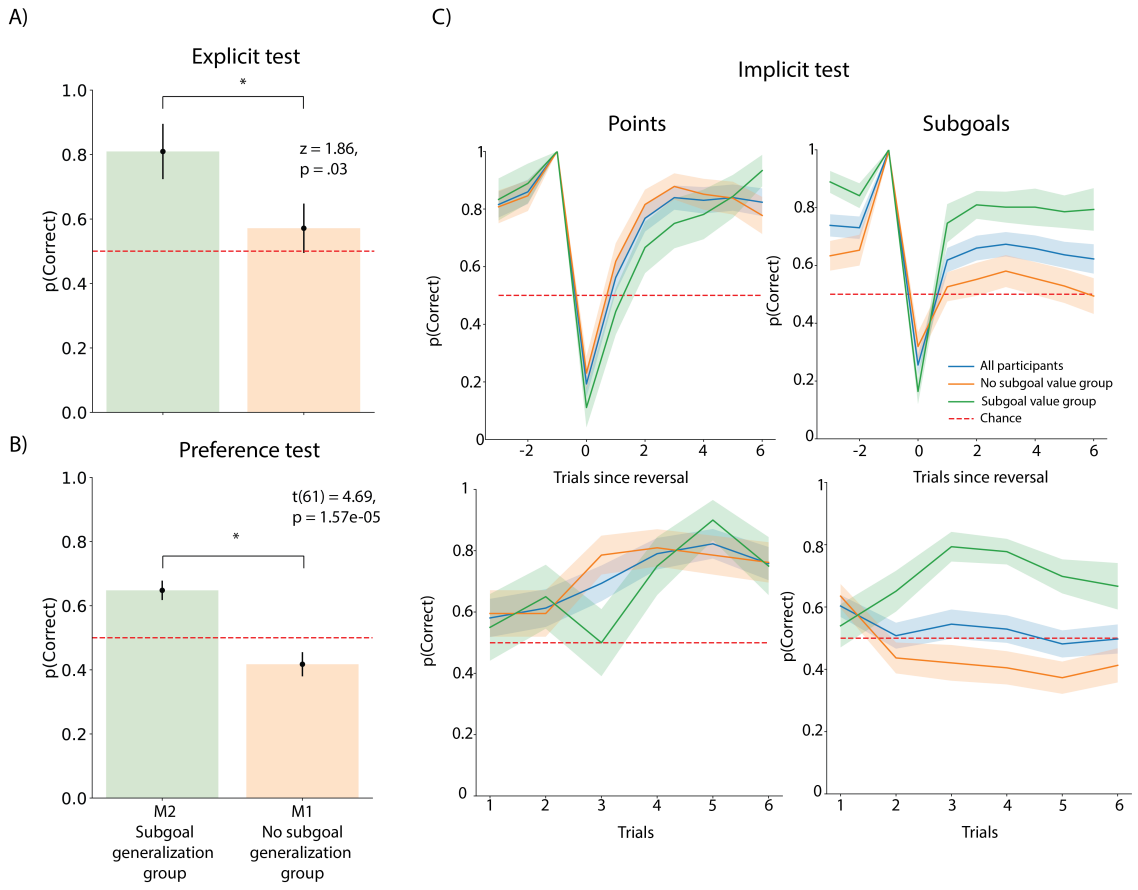


Figure 3.6: A) Participants who showed evidence of subgoal generalization based on modeling (M2 group) were able to identify subgoal features more frequently than expected by chance, and compared to participants who showed no evidence of subgoal generalization (M1 group). B) Participants in M2 group were also more likely than those M1 group to show preference for token dimensions that previously signaled subgoals, and C) displayed patterns of behavior consistent with aligning their choices to produce previous subgoals as outcomes.

We examined the training data further to identify differences in behavioral patterns between participants in the M1 and M2 groups, aiming to understand how these differences might influence the generalization of subgoals. However, our analysis did not revealed significant differences in the probability of subgoal identification among participants or in how various subgoal models affected their accuracy and sequence repetition (Supplementary Fig. 3.11).

Overall, our findings suggest that subgoals can influence decision-making through mechanisms that are distinct from external rewards or the element of surprise. However, it's

important to note that the generalization of subgoals (an important subgoal property) was observed in only a subset of our sample, seeming to be contingent on explicit recognition of subgoal features.

3.5 Discussion

Subgoals enable structured problem solving, and are essential for supporting learning in the absence of immediate feedback, by signaling what information is meaningful and should be encoded (Baldassarre and Mirolli, 2013; Chentanez et al., 2004; Eckstein and Collins, 2021; Singh et al., 2010). However, the nature of pseudoreinforcing effect of subgoals is unclear, as most previous research has conflated subgoals with rewards (by having subgoals always be experienced in instances where rewards are observed, McGovern and Barto, 2001) and general factors that drive curiosity/intrinsic motivation (Chentanez et al., 2004; Eckstein and Collins, 2021; Singh et al., 2010). Thus, it is not clear whether/how the subgoal effects on behavior would manifest once stripped of all of these components. In this project, we aimed to decouple subgoals from rewards and surprise/novelty in order to test a more isolated effect of subgoals on learning, and whether these subgoals subsequently carry reinforcing properties to a separate set of tasks (test phases). Our results replicated previous results (Eckstein and Collins, 2021) by confirming that participants were sensitive to hierarchical representations in the task, reflected in the composition of action sequences. Furthermore, we found that in the sample of participants who showed evidence of engaging in the task many inferred correct subgoals during the learning phase - evidence of isolated pseudoreinforcing subgoal effect on learning. However, a large proportion of this sample did not also show evidence of generalizing subgoals outside of the context in which they're acquired - suggesting that perhaps some of the properties our experimental manipulation stripped away from subgoals may be important for their robust generalization.

It is important to note that our task was very challenging; on each trial participants had a high number of possible individual-action and action-sequence combinations to explore. As a function of this, it is not surprising that many of the participants did not infer a correct 4-key sequence until later blocks (if at all). This impacted our data in two major ways: 1) high exclusion rate, and 2) of the participants whose data was included many took multiple blocks to solve the task, but their patterned exploration of sequences disqualified them from exclusion criteria, as this implied possible strategic behavior and not noise. This potentially made subgoal inference less robust, resulting in a fairly limited number of participants who showed evidence of subgoal generalization. It is also noteworthy that we could not have made the task more simple, as further simplifying the action sequences or number of subgoal dimensions would have trivialized the hierarchy in our task.

Our results pose an interesting question about the extent to which subgoal learning inference is deliberate or implicit. We have designed different test phases to help us assess the explicit bias for subgoals (e.g. preference test and final question) and more how previously

learned subgoals might implicitly guide action selection in a novel task (implicit test). We found that participants who showed the implicit effect of subgoals on action selection also displayed the bias towards subgoals in the preference test, and were able to correctly identify the correct subgoal dimension in the final question. This suggests that the ability to generalize the subgoals between different tasks (including the ones where subgoals are treated as new outcomes) might be contingent on whether they are explicitly recognized as subgoals. Furthermore, subgoal inference may be incidental, since our task can in theory be solved without paying any attention to the subgoals (although we did implement occasional probe checks to ensure participants were paying attention to the tokens, we cannot determine with certainty the extent to which subgoals were actively intentionally or incidentally learned).

An important limitation of our results is that only one third of the sample showed effect of generalization. We attempted to trace the differences in generalization to potential differences in behavioral pattern in the learning phase. Our follow-up analyses did not reveal any major differences between the two groups of participants (Supplementary Fig. 3.11), providing no clear explanation as to why some participants were able to generalize subgoals and some were not. It is possible that generalization variability may be explained by some aspects of our challenging task that the performance analyses we designed don't capture or are not sensitive to (such as a test which would probe explicit/implicit recognition during learning).

While we aimed to render the pseudoreinforcing effect of subgoals independent from reward association and factors that may drive learning purely due to an unexpected occurrence (novelty/surprise) to test the extent to which it can be isolated, it is very likely that these factors do affect what people perceive to be subgoals, and how they are used to decompose tasks (Diuk et al., 2013; Solway et al., 2014; Tomov et al., 2021). It will be extremely valuabe for the future work to systematically vary the factors of reward predictiveness, surprise, and structural frequency to evaluate which one of these is more critical do defining subgoals.

The concept of subgoals is valuable for constructing a formal theory of how humans learn to solve complex tasks, and generalize behavior - features which are still challenging to achieve for artificial agents. Our results suggest that subgoal effects on behavior can be achieved independently from reward association and surprise; nevertheless, despite the fact that overall participants showed inference of subgoals only a subset of participants engaged in subgoal generalization. Future work will be required to construct a more careful experimental design that might be able to identify learning patterns that may permit generalization from those that do not.

3.6 Supplementary Materials

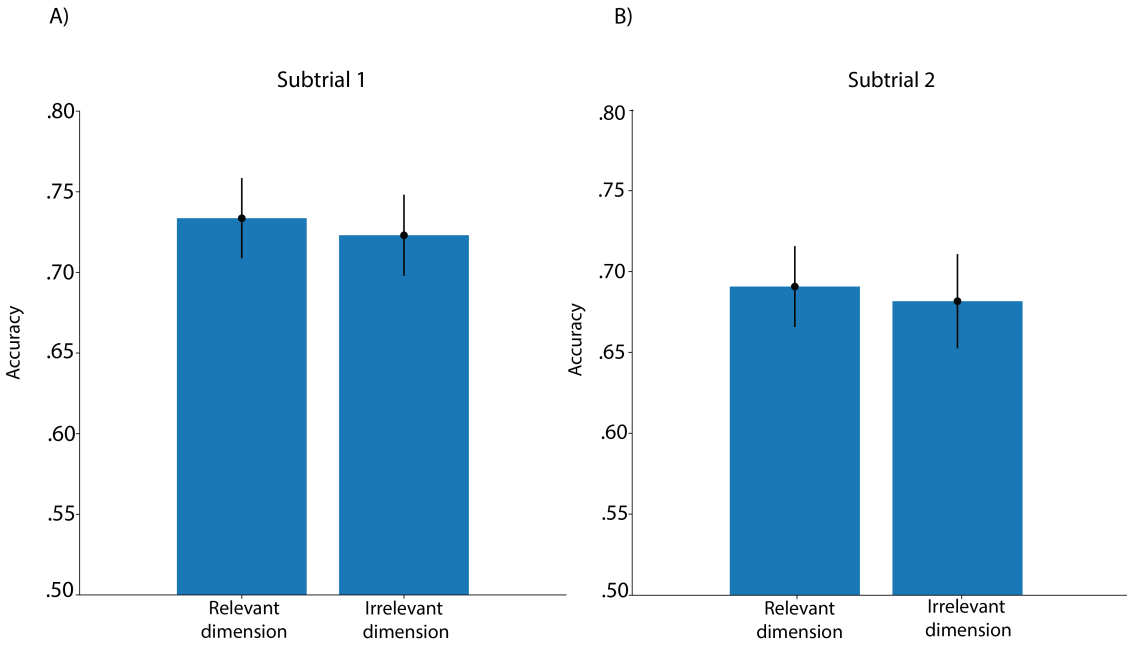


Figure 3.7: Participants' accuracy on probe trials, evaluated by whether they identified irrelevant/relevant dimension of the token they observed on the most recent trial on A) subtrial 1 and subtrial 2. Chance level is 1/3.

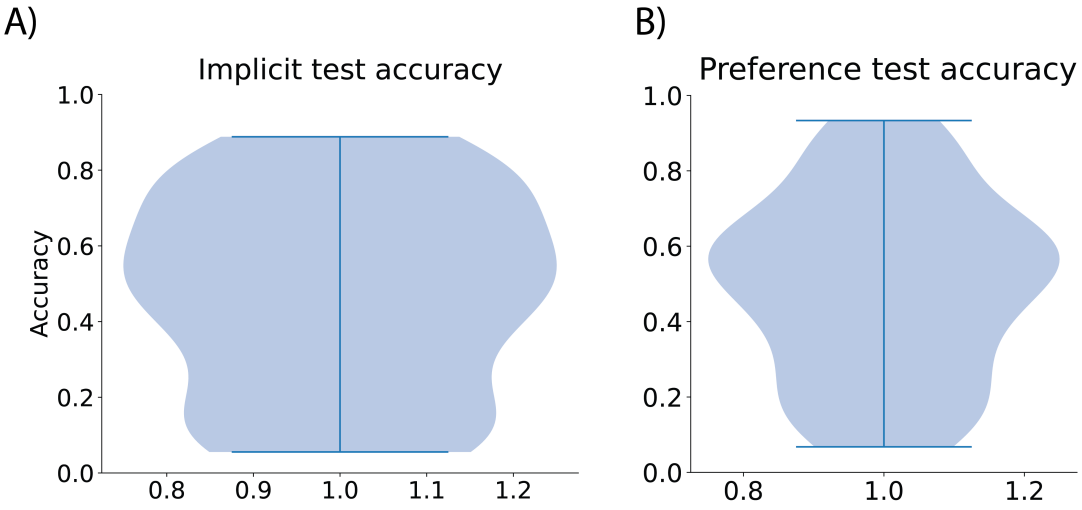


Figure 3.8: Distribution of performance in A) implicit and B) preference test.

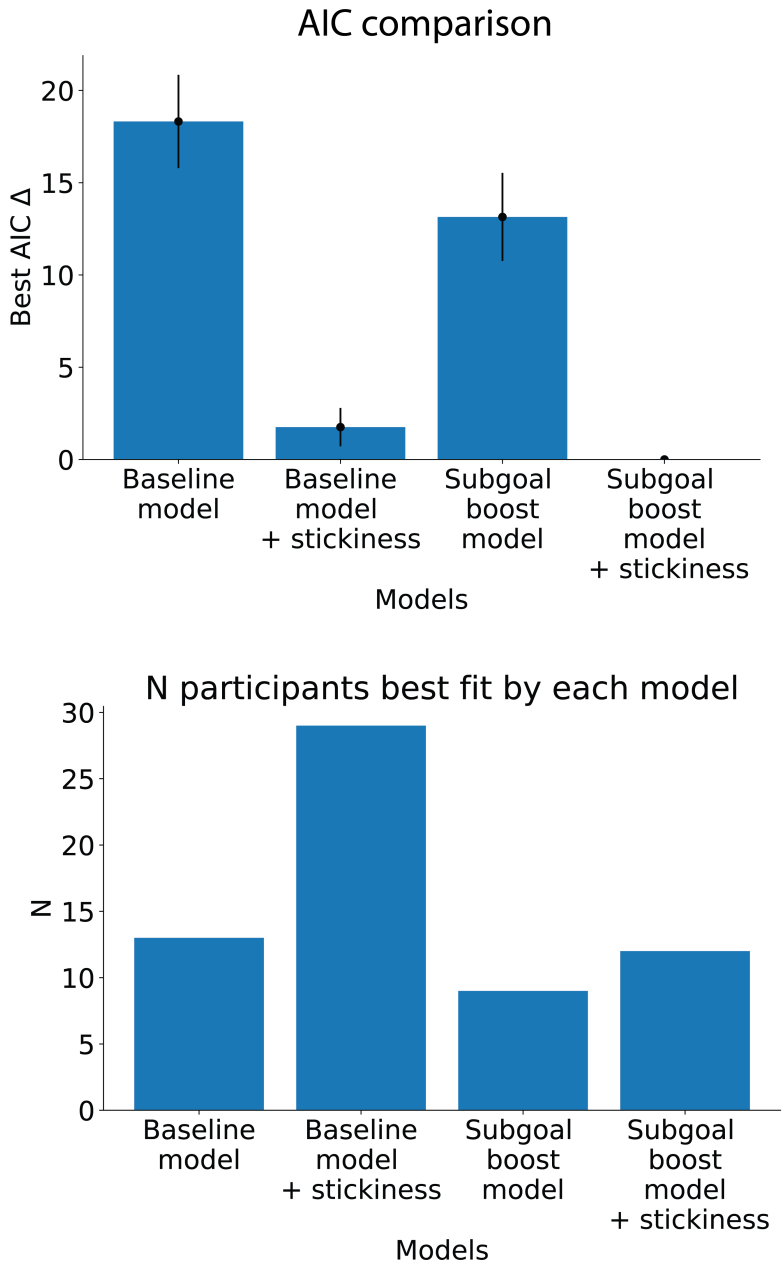


Figure 3.9: Model comparison based on AIC scores (top) and number of participants best fit by each of the models (bottom).

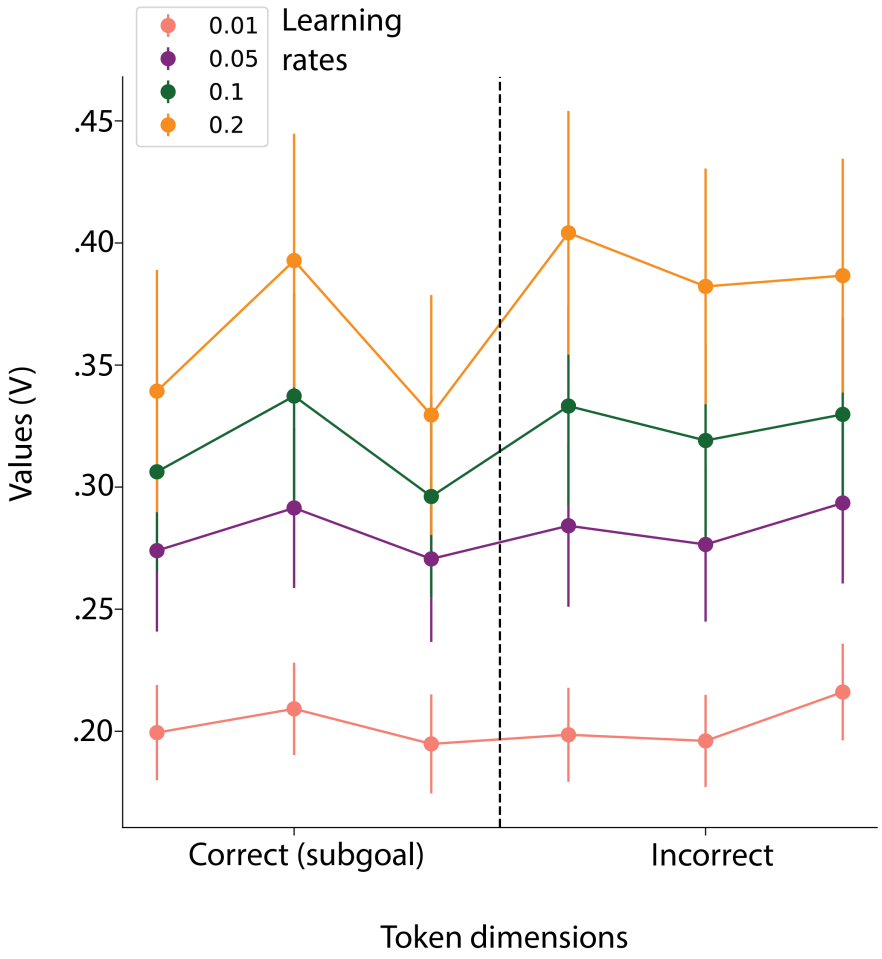


Figure 3.10: Model comparison based on AIC scores (top) and number of participants best fit by each of the models (bottom).

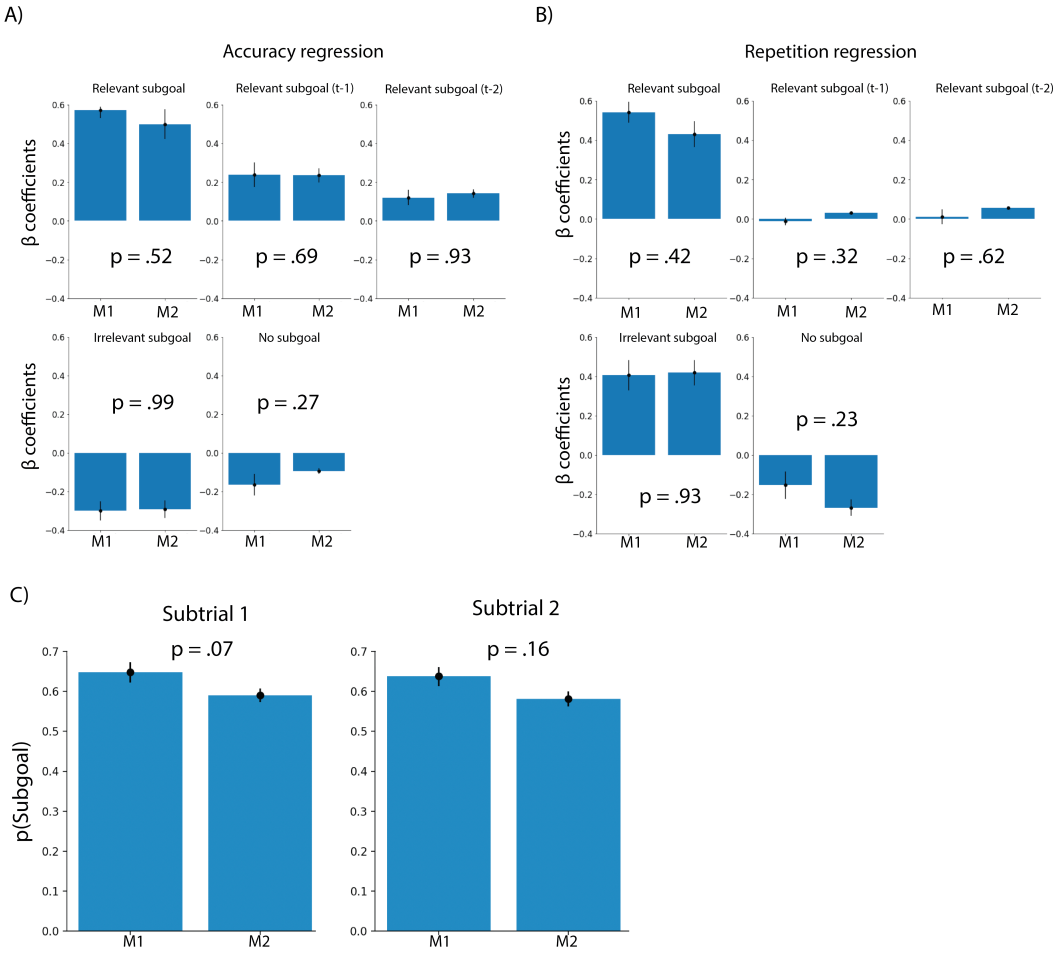


Figure 3.11: Group differences in training phase performance between participants in M1 and M2 groups. Two groups did not differ significantly in subgoals' impact on A) accuracy, B) sequence repetition or C) overall probability of selecting the subsequence that led to subgoals on 2 subtrials. P-values are based on Mann-Whitney U tests.

Chapter 4

Artificial neural networks as tools for fitting cognitive models

(Currently under review at PLOS Computational Biology as: Rmus, M., Pan, T., Xia, L. & Collins, A. G. E. (2024). Artificial neural networks for model identification and parameter estimation in computational cognitive models)

4.1 Abstract

Computational cognitive models have been used extensively to formalize cognitive processes. Model parameters offer a simple way to quantify individual differences in how humans process information. Similarly, model comparison allows researchers to identify which theories, embedded in different models, provide the best accounts of the data. Cognitive modeling uses statistical tools to quantitatively relate models to data that often rely on computing/estimating the likelihood of the data under the model. However, this likelihood is computationally intractable for a substantial number of models. These relevant models may embody reasonable theories of cognition, but are often under-explored due to the limited range of tools available to relate them to data. We contribute to filling this gap in a simple way using artificial neural networks (ANNs) to map data directly onto model identity and parameters, bypassing the likelihood estimation. We test our instantiation of an ANN as a cognitive model fitting tool on classes of cognitive models with strong inter-trial dependencies (such as reinforcement learning models), which offer unique challenges to most methods. We show that we can adequately perform both parameter estimation and model identification using our ANN approach, including for models that cannot be fit using traditional likelihood-based methods. We further discuss our work in the context of the ongoing research leveraging simulation-based approaches to parameter estimation and model identification, and how these approaches broaden the class of cognitive models researchers can quantitatively investigate.

4.2 Introduction

Computational modeling is an important tool for studying behavior, cognition, and neural processes. Computational cognitive models translate scientific theories into algorithms using simple equations with a small number of interpretable parameters to make predictions about the cognitive or neural processes that underlie observable behavioral or neural measures. These models have been widely used to test different theories about cognitive processes that shape behavior and relate to neural mechanisms (Lee and Webb, 2005; Montague et al., 2012; Palminteri et al., 2017; Shultz, 2003). By specifying model equations, researchers can inject different theoretical assumptions into most models, and simulate synthetic data to make predictions and compare against observed behavior. Researchers can quantitatively arbitrate between different theories by comparing goodness of fit (Akaike, 1998, Wei and Jiang, 2022) across different models. Furthermore, by estimating model parameters that quantify underlying cognitive processes, researchers have been able to characterize important individual differences (e.g. developmental: Eppinger et al., 2013; Hauser et al., 2015; Nussenbaum et al., 2022; Rmus et al., 2023; clinical: C. Chen et al., 2015; Collins et al., 2014; Gillan et al., 2016; Peterson et al., 2009; Zou et al., 2022) as well as condition effects (Sheynin et al., 2015; Weber et al., 2022).

Researchers' ability to benefit from computational modeling crucially depends on the availability of methods for model fitting and comparison. Such tools are available for a large group of cognitive models (such as, for example, reinforcement learning and drift diffusion models). Examples of commonly used model parameter fitting tools include maximum likelihood estimation (MLE, Myung, 2003), maximum a-posteriori (MAP, Cousineau and Helie, 2013), and sampling approaches (Baribault and Collins, 2023; Lee, 2011).Examples of model comparison tools include information criteria such as AIC and BIC (Akaike, 1998; Schwarz, 1978), and Bayesian group level approaches, including protected exceedance probability (Piray et al., 2019; Rigoux et al., 2014). These methods all have one important thing in common - they necessitate computing the likelihood of the data conditioned on models and parameters, thus limiting their use to models with tractable likelihood. However, many models do not have a tractable likelihood. This severely limits the types of inferences researchers can make about cognitive processes, as many models with intractable likelihood might offer better theoretical accounts of the observed data. Examples of such models include cases where observed data (e.g. choices) might depend on latent variables - such as the unobserved rules that govern the choices (Eckstein and Collins, 2020; Frank and Badre, 2012; Solway et al., 2014), or a latent state of engagement (e.g. attentive/distracted, Ashwood et al., 2022; Findling et al., 2021) a participant/agent might be in during the task. In these cases, computing the likelihood of the data often demands integrating over the latent variables (rules/states) across all trials, which grows exponentially and thus is computationally intractable. This highlights an important challenge - computing likelihoods is essential for estimating model parameters, and performing fitness comparison/model identification, and alternative models are less likely to be considered or taken advantage of to a greater extent.

Some existing techniques attempt to bridge this gap. For example, Inverse Binomial Sampling (van Opheusden et al., 2020), particle filtering (Djuric et al., 2003), and assumed density estimation (Minka, 2013) provide approximate solutions to the Bayesian inference process in specific cases. Many of these methods, however, require advanced mathematical expertise for effective use and adaptation beyond specific cases they were developed for, making them less accessible many researchers. Approximate Bayesian Computation (ABC, Lintusaari et al., 2017; Palestro et al., 2018; Sunnåker et al., 2013; Turner and Sederberg, 2014; Turner et al., 2013) offers a more accessible avenue for estimating parameters in models limited by intractable likelihoods. More widely employed in cognitive modeling, the approach of basic ABC rejection algorithms involves translating trial-level data into summary statistics. Parameter values of the candidate model are then selected based on their ability to produce simulated data that is closely aligned with summarized data, guided by some predefined rejection criterion.

While ABC rejection algorithms provide a useful workaround solution, it's important to acknowledge their inherent limitations. Specifically, ABC results are sensitive to the choice of summary statistics (and rejection criteria) and sample efficiency of ABC demonstrates scales poorly in cases of high-dimensional data (Cranmer et al., 2020; Lavin et al., 2021; Sunnåker et al., 2013). Recent strides in the field of simulation-based inference/likelihood-free inference have addressed these limitations by using artificial neural network(ANN) structures designed to optimize summary statistics, and consequently infer parameters. These methods enable automated (or semi-automated) construction of summary statistics, minimizing the effect the choice of summary statistics may have on the accuracy of parameter estimation (Y. Chen et al., 2020; Fearnhead and Prangle, 2012; Jiang et al., 2017; Lavin et al., 2021; Radev, Mertens, et al., 2020; Radev, Voss, et al., 2020). This innovative approach serves to amortize the computational cost of simulation-based inference, opening new frontiers in terms of scalability and performance (Boelts et al., 2022; Fengler et al., 2021; Ghaderi-Kangavari et al., 2023; Radev, Mertens, et al., 2020; Radev, Voss, et al., 2020; Radev et al., 2021; Schmitt et al., 2021; Sokratous et al., 2023).

Here, we test a related, general approach that leverage advances in artificial neural networks (ANNs) to estimate parameters and perform model identification for models with and without tractable likelihood, entirely bypassing the likelihood estimation (or approximation) step. ANNs have been successfully used to fit intractable models in different fields, including weather models (Lenzi et al., 2023) and econometric models (Wei and Jiang, 2022), and more recently cognitive models of decision making Radev, Mertens, et al., 2020; Radev, Voss, et al., 2020. We develop similar approaches to specifically target the intractability estimation problem in the field of computational cognitive science, including both parameter estimation and model identification, and thoroughly test it in a challenging class of models where there are strong dependencies between trials (e.g. learning experiments).

Our approach relies on the property of ANNs as universal function approximators. The ANN structure we implemented was a recurrent neural network (RNN) with feed-forward layers inspired by Dezfouli et al., 2019 (Fig: 4.1) that is trained to estimate model parame-

ters, or identify which model most likely generated the data based on input data sequences simulated by the cognitive model. Our approach is similar to previous work in the domain of simulation-based inference (Radev, Mertens, et al., 2020; Radev, Voss, et al., 2020), with a difference that such architectures are specifically designed to optimize explicit summary statistics that describe the data patterns (e.g. invertible networks). Here, rather than emphasizing steps involving the reduction of data dimensionality through the creation (and selection) of summary statistic vectors and subsequent inference based on parameter value samples, our focus is on the direct translation of raw data sequences into precise parameter estimates or the identification of the source model (via implicit summary statistics in network layers).

To validate and benchmark our approach, we first compared it against standard model parameter fitting methods most commonly used by cognitive researchers (MLE, MAP, rejection ABC) in cognitive models from different families (reinforcement learning, Bayesian Inference) with tractable likelihood. Next, we demonstrated that neural networks can be used for parameter estimation of models with intractable likelihood, and compared it to standard approximation method (ABC). Finally, we showed that our approach can also be used for model identification. Our results showed that our method is highly successful and robust at parameter and model identification while remaining technically lightweight and accessible. We highlight the fact that our method can be applied to standard cognitive data sets (i.e. with arbitrarily small number of participants, and normal number of trials per participant), as the ANN training is fully done on a large simulated data set. Our work contributes to the ongoing research focusing on leveraging artificial neural networks to advance the field of computational modeling, and provides multiple new avenues for maximizing the utility of computational cognitive models.

4.3 Results

We focused on two distinct artificial neural network (ANNs) applications in cognitive modeling: parameter estimation and model identification. Specifically, we built a network with a structure suitable for sequential data/data with time dependencies (e.g. recurrent neural network (RNN); Dezfouli et al., 2019). Training deep ANNs requires large training data sets. We generated such a data set at minimal cost by simulating a cognitive computational model on a cognitive task a large number of times. Model behavior in the cognitive task (e.g. a few hundred trials of stimulus-action pairs or stimulus-action-outcome triplets (depending on the task) for each simulated agent) constituted ANN's training input; true known parameter values (or identity of the model) from which the data was simulated constituted ANNs' training targets. We evaluated the network's training performance in predicting parameter values/identity of the model in a separate validation set, and tested the trained network on a held out test set. We tested RNN variants and compared their accuracy against traditional likelihood-based model fitting/identification methods using both likelihood-tractable

and likelihood-intractable cognitive models. See Methods section for details on the ANN training and testing process.

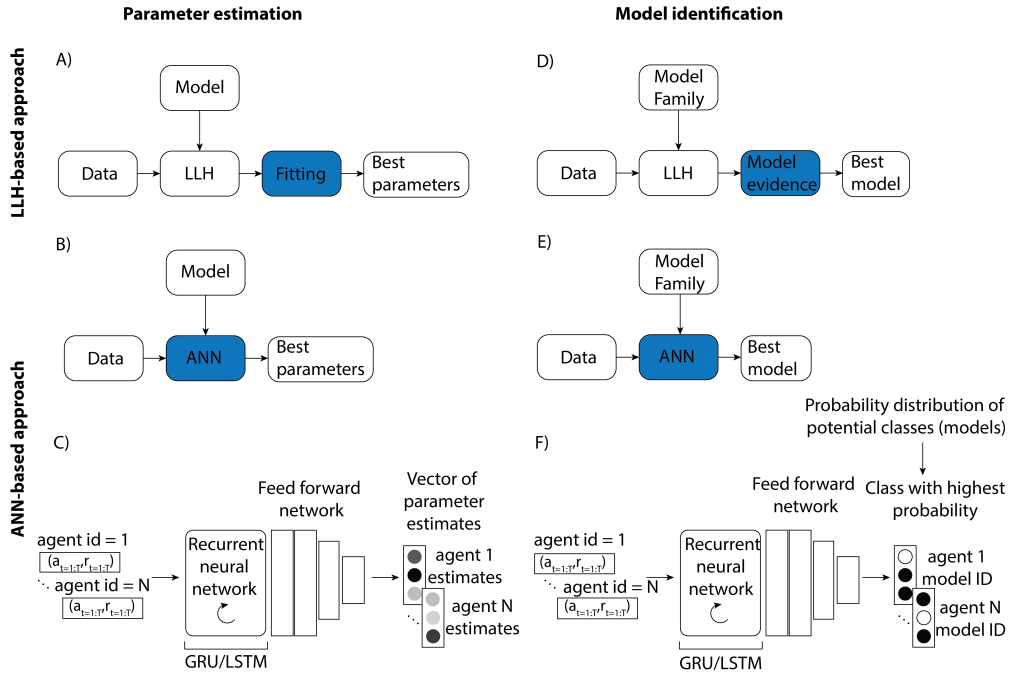


Figure 4.1: Artificial neural network (ANN) approach. A) Traditional methods rely on computing log-likelihood (LLH) of the data under the given model, and optimizing the likelihood to derive model parameter estimates. B) The ANN is trained to map parameter values onto data sequences using a large simulated data set; the trained network can then be used to estimate cognitive model parameters based on new data without the need to compute or approximate likelihood. C) The ANN structure inspired by Dezfouli et al., 2019 is suitable for data with strong inter-trial dependencies: it consists of an RNN and fully connected feed-forward network, with an output containing ANN estimates of parameter values the data was simulated from for each agent. D) As in parameter estimation, traditional tools for model identification rely on likelihood to derive model comparison metrics (e.g. AIC, BIC) that are used to determine which model likely generated the data. E) ANN is instead trained to learn the mapping between data sequences and respective cognitive models the data was simulated from. F) Structure of the ANN follows the structure introduced for parameter estimation, with the key difference of final layer containing the probability distribution over classes representing model candidates, with highest probability class corresponding to the model the network identified as the one that likely generated the agent's data.

Parameter recovery

Benchmark comparison

First, we sought to validate our ANN method and compare its performance to existing methods by testing it on standard likelihood-tractable cognitive models of different levels of complexity in the same task: 2-parameter ($2P-RL$) and 4-parameter ($4P-RL$) reinforcement learning models commonly used to model behavior on reversal tasks (Gläscher et al., 2009; Hampton et al., 2006; Hauser et al., 2015; Peterson et al., 2009), as well as Bayesian Inference model (BI) and Bayesian Inference with Stickiness ($S-BI$) as an alternative model family that has been found to outperform RL in some cases (Costa et al., 2015; Perfors et al., 2011; Särkkä and Svensson, 2023). We estimated model parameters using multiple traditional methods for computing (maximum likelihood and maximum a-posteriori estimation; MLE and MAP) and approximating (Approximate Bayesian Computation; ABC) likelihood. We used the results of these tools as a benchmark for evaluating the neural network approach. Next, we estimated parameters of these models using two variants of RNNs: with gated recurrent units (GRUs) or Long-Short-Term-Memory units (LSTM).

We used the same held out data set to evaluate all methods (the test set the ANN has not observed yet, see simulation details). For each of the methods we extracted the best fit parameters, and then quantitatively estimated the method's performance as the mean squared error (MSE) between estimated and true parameters across all agents. Methods with lower MSE indicated better relative performance. All of the parameters were scaled for the purpose of loss computation, to ensure comparable contribution to loss across different parameters. To quantify overall loss for a cognitive model we averaged across all individual parameter MSE scores; to calculate fitting method's MSE score for a class of cognitive models (e.g. likelihood tractable models) we averaged across respective method's MSE scores for those models (See Methods for details about method evaluation).

First, we examined the performance of standard model-fitting tools (MLE, MAP and ABC). The standard tools yielded a pattern of results that are expected based on noisy, realistic-size data sets (with several-hundred trials per agent). Specifically, we found that MAP outperformed MLE (Fig. 4.2A, average MSEs: $MLE = .67, MAP = .35$), since the parameter prior applied in MAP regularizes the fitting process. ABC was also worse compared to MAP (Fig. 4.2A, average MSE: $ABC = .53$). While fitting process is also regularized in ABC, worse performance in some models can be attributed to signal loss that arises from approximation to the likelihood. Next, we focused on the ANN performance; our results showed that for each of the models, ANN performed better than or just as well as the traditional methods (Fig. 4.2A, average MSEs for different RNN variants: $GRU = .32, LSTM = .35$). Better network performance was more evident for parameter estimation in more complex models (e.g. models with higher number of parameters such as 4P-RL and S-BI; average MSE across these 2 models: $MLE = .95, MAP = .43, ABC = .71, GRU = .38, LSTM = .44$).

Next, we visualized parameter recovery. We found that for each of the cognitive models the parameter recovery was largely successful (Spearman ρ correlations between true parameter values and estimated values: β $\rho_{MAP}, \rho_{GRU} = [.90, .91]$, α^+ $\rho_{MAP}, \rho_{GRU} = [.53, .52]$, α^- $\rho_{MAP}, \rho_{GRU} = [.88, .89]$, κ: $\rho_{MAP}, \rho_{GRU} = [.78, .79]$, Fig. 4.2B; all correlations were significant at $p < .001$). For conciseness, we only show recovery of the more complex model parameters from the RL model family (and only MAP method as it performed better compared to ABC and MLE, as well as only GRU since it performed better than LSTM), as we would expect a more complex model to emphasize superiority of a fitting method more clearly compared to simpler models. Recovery plots of the remaining models (and respective fitting methods) can be found in supplementary materials. Our results suggest that 1) ANN performed as well as traditional methods in parameter estimation based on the MSE loss; 2) more complex models may limit accuracy of parameter estimation in traditional methods that neural networks appear to be more robust against. We note that for the $4P-RL$ model, parameter recovery was noisy for all methods, with some parameters being less recoverable than others (e.g. α^+ , Fig. 4.2B). This is an expected property of cognitive models applied to realistic-sized experimental data as found in most human experiments (i.e. a few hundred trials per participant). To check whether the limited recovery can be attributed to parameter identifiability rather than pitfalls of any specific method, we looked at the correlation between parameter estimates obtained using the standard model fitting method (MAP) and the ANN (GRU) (Fig. 4.16) - with parameters that are not well recovered (e.g. α^+ in 4P-RL model) being of particular interest. High correlation between estimated parameters obtained via 2 methods imply systematic errors in parameter identification that apply to both methods - thus suggesting that the weaker correlation between true and fit parameters for some parameters is more likely due to limitations in the model applied to the data set than method specifications such as poor optimization performance. We further discuss the implications in discussion section - highlighting that computational models should be carefully crafted and specified regardless of the tools used for model fitting.

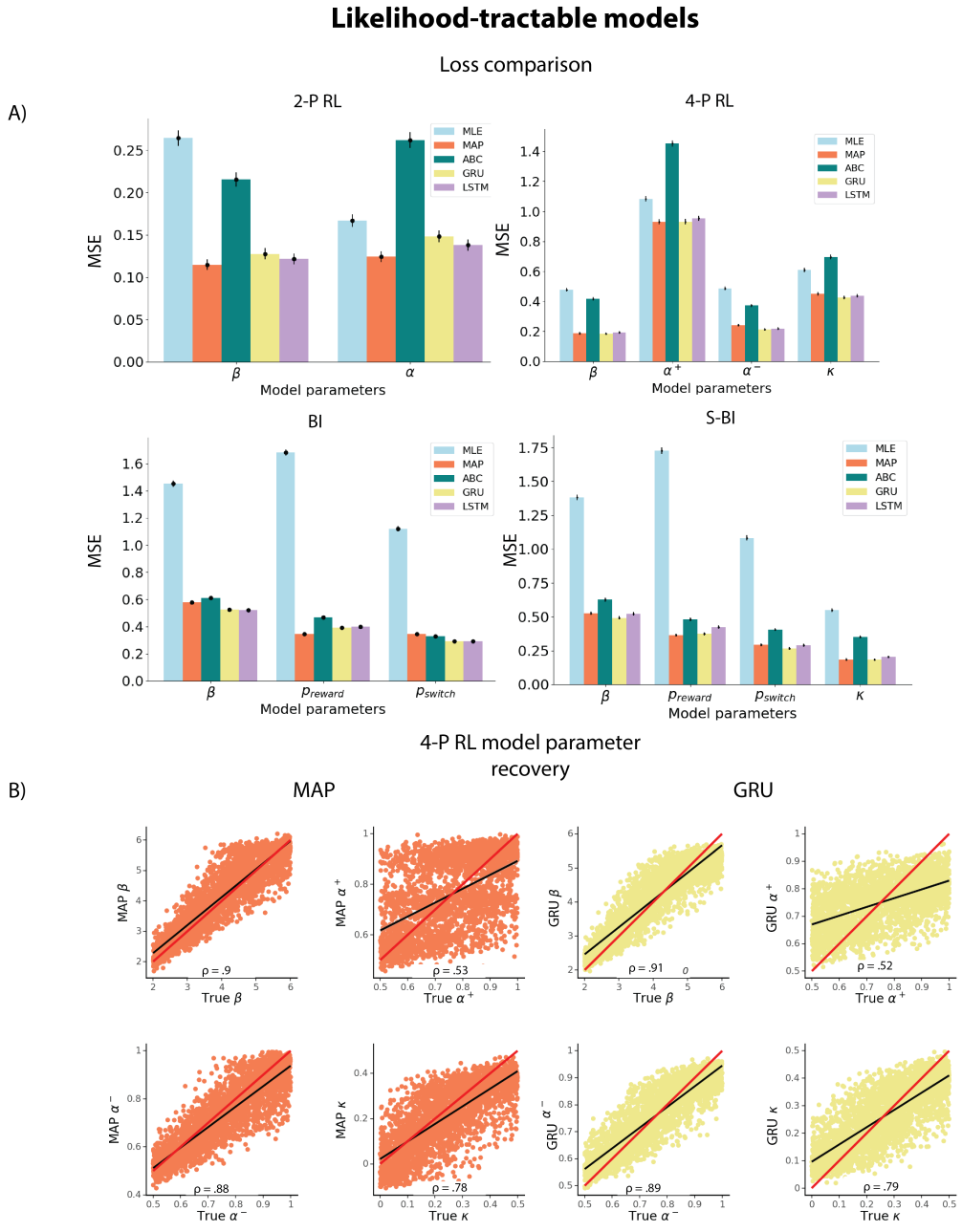


Figure 4.2: A) Parameter recovery loss from the held out test set for the tractable-likelihood models (2P-RL, 4P-RL, BI, S-BI) using each of the tested methods. Loss is quantified as the mean squared error (MSE) based on the discrepancy between true and estimated parameters. Bars represent loss average for each parameter across all agents, with errorbars representing standard error across agents. B) Parameter recovery from the 4P-RL model using MAP and GRU. ρ values represent Spearman ρ correlation between true and estimated parameters. Red line represents a unity line ($x = y$) and black line represents a least squares regression line. All correlations were significant at $p < .001$.

Testing in cognitive models with intractable likelihood

Next, we tested our method in two examples of computational models with intractable likelihood. As a comparison method, we implemented Approximate Bayesian Computation (ABC), alongside our ANN approach to estimate parameters. The two example likelihood-intractable models we used had in common the presence of a latent state which conditioned sequential updates: RL with latent attentive state ($RL-LAS$), and a form of non-temporal hierarchical reinforcement learning (HRL, Eckstein and Collins, 2020). Since we cannot fit these models using MAP or MLE we used only ABC as a benchmark. Because we found LSTM RNN to be more challenging to train and achieve similar results when compared to GRU, we focused on GRU for the remainder of comparisons. We found that average MSE was much lower for the neural network compared to ABC for both RL-LAS (Fig. 4.3A, average MSEs: $ABC = .62, GRU = .21$) and HRL (Fig. 4.3A, average MSEs: $ABC = .28, GRU = .19$). Spearman correlations were noisier for ABC compared to GRU in both models (Fig. 4.3B, **RL-LAS** : β $\rho_{ABC}, \rho_{GRU} = [.72, .91]$, α $\rho_{ABC}, \rho_{GRU} = [.83, .95]$, T $\rho_{ABC}, \rho_{GRU} = [.5, .81]$; **HRL** : β $\rho_{ABC}, \rho_{GRU} = [.86, .89]$, α $\rho_{ABC}, \rho_{GRU} = [.85, .9]$; all correlations were significant at $p < .001$). Furthermore, some parameters were less recoverable than others (e.g. the T parameter in RL-LAS model, which indexed how long participants remained in an inattentive state); this might be in part due to less straightforward effect of T on behavior; see supplementary materials (Fig. 4.12). Note that in order to obtain our ABC results we had to perform an extensive exploration procedure to select summary statistics - ensuring reasonable ABC results. Indeed, the choice of summary statistics is not trivial and represents an important difficulty of applying basic rejection ABC (Lavin et al., 2021; Sunnåker et al., 2013), that we can entirely bypass using our new neural network approach. We acknowledge that recent methods that rely on ANNs replaced standard ABC methods by automating (or semi-automating) construction of summary statistics (Y. Chen et al., 2020; Fearnhead and Prangle, 2012; Jiang et al., 2017; Lavin et al., 2021; Lenzi et al., 2023; Radev, Mertens, et al., 2020; Radev, Voss, et al., 2020). However, we aimed to explore an alternative approach, independent of explicit optimization of summary statistics, and focused on the ABC instantiation that has been most frequently implemented in the field of cognitive science as a benchmark(Sunnåker et al., 2013; Turner and Sederberg, 2014; Turner et al., 2013).

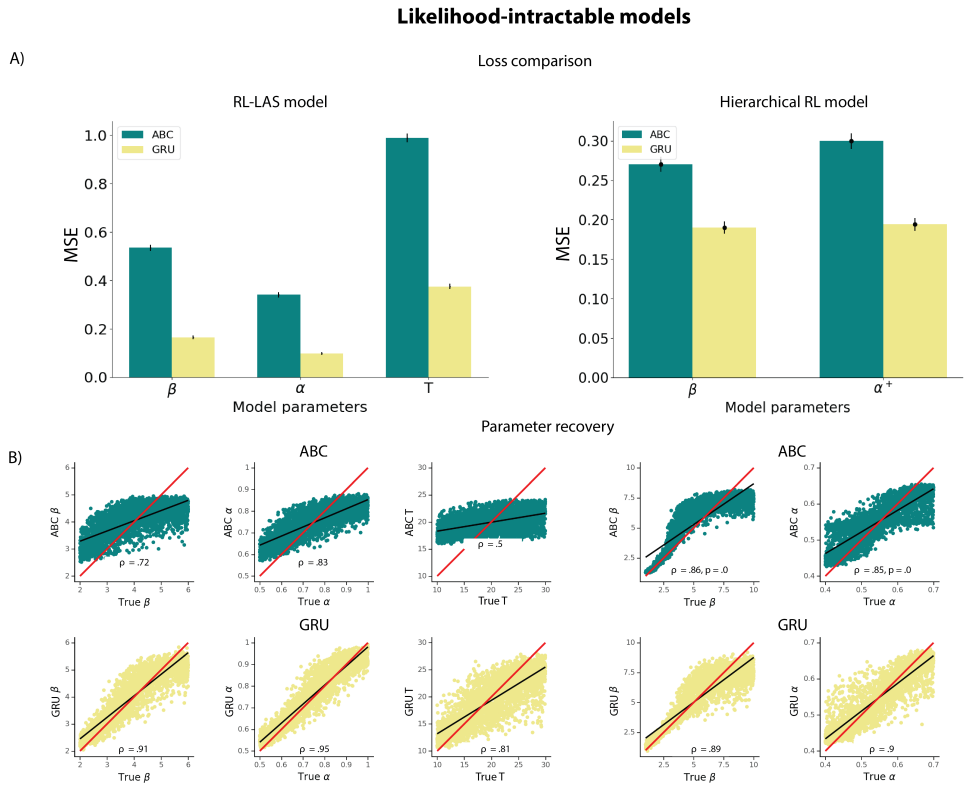


Figure 4.3: A) Parameter recovery loss from the held out test set for the intractable-likelihood models (RL-LAS, HRL) using ABC and GRU network. Loss is quantified as the mean squared error (MSE) based on the discrepancy between true and estimated parameters. Bars represent MSE average for each parameter across all agents, with errorbars representing standard error across agents; see supplementary (Fig. 4.23) for variability across seeds. B) Parameter recovery from the RL-LAS and HRL models using ABC (green) and GRU network (yellow). ρ values represent Spearman ρ correlation between true and estimated parameters. Red line represents a unity line ($x = y$) and black line represents a least squares regression line.All correlations were significant at $p < .001$.

Uncertainty of parameter estimates

Thus far, we have outlined a method that provides point estimates of parameters based on input data sequences, as is typically the use for much lightweight cognitive modeling (e.g. maximum likelihood estimation or MAP). However, it is sometimes also valuable to compute the uncertainty associated with these estimates (Lee, 2011). It is possible to extend our approach to add this capability. While there are various alternative ways to do so

(e.g. Bayesian neural networks), the approach we have opted for is incorporating evidential learning into our method (Amini et al., 2020). Evidential learning differs from Bayesian networks in that it places priors over likelihood function, rather than network weights. The network leverages this property to learn both statistical (aleatoric) and systematic (epistemic) uncertainty during the process of estimating a continuous target based on the input data sequences. This marks a shift from optimizing a network to minimize errors based on average prediction, without considering uncertainty.

We applied our method with integrated evidential learning to tractable and intractable versions of the RL models (2P-RL and RL-LAS, Fig. 4.4). We found that incorporating this modification did not compromise the point estimate parameter recovery (e.g. compared to our baseline method focused only on maximizing the accuracy of the predictions). Additionally, it enabled the estimation of the uncertainty around the point estimate, as demonstrated by Amini et al., 2020. This extension appears to be more computationally expensive (with longer training periods) than our original method, but not to a prohibitive extent.

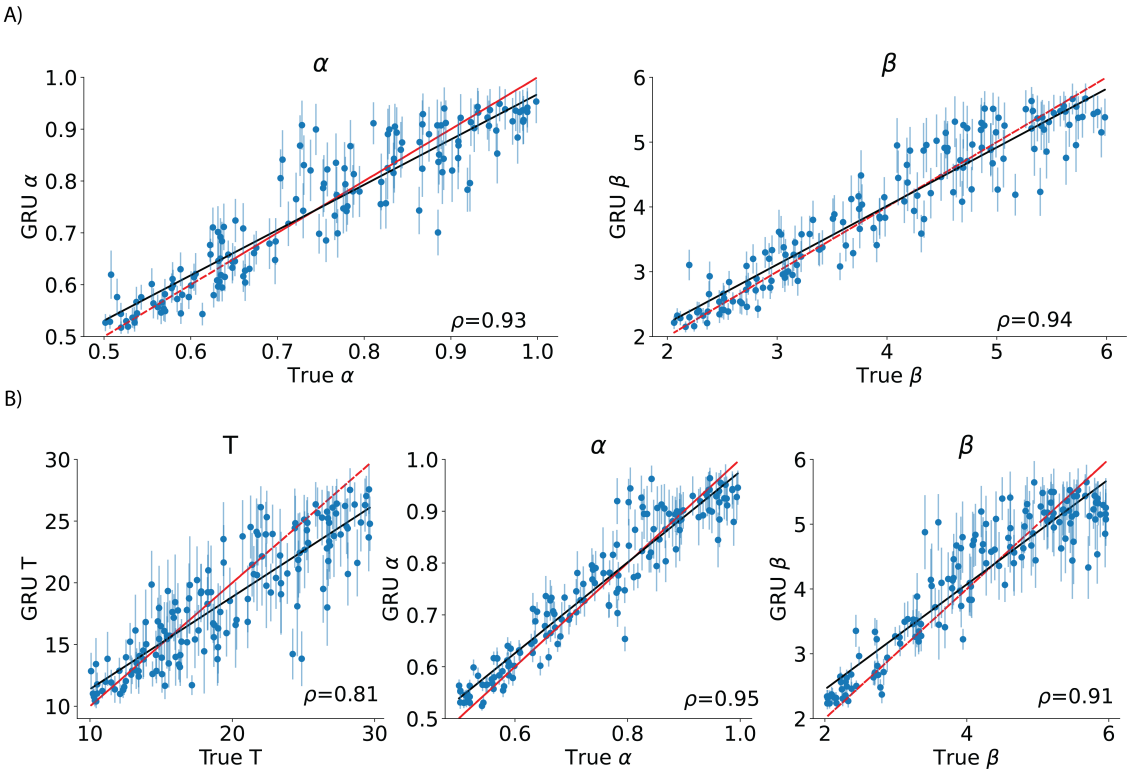


Figure 4.4: Using evidential learning to evaluate uncertainty of parameter estimates for A) 2-parameter RL model (tractable likelihood) and B) RL model with latent attention states (intractable likelihood). Vertical lines around point estimates illustrate model uncertainty. We are showing only 100 data points for the purpose of cleaner visualization, Spearman ρ values are computed based on the total number of agents in the held-out test data (3k).

Model identification

We also tested the use of our ANN approach for model identification. Specifically, we simulated data from different cognitive models, and trained the network to make a prediction regarding which model most likely generated the data out of all model candidates. The network architecture was identical to the network used for parameter estimation, except that the last layer became a classification layer (with one output unit per model category) instead of a regression layer (with one output unit per target parameter).

For models with tractable likelihood, we performed the same model identification process using AIC (Akaike, 1998) that relies on likelihood computation, penalized by number of parameters, to quantify model fitness as a benchmark. We note that another common criterion, BIC (Wei and Jiang, 2022), performed more poorly than AIC in our case.The best

fitting model is identified based on the lowest AIC score - a successful model recovery would indicate that the true model has the lowest AIC score compared to other models fit to that data. To construct the confusion matrix, we computed best AIC score proportions for all models, across all agents, for data sets simulated from each cognitive model (Fig: 4.5; see methods).

As shown in Figure 4.5A, model identification performed using our ANN approach was better compared to the AIC confusion matrix, with less "confusion" - lower off-diagonal proportions compared to diagonal proportions (correct identification). Model identification using AIC is likely in part less successful due to some models being nested in others (e.g. $2P - RL$ in $4P - RL$, BI in $S - BI$). Specifically, since AIC score represents a combination of likelihood and penalty incurred by the number of parameters it is possible that the data from more complex models is incorrectly identified as better fit by a simpler version of that model (e.g. the model with fewer parameters; an issue which would be more pronounced if we used a BIC confusion matrix). The same phenomenon is observed with the network, but to a much lesser extent, showing better identification out of sample - even for nested models. Furthermore, the higher degree of ANN misclassification observed for $BI/S - BI$ was driven by $S - BI$ simulations with stickiness parameter close to 0, which would render the BI and $S - BI$ non-distinguishable (Fig. 4.13).

Because we cannot compute the likelihood for our likelihood-intractable models based on closed-form solutions via MAP, we only report the confusion matrices obtained from our ANN approach In the first confusion matrix we performed model identification for $2P - RL$ and $RL - LAS$, as we reasoned these two models differ by only one mechanism (occasional inattentive state), and thus could potentially pose the biggest challenge to model identification. In the second confusion matrix, we included all models used to simulate data on the HRL task (*HRL model*, *Bayesian inference model*, *Bayesian inference with stickiness model*). In both cases, the network successfully identified the correct models as true models, with a very small degree of misidentification, mostly in the nested models. Based on our benchmark comparison to AIC, and the proof of concept identification for likelihood intractable models, our results indicate that the ANN can be leveraged as a valuable tool in model identification.

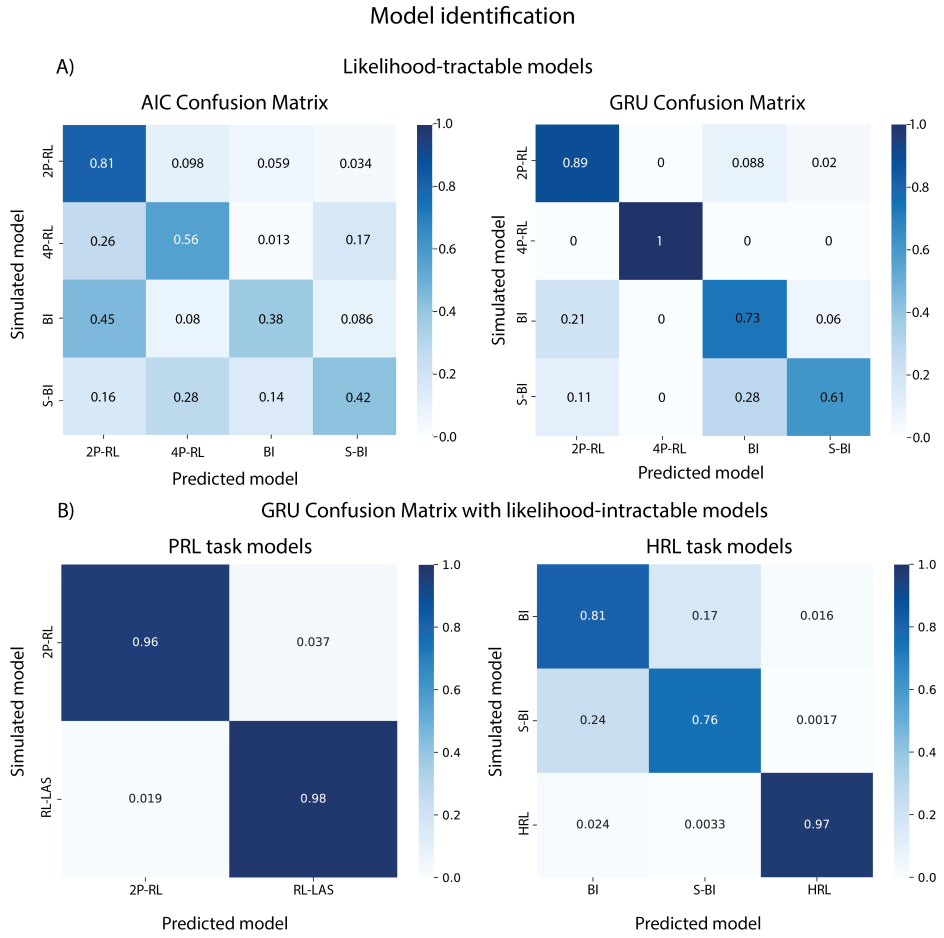


Figure 4.5: Model identification results.A) Confusion matrix of likelihood-tractable models from PRL task based on 1) likelihood/AIC metric, and 2) ANN identification. AIC confusion matrix revealed a much higher degree of misclassification (e.g. true simulated model being incorrectly identified as a different model). B) Confusion matrix of likelihood-intractable models using ANN (2P-RL and RL-LAS models were simulated on the PRL task; HRL, BI and S-BI models were simulated on the HRL task).

Robustness tests

Robustness tests: influence of different input trial sequence lengths

ANNs are sometimes known to fail catastrophically when data is different from the training distribution in minor ways (Liang et al., 2017; Moosavi-Dezfooli and Alhussein Fawzi, 2017; Nguyen et al., 2015; Szegedy et al., 2013). Thus, we investigated the robustness of our

method to differences in data format we might expect in empirical data, such as different numbers of trials across participants. Specifically, we conducted robustness experiments by varying the number of trials in each individual simulation contributing to training or test sets, fixing the number of agents in the training set.

To evaluate the quality of parameter recovery, we used the coefficient of determination score (R^2) which normalizes different parameter ranges. We found that the ANNs trained with a higher trial number reach high R^2 scores in long test trials. However, their performance suffers significantly with smaller number of test trials. The results also show a similar trend in model identification tasks except that training with higher trial number doesn't guarantee a better performance. For instance, the classification accuracy between HRL task models of the ANN trained with 300 trials reaches 87% while the ANN trained with 500 trials is 84%.

Data-augmentation practices in machine learning increase robustness of models during training (Shorten and Khoshgoftaar, 2019) by introducing different types of variability in the training data set (e.g. adding noise, different data sizes). Specifically, slicing time-series data into sub-series is a data-augmentation practice that increases accuracy (Iwana and Uchida, 2021). Thus, we trained our ANN with the fixed number of simulations of different trial numbers. As predicted, we found that the ANNs trained with a mixture of trial sequence lengths across simulations (purple line) consistently yielded better performance across different numbers of test trials for both parameter recovery and model identification (Fig. 4.6A,B).

Robustness tests: prior parameter assumptions

We also tested the effects of incorrect prior assumptions about the parameter range on method performance. Specifically we 1) trained the network using data simulated from a narrow range of parameters (theoretically informed) and 2) trained the network based on broader range of parameter values. Next, we tested both networks in making out-of-sample predictions for test data sets that were simulated from narrow and broad parameter ranges respectively. The network trained using a narrow parameter range made large errors at estimating parameters for data simulated outside of the range it was trained on; training the network on a broader range overall resulted in smaller error, with some loss of precision for the parameter values in range of most interest (e.g. the narrow range of parameters the alternative network is trained on). We observed similar results with MAP, where we specified narrow/broad prior (where narrow prior would place high density on a specific parameter range). Notably, training the network using a broader range of parameters while oversampling from a range of interest yielded more accurate parameter estimation compared to MAP with broad priors (Approach described in Fig. 4.15).

Robustness tests: model misspecification

In addition to testing the effects of incorrect priors, we also tested the effect of model misspecification on standard method and ANN performance (focusing on MAP and GRU network, as they performed the best in parameter recovery tests on benchmark models). We fit the Bayesian inference model (without stickiness) to the data simulated from the Bayesian inference model with stickiness using MAP. For the ANN, we trained the neural network to estimate parameters of the Bayesian inference model, and tested it on the separate test set data simulated from the Bayesian inference model with stickiness. For each method, we looked at the correlation between the ground truth Bayesian inference with stickiness model parameters, and the method's parameter estimates (Fig. 4.19). Our results suggest that the parameters shared between the 2 models are reasonably recoverable using both MAP and ANN (e.g. the recovery is noisier but comparable to that of parameters in Bayesian models without model misspecification (Figs. 4.10, 4.11)); furthermore, the correlation between ground truth and estimated values is similar for the two methods.

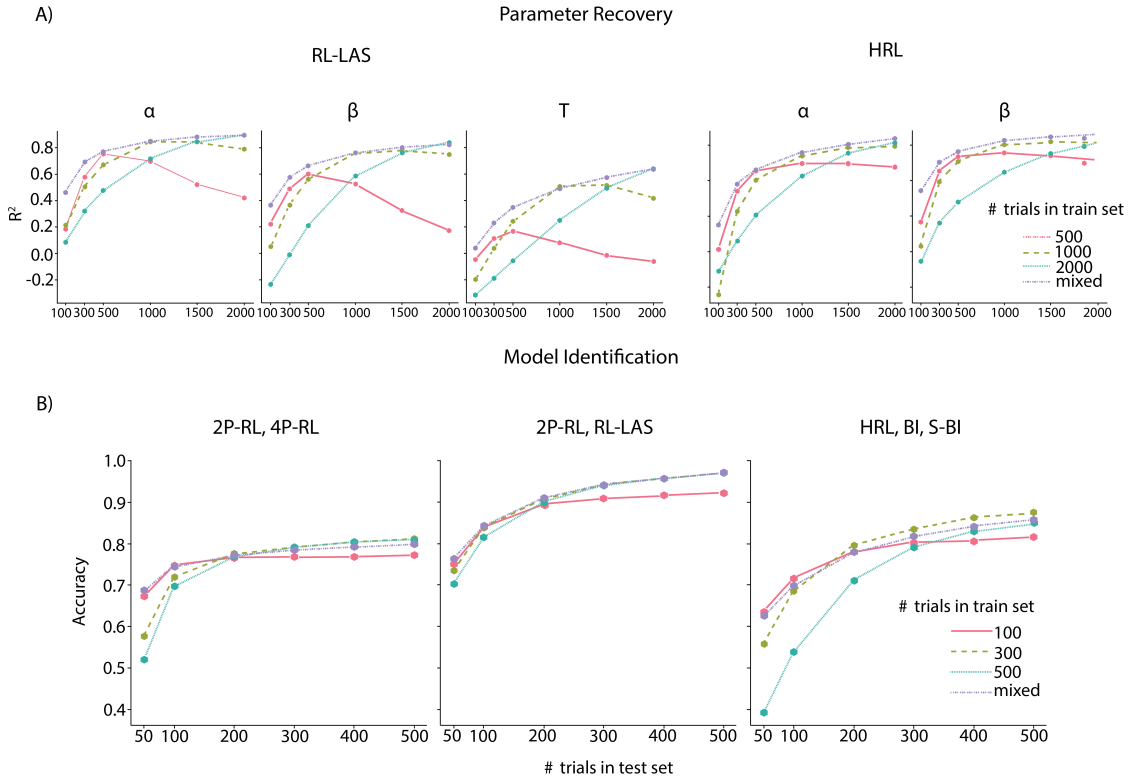


Figure 4.6: Robustness checks using different training (different line colors) and testing (x-axis) trial sequence lengths. A) Parameter estimation in both RL-LAS and HRL show that training with a mixture of trial sequence lengths (purple line) yields more robust out-of-sample parameter value prediction compared to fixed trial sequence lengths. B) Best model identification results, performed on different combinations of model candidates, were also yielded by mixed trial sequence length training. The number of agents/simulations used for training was kept constant across all the tests (N agents = 30k).

To make the model misspecification more extreme, we additionally simulated data from a Bayesian inference model, and estimated RL model parameters from the simulated data. We did this using standard methods (MAP) and ANN, and repeated the same process in reverse (simulating data from an RL model, and fitting Bayesian inference model parameters). We found that both MAP and ANN exhibited similar patterns. That is, in the case of simulating Bayesian inference model and fitting RL model parameters, the estimated β captured the variance from the true β and p_{switch}, while the estimated α parameter captured the variance driven by the Bayesian updating parameters p_{reward} and p_{switch} (Supplementary fig. 4.20). In the case of simulating RL model and fitting Bayesian inference model parameters, p_{switch}

parameter captured the noise in the simulated data coming from the β parameter, and the variance from the α parameter was attributed to the p_{reward} parameter (Supplementary fig. 4.21). We also correlated parameter estimates generated by the two methods. High correlation implies that MAP and GRU generate similar parameter estimates, suggesting that they are impacted by model misspecification in a similar way (Supplementary fig. 4.17).

4.4 Discussion

Our results demonstrate that artificial neural networks (ANNs) can be successfully and efficiently used to estimate best fitting free parameters of likelihood-intractable cognitive models, in a way that is independent of likelihood approximation. ANNs also show remarkable promise in successfully arbitrating between competing cognitive models. While our method leverages "big data" techniques, it does not require large experimental data sets: indeed, the large training set used to train the ANNs is obtained purely through efficient and fast model simulation. Thus, our method is applicable to any standard cognitive data set with a normal number of participants and trials per participants. Furthermore, while our method requires some ability to work with ANNs, it does not require any advanced mathematical skills, making it largely accessible to the broad computational cognitive modeling community.

Our method adds to a family of approaches from other attempts at using neural networks for fitting computational cognitive models. Specifically, previous work leveraging amortized inference has focused on taking advantage of large-scale simulations and invertible networks. This approach involves training the summary segment of the network to adeptly learn relevant summary statistic vectors, while concurrently training the inference segment of the network to approximate the posterior distribution of model parameters based on the outputs generated by the summary network (Radev, Mertens, et al., 2020; Radev, Voss, et al., 2020; Schmitt et al., 2021). This method has successfully been applied to both parameter estimation and model identification (and performs in a similar range as our method when applied to intractable models we implemented in this paper), bypassing many issues of ABC. In parallel, work by Fengler et al., 2021 showcased Likelihood Approximation Networks (LANs) as a method that approximates likelihood of sequential sampling models (but requires ABC-like approaches for training), and recovers posterior parameter distributions with high accuracy for a specific class of models (e.g. drift diffusion models); more recently, Boelts et al., 2022 used a similar approach with higher training data efficiency. Work by Lueckmann et al., 2017 used Approximate Bayesian Computation (ABC) in conjunction with mixture density networks to map data to parameter posterior distributions. Unlike most of these approaches our architecture is not dependent on Boelts et al., 2022; Fengler et al., 2021 or explicitly designed to optimize Radev, Mertens, et al., 2020; Radev, Voss, et al., 2020; Schmitt et al., 2021 summary statistics. By necessity, hidden layers of our network do implicitly compute a form of summary statistic that are translated into estimated parameters/model class in the

output layer; however, we do not optimize for such statistics explicitly, beyond their ability to support parameter/model recovery.

Other approaches have used ANNs for different purposes than fitting cognitive models (Thompson et al., 2022). For example, Dezfouli et al., 2019 leveraged flexibility of RNNs (which inspired our network design) to map data sequences onto separable latent dimensions that have different effects on decision-making behavior of agents, as an alternative to cognitive models that make more restrictive assumptions. Similarly, work by Ger et al., 2023 also used RNNs to estimate RL parameters and make predictions about behavior of RL agents. Our work goes further than this approach in that it focuses on both parameter recovery and model identification of models with intractable likelihood, without relying on likelihood approximation. Furthermore, multiple recent papers (Eckstein et al., 2023; Ji-An et al., 2023) use ANNs as a replacement for cognitive models, rather than as a tool for supporting cognitive modeling as we do, demonstrating the number of different ways ANNs are taking a place in computational cognitive science.

It is important to note that while ANNs may prove to be a useful tool for cognitive modeling, one should not expect that their use immediately fixes or overrides all issues that may arise in parameter estimation and model identification. For instance, we have observed that while ANNs outperformed many of the traditional likelihood-based methods, recovery for some model parameters was still noisy (e.g. learning rate α in the 4P-RL model, Fig. 4.2). This is a property of cognitive models when applied to experimental applied to data sets that range in hundreds of trials. Standard methods (e.g. MAP) fail in a similar way - as shown by the high correlation between MAP and ANN parameter estimates (Fig. 4.16), which suggests that parameter recovery issues have more to do with identifiability limitations of the data and model, rather than other issues such as optimization method. Similarly, often times model parameters are not meaningful in certain numerical ranges, and sometimes model parameters trade off in how they impact behavior through mathematical equations that define the models - making the parameter recovery more challenging. Furthermore, when it comes to model identification, particularly with nested models, the specific parameter ranges can influence the outcome of model identification, favoring simpler models over more complex ones (or vice versa). This was evident in our observations regarding the confusion between Bayesian inference models with and without stickiness, wherein the ground truth values of stickiness played a decisive role in the model identification. This is to say ANNs should be treated as a useful tool that is only useful if the researchers apply significant forethought to developing appropriate, identifiable cognitive models.

In a similar vein, it is important to recognize that the potential negative implications of model misspecification extend to neural networks, much like they impact traditional model-fitting approaches. For instance, our estimation of parameters may be conducted under the assumption of model X, whereas, in reality, model Y might be the most suitable for explaining the data - leading to poor parameter estimation and model predictions. Our test of the systematic effects of model misspecification involved utilizing a network trained to estimate parameters from one model (e.g. Bayesian Inference) to predict parameters for the held-out

test set data simulated from a different model (e.g. Bayesian Inference with stickiness, or RL). We compared this to model misspecification with a standard MAP approach.Notably, neither method exhibited significant adverse effects. When models were nested, the parameters shared between the two models were reasonably well recovered. When the model misspecificpation was more extreme (with models from different families), we again observed similar effects on the two methods, where variance driven by one parameter tended to be recovered similarly. Thus, our approach appears equally (but not worse) subject to the risk of model misspecification as other fitting methods. In light of these findings, our key takeaway is to exercise caution against assuming that the use of a neural network remedies all issues typically associated with modeling. Instead, we advocate for the application of conventional diagnostics (e.g., model comparison, predictive checks) that are commonly employed in standard methods to ensure robust and accurate results.

Relatedly, we have shown that the parameter estimation accuracy varies greatly as a function of the parameter range the network was trained on, along with whether the underlying parameter distribution of the held out test-set is included in that range or not. This is an expected property of ANNs that are known to underperform when the test data systematically differs from training examples (Liang et al., 2017; Nguyen et al., 2015; Szegedy et al., 2013). As such, the range of parameters/models used for inputs constitutes a form of prior that constrains the fit, and it is important to carefully specify it with informed priors (as is done with other methods, such as MAP). We found that training the network using a broader parameter range, while heavily sampling from a range of interest (e.g. plausible parameter values based on previous research) affords both accurate prediction for data generated outside of the main expected range, with limited loss of precision within the range of interest (Fig. 4.15). This kind of practice is also consistent with practices in computational cognitive modeling, where a researcher might specify (e.g. using a prior) that parameter might range between two values, with most falling within a certain, more narrow range.

One factor that is specific to ANN-based methods (as opposed to standard methods) is the effect different hyperparameters (e.g. size of the neural network, choice of the learning rate, dropout values, etc.) may have on network performance - commonly resulting in overfitting or underfitting. We observed that the network performance, particularly in parameter recovery, is most significantly influenced by the number of units in the GRU layer and the chosen dropout rate. A suitable range for the number of GRU units is typically between 90 and 256, covering the needs of most cognitive models. A dropout rate within the range of 0.1 to 0.2 is generally sufficient. We have outlined the details of parameter ranges we tested in the table in supplementary materials (4.1). To address this challenge, we employed an automated hyperparameter tuning approach, as outlined by Bergstra, Yamins, and Cox (2013). This Bayesian optimization for tuning hyper-parameters helps reduce the time required to obtain an optimal parameter set by learning from previous iterations. Additionally, in the process of training a neural network, the initialized random weights play a significant role in determining the network's convergence and the final performance. Different random seeds can result in different initializations of the network weights, which may affect the optimization process

downstream, and potentially yield different final solutions. It is important to be mindful of this; we have inspected effects of setting different seeds on our network performance (Fig. 4.23), and found that overall network performance was stable across different seeds, with slight variations (1 seed) for both parameter estimation and model identification - showcasing the need for cautious practice of inspecting network's performance under multiple seeds.

We compared our artificial neural network approach against existing methods that are commonly used to estimate parameters of likelihood-intractable models (e.g. ABC, Sisson et al., 2018; Sunnåker et al., 2013). While traditional rejection ABC provides a workaround solution, it also imposes certain constraints. Specifically, it is more suitable for data with no sequential-dependencies, and the accuracy of parameter recovery is largely contingent on selection of appropriate summary statistics, which is not always a straightforward problem. More recent advances in the domain of simulation-based inference (Fearnhead and Prangle, 2012; Jiang et al., 2017; Lavin et al., 2021; Radev, Mertens, et al., 2020) solve many ABC-issues by automating the process of construction of summary statistics. For the purpose of this project we have focused on the methods that are most commonly used in cognitive modeling (e.g. maximum likelihood/maximum a posteriori), but future work should extend to conducting the same benchmarking procedure involving these inference methods.

Alternative approximation methods (e.g. particle filtering (Djuric et al., 2003); density estimation (Minka, 2013)); inverse binomial sampling (van Opheusden et al., 2020) may prove to be more robust, but frequently require more advanced mathematical knowledge and model case-based adaptations, or are more computationally expensive; indeed, some of them may not be usable or tractable in our type of data and models where there are sequential dependencies between trials Acerbi and Ma, 2017; van Opheusden et al., 2020. ANN-based methods such as ours or others' Radev, Mertens, et al., 2020; Radev, Voss, et al., 2020; Sokratous et al., 2023, on the other hand, offers a more straightforward and time-efficient path to both parameter estimation and model identification. Developing more accessible and robust methods is critical for advances in computational modeling and cognitive science, and the rising popularity of deep learning puts neural networks forward as useful tools for this purpose. Our method also offers an advantage of requiring very little computational power. The aim of the project at its current state was not to optimize our ANN training in terms of time and computing resources; nevertheless, we used Nvidia V100 GPUs with 25 GB memory and required at most 1 hour for model training and predictions. This makes the ANN tool useful, as it requires a low amount of computing resources and can be done fast and inexpensively. All of our code is shared on GitHub.

We primarily focused on extensive tests using synthetic data, in particular in the context of learning experiments that present important challenges for some methods (such as BADS (Acerbi and Ma, 2017) or ABC (Sunnåker et al., 2013; Turner and Sederberg, 2014; Turner et al., 2013) due to the dependency between trials, and have not been thoroughly investigated with other ANN-based approaches. A critical next step will be to further validate our approach using empirical data (e.g. participant data from the tasks). Similarly, we relied on RNNs due to their flexibility and capacity to handle sequential data. However, it will

be important to explore different structures, such as transformers (Devlin et al., 2018), for potentially improved accuracy in parameter recovery/model identification, as well as alternative uses in cognitive modeling.

In addition, our baseline approach lacks the capability to quantify the complete uncertainty in parameter estimation, offering only point estimates. This is similar to many lightweight cognitive modeling approaches (such as MAP and LLH), but stands in contrast to other methods that integrate simulation-based inference with neural network structures (Boelts et al., 2022; Fengler et al., 2021; Radev, Mertens, et al., 2020; Radev, Voss, et al., 2020; Radev et al., 2021), where the ability to capture full uncertainty represents a notable strength. Nevertheless, we have showcased that our method can easily be extended to provide uncertainty estimates by incorporating evidential learning techniques (Amini et al., 2020), at a slight computational cost, but minimal impact on point estimates' accuracy. Furthermore, we included both RL and Bayesian inference models to demonstrate our approach can work with different classes of computational models. Future work will include additional models (e.g. sequential decision making models) to further test robustness of our approach.

In conclusion, we propose an accessible ANN-based method to perform parameter and model identification across a broad class of computational cognitive models for which application of existing methods is challenging. Our work should contribute to a growing literature focused on developing new methods that will allow researchers to quantitatively test a broader family of theories than previously possible.

4.5 Methods

Tasks

Probabilistic reversal learning task. We have simulated data from different models (see the Models section) on a simple probabilistic reversal learning task (PRL; Cools et al., 2002). In the task, an agent chooses between two actions on each trial, and receives binary outcome ($r = 1$ [reward] or $r = 0$ [no reward]). One of the two actions is correct for a number of trials; a correct action is defined as the action that gets rewarded with higher probability (e.g. $p(r = 1|action = correct) = 0.80$), with $1 - p$ probability of getting no reward if selected. After a certain number of trials, the correct action reverses; thus the action that was previously rewarded with low probability becomes the more frequently rewarded one (Fig: 4.1). This simple task (and its variants) have been extensively used to provide mechanistic insights into learning from reinforcement, inferring probabilistic structure of the environment, and people's ability (or failure) to update the representation of a correct choice.

Hierarchical reinforcement learning task. We developed a novel task environment that can be solved using a simple but plausible model with intractable likelihood. In this task, an agent observes N arrows (in different colors), each pointing at either left or right direction. The agent needs to learn which arrow is the correct one, by selecting an action that

corresponds to either left or right side (consistent with the direction the arrow is pointing at) in order to get rewarded. Selecting the side the correct arrow is pointing at rewards the agent with high probability ($p = .9$); choosing an action by following direction of other arrows leads to no reward ($r = 0$) with same high probability. The correct arrow changes unpredictably in the task, which means that the agent must keep track of which arrow most reliably leads to the reward, and update accordingly upon the change. We refer to this task structure as hierarchical because the choice policy (left/right) depends on the higher-level rule (color) agents choose to follow.

Cognitive Models

PRL task models

.

We implemented multiple models of the PRL task to test the artificial neural network (ANN) approach to parameter estimation. First, we cover the benchmark models; these are the models that we can fit using traditional methods (MLE, MAP), as well as the ANN, to ensure that we can justify using the ANN if it performs at least just as well as (or better than) the traditional methods.

Reinforcement learning models family.

Two-parameter reinforcement learning model. We simulated artificial data on the PRL task using a simple 2-parameter reinforcement learning model (2P-RL). The model assumes that the agent tracks the value of each action contingent on the reward history, and uses these values to inform the action selection on each trial.

The model uses simple delta rule to update action values on each trial upon outcome observation, by first computing the reward prediction error (RPE, δ) as the discrepancy between the expected and the observed outcome, and then adjusting the value of the chosen action using the RPE scaled by the learning rate (α) (Sutton and Barto, 2018):

$$\begin{aligned} \delta &= r - Q_t(a) \\ Q_{t+1}(a) &= Q_t(a) + \alpha\,\delta \end{aligned} \tag{4.1}$$

We also allowed for counterfactual updating, where the value of the non-chosen action also gets updated on each trial (Eckstein, Master, Dahl, et al., 2022; Hauser et al., 2015):

$$\begin{aligned} \delta_{\text{unchosen}} &= (1 - r) - Q_t(1 - a) \\ Q_{t+1}(1 - a) &= Q_t(1 - a) + \alpha\,\delta_{\text{unchosen}} \end{aligned} \tag{4.2}$$

The action values are transformed into action probabilities using the softmax function, thus defining a policy where actions with higher value are chosen with higher probabilities. The β parameter controls how deterministic the choices are with higher values of β corresponding to more deterministic choices:

$$P(a) = \frac{\exp(\beta\ Q_t(a))}{\sum_{i=1}^{n_A} \exp(\beta\ Q_t(a_i))} \tag{4.3}$$

The 2p-RL model contained following free parameters: learning rate (α) and softmax beta (β).

Four-parameter reinforcement learning model. The four parameter RL (4P-RL) model follows the same updating and policy structure as the 2-parameter RL, with 2 main differences. The 4P-RL model differentiates between positive and negative feedback (Niv et al., 2012), by using different learning rates - α^+ and α^- for updating action values after positive and negative outcomes respectively:

$$Q_{t+1}(a) = \begin{cases} Q_t(a) + \alpha^+\,\delta & \text{if } \delta\ > 0 \\ Q_t(a) + \alpha^-\,\delta & \text{if } \delta\ \leq 0 \end{cases}$$

Furthermore, 4P-RL model also includes the stickiness parameter κ which captures the tendency to repeat choice from the previous trial:

$$P(a) \propto exp(\beta Q + \kappa\ \text{same}(a, a_{t-1})) \tag{4.4}$$

Like in the 2P-RL we also included counterfactual updating of values for non-selected actions. The 4P-RL model included following free parameters: positive learning rate (α^+), negative learning rate (α^-), softmax beta (β) and stickiness (κ).

Bayesian models family.

Bayesian inference model. Bayesian inference model (BI) assumes that an agent infers the latent state in the environment, updates the latent state based on new observations, and uses the inference process to make rewarding choices. For instance, in the PRL task, the agent infers a latent state corresponding to the correct action ($C_t : a_{right} = cor$ or $C_t : a_{left} = cor$) at time t. The agent tracks and updates their belief over which one of the two actions is currently the correct one based on 1) their estimate of the switch frequency (p_{switch}) and 2) how noisy the reward is (p_{reward}) from the history of observations up to the previous trial H_{t-1}. On each trial, the belief is updated according to the Bayes rule - based on the prior belief (agent's model of the task) and likelihood of observed evidence (the outcome given the choice):

$$p(C_t = i|r_t, a_t, H_{1:t-1}) = \frac{P(r_t|C_t = i, a_t)P(C_t = i|H_{1:t-1})}{\sum j(P(r_t|C_t = j, a_t)P(C_t = j|H_{1:t-1})}$$

where i and j are in [left/right], $p(C_t = i|H_{1:t-1})$ is the prior probability, and $p(r_t|C_t = i, a_t)$ is the likelihood of outcome given the action. The likelihood is defined in accordance to whether the choice matches the latent state:

$$p(r_t = 1|a_t = i, C_t = i) = p_{reward}$$

where p_{reward} is the parameter controlling the probability of receiving the reward given the choice of correct action. Posterior belief for the correct action is updated to a prior belief for the upcoming trial in accordance with the p_{switch} parameter, which determines the probability that the correct action might have reversed on the current trial:

$$p(C_{t+1} = i|H_{1:t-1}) = (1 - p_{switch})p(C_t = i|H_{1:t-1}) + p_{switch}(1 - p(C_t = i|H_{1:t-1}))$$

Like in the RL models, the action selection in Bayesian models also followed the softmax policy; however, instead of being informed by the Q values the action probabilities were determined by the belief W given the choice and reward history H and the choice parameter β:

$$W_{t+1} = p(C_{t+1} = i|H_{1:t})$$

$$P(a_{t+1}) = \frac{\exp(\beta\ W_i(t+1))}{\sum_{i=j} \exp(\beta\ W_j(t+1))}$$

The BI model included following parameters: inferred probability of reward given the action determined by the current belief (p_{reward}), likelihood of the correct action reversing (p_{switch}) and softmax beta (β).

Bayesian inference model with stickiness. We also added a variation of the Bayesian inference model that accounts for sticky choice behavior (e.g. repeating actions) by introducing a stickiness parameter κ that augments the belief associated with the action chosen on the previous trial:

$$W_{t+1} = p(C_{t+1} = i|H_{1:t}) + \kappa(i = a_t)$$

Intractable likelihood

As a proof of concept, we implemented a simple model that assumes a latent state of agent's attention (engaged/disengaged). This model can't be fit using methods that rely on computing likelihood. While models can have intractable likelihood for a variety of reasons, we focused on leveraging latent variables (e.g. attention state), that are not readily observable in the data. Thus, in the data that is being modeled, only the choices are observed - but

not the state the agent was in while executing the choices. The learned choice value which affects the choice likelihood depends on the trial history, including which state the agent was in. Thus, if there are 2 such states, there are 2^N possible sequences that may result in different choice value estimates after N trials. To estimate choice values and likelihood on any given trial one must integrate over the uncertainty of an exponentially increasing latent variable - thus making the likelihood intractable.

RL and latent engagement state . We simulated a version of a 2p-RL model for a probabilistic reversal learning (PRL) task that also assumes that an agent might occupy two of the latent attention states — engaged or disengaged— during the task (RL-LAS). The model assumes that in the engaged state an agent behaves in accordance with the task structure (e.g. tracks and updates values of actions, and uses action values to inform action selection). In the disengaged state, an agent behaves in a noisy way, in that 1) it does not update the Q value of actions , and 2) chooses between the two actions randomly (e.g. uniform policy) instead of based on their value (e.g. through softmax). [1] The agent can shift between different engagement states at any point throughout the task, and the transition between the states is controlled by a parameter τ. Specifically, for each agent we initialized a random value T between 10 and 30 (which roughly maps onto approximately how many trials an agent spends in a latent attention state), and then used a non-linear transformation to compute τ: 1-(1/T). The value of τ, thus quantifies the probability of transitioning between the two states. The agent was initialized to be in an attentive state at the onset of trials.

The likelihood of this model can be computed:

$$\begin{aligned}\mathcal{L}(\theta) &= \sum_{t=1}^{T} \log \mathbb{P}(a_t|h_t, \overline{h}_{t-1}, \theta) \\ &= \sum_{t=1}^{T} \log\Big(\sum_{l} \mathbb{P}(a_t|h_t, ls_t = l; \theta)\mathbb{P}(ls_t = l, \overline{h}_{t-1}; \theta)\Big)\end{aligned}$$

where ls = latent state, l $\in$ { 0 = disengaged state, 1 = engaged state }, $\overline{h}_{t-1}$ corresponds to the history of actions and rewards up to the trial t. However, it is in practice intractable, because of the sum over latent states in the equation, which cannot be factored out.

Cognitive models of the HRL task

Bayesian models of the HRL task .

[1] Note that assumption 1) is different from a previous version of the model our group considered (Li, Shi, Li, and Collins, 2023; Li, Shi, Li, and Collins, 2023), and is the core assumption that renders the likelihood intractable.

Bayesian models of the HRL task assume an inference process over the latent variable of which arrow is currently the valid arrow, and thus which side (R/L) (given the current trial's set of arrows) is most likely to result in positive outcome. The inference relies on the generative model of the task determined by parameters p_{switch} and p_{reward}, history of trial observations O_t, set of arrows and stochastic choice based on this inference. Initial prior belief over arrows is initialized uniformly $prior = 1/nA$, where nA corresponds to the number of arrows.

To determine the agent policy over arrows at trial t, we first implemented a softmax function with decision parameter β and prior belief of which arrow is the correct one; once the arrow is selected , the agent implements an ϵ-greedy policy conditioned on the selected arrow A_t to choose a R/L side:

$$P(side(A_t)|A_t) = 1 - \epsilon$$

Likelihood $p(r_t = 1|A_t, side(A_t))$ and posterior are then updated into the prior belief for the next trial using the p_{switch} model of the task parameter:

$$p(C_{t+1} = i|O_{1:t-1}) = (1 - p_{switch}) * p(C_t = i|O_{1:t-1}) + p_{switch}(1 - p(C_t = i|O_{1:t-1}))$$

This belief is subsequently used to inform arrow choices on the next trial. This model differs from the Bayesian Inference model for the probabilistic task in that 1) p_{reward} and p_{switch} parameters are not free/inferred and 2) the choice of the side is stochastic, allowing for a potential lapse in selecting the side that is not consistent with the selected arrow. This model, thus has following free parameters: decision parameter β and noise parameter ϵ. Like in the in Bayesian inference model for the PRL task, we also tested the model variant with stickiness κ parameter that biases beliefs associated with the arrow/side chosen on the previous trial. Both models have tractable likelihoods.

Hierarchical reinforcement learning . We also simulated a simple hierarchical reinforcement learning (HRL) model to simulate the performance on a HRL task (see tasks section, 4.7). This model assumes that an agent tracks the value of each of the arrows, and chooses between the arrows noisily:

$$P(arrow) = \frac{\exp(\beta\ Q_t(arrow))}{\sum_{i=1}^{n_A} \exp(\beta\ Q_t(arrow_i))} \tag{4.5}$$

We have also explored the model with an assumption that an agent has a tendency to repeat the choice from the previous trial, captured by the stickiness parameter κ:

$$P(arrow) = \frac{\exp(\beta\ Q_t(arrow) + \kappa(arrow = arrow_{t-1}))}{\sum_{i=1}^{n_A} \exp(\beta\ Q_t(arrow_i))} \tag{4.6}$$

Once the agent chooses the arrow,it greedily chooses the direction based on which side (left/right) the arrow is pointing at (observable). Note that we only know the side the agent selected (left/right), because the arrow the agent chooses is non-observable. The agent then observes an outcome, and updates the value of the selected arrow based on the observed outcome:

$$Q_{t+1}(\text{arrow}) = Q_t(\text{arrow}) + \alpha(r - Q_t(\text{arrow}))$$

In the case of this model, the likelihood is intractable because of the need to integrate over uncertainty of what rule (which arrow) the agent followed on all of the past trials; because the integration exponentially increases with each time point, the likelihood is not tractable beyond the first several trials:

$$\begin{aligned}\mathcal{L}(\theta) &= \sum_{t=1}^{T} \log \mathbb{P}(a_t|h_t, \overline{h}_{t-1}, \theta) \\ &= \sum_{t=1}^{T} \log \mathbb{P}\Big(\sum_{c} \mathbb{P}(a_t|h_t, rule_t = c; \theta)\mathbb{P}(rule_t = c, \overline{h}_{t-1}; \theta)\Big)\end{aligned}$$

where a_t corresponds to the action dictating which side the agent selected (left/right), $\overline{h}_{t-1}$ corresponds to the task history encoding rewards, selected actions/sides, arrow directions, and c correspond to identity/color of the correct arrow.

Likelihood-dependent methods.

Maximum likelihood and Maximum a posteriori estimation

Maximum likelihood estimation (MLE) represents a cornerstone of modeling that leverages probability theory and estimation of likelihood ($P(D|M, \theta)$) of the data given the model parameters and assumptions (Myung, 2003). The parameter estimates are determined as the values that maximize the likelihood of the data:

$$\begin{aligned}\theta_{MLE} &= argmax P(D|\theta) \\ &= argmax \prod_i P(D_i|\theta) \\ &= argmax \sum_i log P(D_i|\theta)\end{aligned}$$

Thus, to estimate best fitting parameters via MLE, the likelihood of the data is computed and maximized with respect to parameter values via an optimization algorithm (often a

blackbox one, such as fmincon in MATLAB or optimize.minimize from scipy toolbox in python). Maximum a posteriori estimation (MAP) relies on much the same principle, with an addition of a prior $p(\theta)$ to maximize the posterior:

$$\theta_{MAP} = argmax \sum_i logP(D_i|\theta)logP(\theta)$$

As a prior for the MAP approach, we used an empirical prior derived from the true simulating distribution of parameters (see supplement for details). We note that this gives an advantage to the MAP method above what would be available for empirical data, allowing MAP to provide a ceiling performance on the test set.

Because MAP and MLE rely on likelihood computation, their use is essentially limited to models with tractable likelihood. We used MAP and MLE to estimate parameters of tractable-likelihood models as one of the benchmarks against which we compared our ANN approach. Specifically, we fit the models to the test-set data used to compute the MSE of the ANN, and compared fit using the same metric across methods (see main text).

Likelihood approximation methods

Because models with tractable likelihood comprise only a small subset of all possible (and likely more plausible) models, researchers have handled the issue of likelihood intractability by implementing various likelihood approximation methods. While there are different likelihood approximation tools, such as particle filtering (Djuric et al., 2003) and assumed density estimation (Minka, 2013), we focus on Approximate Bayesian Computation (ABC; Lintusaari et al., 2017; Palestro et al., 2018; Sisson et al., 2018; Sunnåker et al., 2013), as it is more widely accessible and does not require more extensive mathematical expertise. ABC leverages large scale model simulation to approximate likelihood. Specifically, a large synthetic data set is simulated from a model, with parameters randomly sampled from a specific range for each agent. Summary statistics that describe the data (e.g average accuracy or variance in accuracy) are used to construct the empirical likelihood that can be used in combination with classic methods.

We implemented a basic form of ABC - the rejection algorithm (Sunnåker et al., 2013). This algorithm first samples a set of model parameters θ, simulates the data $\hat{D}$ from the model M using these parameters and computes the predetermined summary statistic $S(\hat{D})$ of the simulated data which we refer to as the sample. The summary statistics of the real data $S(D)$ and the sample $S(\hat{D})$ are then compared - if the distance between the two sets of summary statistics ρ is greater than the predetermined criterion ϵ, the sample is rejected:

$$\rho(S(\hat{D}), S(D)) \leq \epsilon$$

The distance metric, like the rejection criterion, is determined by the researcher. The samples that are accepted are the samples with distance to the real data smaller than the

criterion, resulting in the conclusion that parameters used to generate the sample data set can plausibly be the ones that capture the target data. Thus, the result of the ABC for each data set is a distribution of plausible parameter values which can be used to obtain point estimates via the mean, median, etc.

ABC is a valuable tool,but standard ABC has serious limitations (Sunnåker et al., 2013). For instance, the choice of summary statistics is not a trivial problem, and different summary statistics can yield significantly different results. Similarly, in the case of rejection algorithm ABC, researchers must choose the rejection criterion which can also affect the parameter estimates. A possible way to address this is using cross validation to determine which rejection criterion is the best, but this also requires specification of the set of possible criteria values for the cross validation algorithm to choose from. Furthermore, one of ABC assumptions is independence of data points, which is violated in many sequential decision making models (e.g. reinforcement learning).

To compare our approach to ABC, we used network training set data as a large scale simulation data set, and then estimated parameters of the held out test set also used to evaluate the ANN.

To apply ABC in our case, we needed to select summary statistics that adequately describe performance on the task. We used the following summary statistics to quantify the agent for the models simulated on the PRL task:

- Learning curves: We computed agents' probability of selecting the correct action, aligned to the number of trials with reference to the reversal point. Specifically, for each agent we computed an average proportion of trials where a correct action was selected N trials before and N trials after the correct action reversal point, for all reversal points throughout the task. This summary statistic should capture learning dynamics, as the agent learns to select the correct action, and then experiences dip in accuracy once the correct actions switch, subsequently learning to adjust based on feedback after several trials.

- 3-back feedback integration: The 3-back analysis quantifies learning as well; however, instead of aligning the performance to reversal points, it allowed us to examine agents' tendency to repeat action selection from the previous trial contingent on reward history - specifically the outcome they observed on the most recent 3 trials. Higher probability of repeating the same action following more positive feedback indicates learning/sensitivity to reward as reported in Zou et al., 2022

- Ab-analysis: The Ab-analysis allowed us to quantify probability of selecting an action at trial t,contingent both on previous reward and action selection history (trials $t-2$ and $t-1$, Beron et al., 2021; Zou et al., 2022).

For the models simulated on a hierarchical task we used the learning curves as summary statistics (same as for the PRL), where reversal points were defined as the switch of the

correct rule/arrow to follow. In addition, we quantified agent's propensity to stick with the previously correct rule/arrow, where the agent should be increasingly less likely to select the side consistent with the arrow that was correct before the switch as the number of trials since the switch increases. Similarly, we used a version of the 3-back analysis where the probability of staying contingent on the reward history referred to the probability of potentially selecting the same cue across the trial window, based on observed choices of the agent. All summary statistics are visualized in the supplementary figure 4.14.

Model comparison

To perform benchmark model comparison, we used the Akaike Information Criterion (AIC) metric (Akaike, 1998), commonly used to evaluate relative model fitness, with an aim of identifying the best model candidate that might have generated the data. The AIC score combines model log likelihood and number of parameters to quantify model fitness, while also penalizing for model complexity in order to prevent overfitting:

$$AIC = -2(LLH) + 2K$$

where K corresponds to the number of parameters. The model with the lowest AIC scores corresponds to the best fitting model for the given data. A related metric that is commonly used is the Bayesian Information Criterion (BIC, Schwarz, 1978), which considers the number of observations (N) as well, and similarly uses the lowest score to signal the best fitting model:

$$BIC = -2(LLH) + K * log(N)$$

We used AIC score as it outperformed BIC model comparison, and thus provided us with ceiling benchmark to evaluate the ANN.

To perform proper model comparison, it is essential to not only evaluate the model fitness (overall AIC/BIC score), but also to test how reliably the true models (that generated the data) can be identified/successfully distinguished from others. To do so, we constructed a confusion matrix based on the AIC score (Fig. 4.5A). We used the test set data simulated from each model, and then fit all candidate models to each of the data sets while also computing the AIC score for each fit. If the models are identifiable, we should observe that AIC scores for true models (e.g. the models the data was simulated from) should be the lowest for that model when it's fit to the data compared to other model candidates.

Artificial neural network-based method

Parameter recovery

To implement ANNs for parameter estimation we have used the relatively simple neural network structure inspired by the previous work (Dezfouli et al., 2019). In all experiments, we

used 1 recurrent GRU layer followed by 3 fully connected dense layers with 2000 dimensional input embeddings (4.1). To train the network, we simulated a training data set using known parameters. For each model, we used 30000 training samples, 3000 validation samples, and 3000 test samples that are generated from simulations separately. For probabilistic RL, the input sequence consisted of rewards and actions. For hierarchical RL, the sides (left/right) of three arrow stimuli are added to the rewards and actions sequences. The network output dimension was proportional to the number of model parameters. We used a $tanh$ activation in the GRU layer, $reLu$ activations in 2 dense layers, and a linear activation at the final output. Additional training details are given below:

- We used *He* normal initialization to initialize GRU parameters (He et al., 2015).
- We used the Adam optimizer with mean square error (MSE) loss and a fixed learning rate of 0.003. Early stopping (e.g. network training was terminated if validation loss failed to decrease after 10 epochs) was applied with a maximum of 200 epochs.
- We selected network hyperparameters with Bayesian optimization algorithms (Bergstra et al., 2013) applied on a validation set. Details of the selected values are shown in Supplementary Materials.

All of the training/validation was run using TensorFlow (Abadi et al., 2016). The training was performed on Nvidia V100 GPUs with 25 GB memory.

Network evaluation. The network predicted the values of parameter on the test set that is unseen in the training and validation. We also conducted robustness tests by varying trial numbers (input size).

To evaluate the output of both ANN and traditional tools we used the following metrics (ensuring our results are robust to the choice of performance quantification):

- Mean squared error (MSE): To evaluate parameter estimation accuracy we calculated a mean squared error between true and estimated model parameter across all agents. Prior to calculating MSE all parameters were normalized, to ensure comparable contribution to MSE across all parameters. Overall loss for a cognitive model (across all parameters) was an average of individual parameter MSE scores. Overall loss for a class of models (e.g. likelihood-tractable models) was an average across all model MSE scores.
- Spearman correlation (ρ): We used Spearman correlation as an additional metric for examining how estimated parameter values relate to true parameter values, with higher Spearman ρ values indicating higher accuracy. We paired Spearman correlations with scatter plots, to visualize patterns in parameter recoverability (e.g. whether there are specific parameter ranges where parameters are more/less recoverable).

- R-Squared (R^2 or the coefficient of determination): R-Squared represents the proportion of variance in true parameters that can be explained by a linear regression between true and predicted values. It thus indicates the goodness of fit of an ANN model. We calculated an R-Squared score for each individual parameters across all agents and used it as an additional evaluation for how well the data fit the regression model.

Uncertainty estimation To compute uncertainty of parameter estimates we have incorporated evidential learning into our method (Amini et al., 2020). In the application of evidential learning to continuous regression (Amini et al., 2020) observed targets follow a Gaussian distribution, characterized by its mean and variance. Conjugate Gaussian prior, normal inverse-gamma distribution, is created by placing a prior on both the mean (Gaussian distribution) and the variance (inverse-gamma distribution). By sampling from this distribution, specific instance of the likelihood function is obtained (based on both mean and the variance). This approach not only aims for accurate target predictions but also takes into account the uncertainty (quantified by the variance term). For more insights and details into evidential learning, refer to the work by Amini et al., 2020. Their research also introduces a regularization term, which is useful for penalizing incorrect evidence and data that falls outside the expected distribution.

For the purpose of visualization (Fig. 4.4) we have created upper and lower bounds of targets by adding/subtracting variance from the predicted target values. We then rescaled these values by applying the inverse scaler (e.g. from the scaler applied to normalize parameters for network training). This provides a scale-appropriate and more interpretable visualization of parameter recovery and uncertainty for each parameter.

Alternative models. We have also tested the network with long short-term memory (LSTM) units since LSTM units are more complex and expressive than GRU units; nevertheless they achieved the similar performance as GRU units but are more computationally expensive, and thus we mostly focused on the GRU version of the model. Since LSTM worked, but not better than GRU, the LSTM results are reported in the supplementary materials.

Model identification

The network structure and training process were similar to that of the network used for parameter recovery, with an exception of the output layer that utilized categorical cross-entropy loss and a softmax activation function. The network validation approach was the same as the one we used for parameter recovery (e.g. based on the held-out test set). We also observed a better performance when training with various trial numbers.

Robustness test: influence of different input trial numbers

For all robustness experiments, we followed the same training procedures as described previously while varying the training data. The details of training data generation are given below:

Parameter Recovery We simulated 30,000 training samples with 2000 trials per simulation in the probabilistic reversal learning task. For shorter fixed trial sequence lengths per training samples (e.g 500), we used the same training set truncated to the first 500 trials. To generate the training data with different trial numbers across training samples, we reused the same training set, with sequences of trials truncated to a given number. There were 6000 training samples of 50, 100, 500, 1000, 1500, and 2000 trials, each.

Model Identification The process of data generation for model identification robustness checks was similar to parameter recovery. However, we only simulated 500 trials for each model because we found no significant increase in accuracy with higher trial numbers.

4.6 Acknowledgments

We thank Jasmine Collins, Kshitiz Gupta, Yi Liu and Jaeyoung Park for their contributions to the project. We thank Bill Thompson and all CCN lab members for useful feedback on the project. This work was supported by NIH R21MH132974.
Code available at the GitHub Repository.

4.7 Acknowledgments

We thank Jasmine Collins, Kshitiz Gupta, Yi Liu and Jaeyoung Park for their contributions to the project. We thank Bill Thompson and all CCN lab members for useful feedback on the project. This work was supported by NIH R21MH132974.
Code available at the GitHub Repository.

4.8 Supplementary Materials

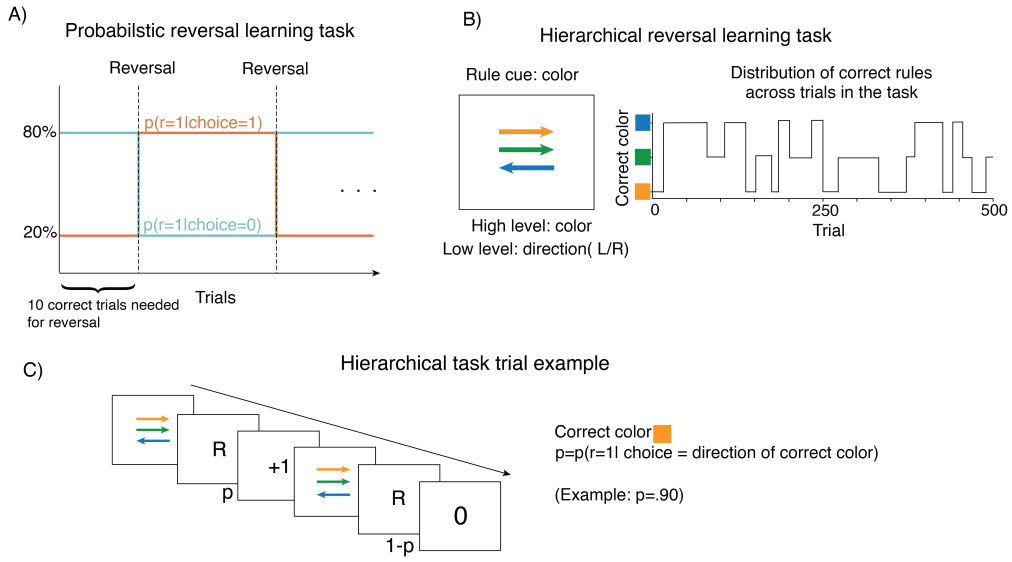


Figure 4.7: Tasks. A) Probabilistic reversal learning task. We simulated artificial agents using cognitive models of behavior on a Probabilistic reversal learning (PRL) task, which provides a dynamic context for studying reward-driven learning (Cools et al., 2001; Lawrence et al., 1999). In this task, an agent chooses between two actions, where one of the actions gets rewarded with higher probability ($p(r) = .80$) and one with lower ($1 - p$). After a certain number of correct trials, the reward probabilities of the two actions reverse. The task provides an opportunity to observe how agents update their model of the task (e.g. correct actions) based on observed feedback. B) Hierarchical reinforcement learning task. In this task, three differently colored arrows represent three potential rules an agent can follow when selecting one of the two actions (left/right) corresponding to the side the chosen arrow is pointing at. Selecting a side consistent with correct arrow is rewarded with probability $p = .90$. Correct arrow switches after a certain number of trials. The task provides a possibility to examine how following latent rules may shape agents' learning behavior.

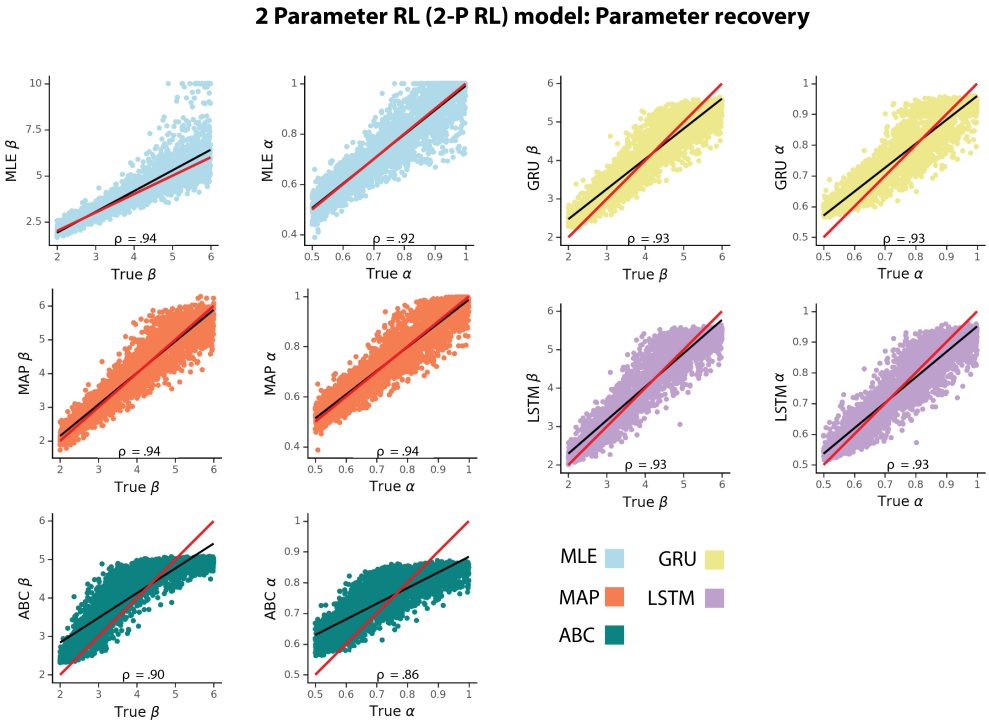


Figure 4.8: 2 Parameter RL (2P-RL) model parameter recovery using different fitting methods. ρ corresponds to Spearman correlation coefficient, red line represents a unity line (x=y), and black line represents a least squares regression line.

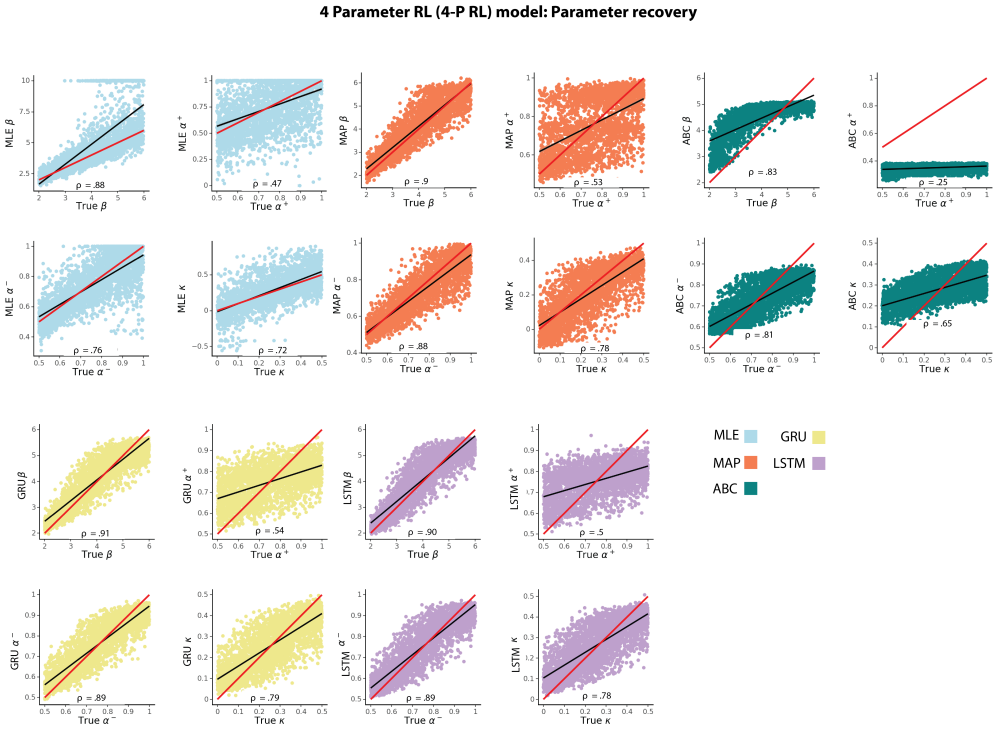


Figure 4.9: 4 Parameter RL model (4P-RL) parameter recovery using different fitting methods. ρ corresponds to Spearman correlation coefficient, red line represents a unity line (x=y), and black line represents a least squares regression line.

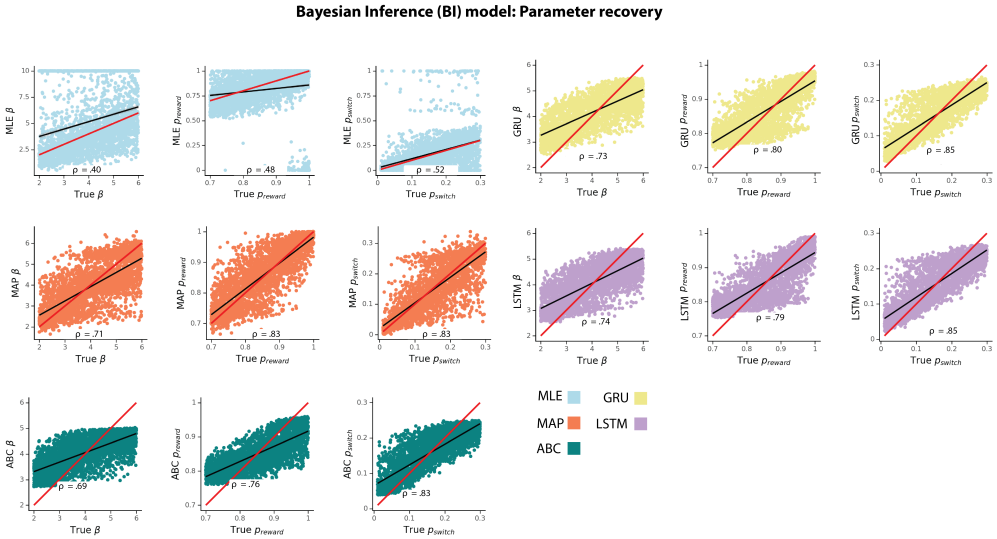


Figure 4.10: Bayesian Inference (BI) model parameter recovery using different fitting methods. ρ corresponds to Spearman correlation coefficient, red line represents a unity line (x=y), and black line represents a least squares regression line.

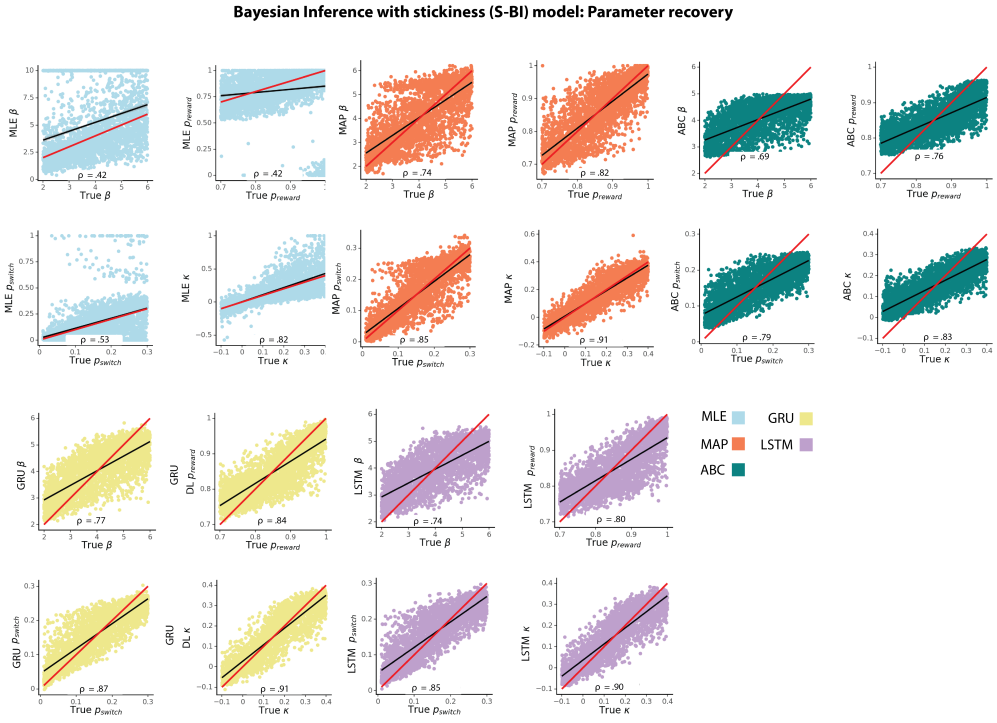


Figure 4.11: Bayesian Inference with stickiness (S-BI) model parameter recovery using different fitting methods. ρ corresponds to Spearman correlation coefficient, red line represents a unity line (x=y), and black line represents a least squares regression line.

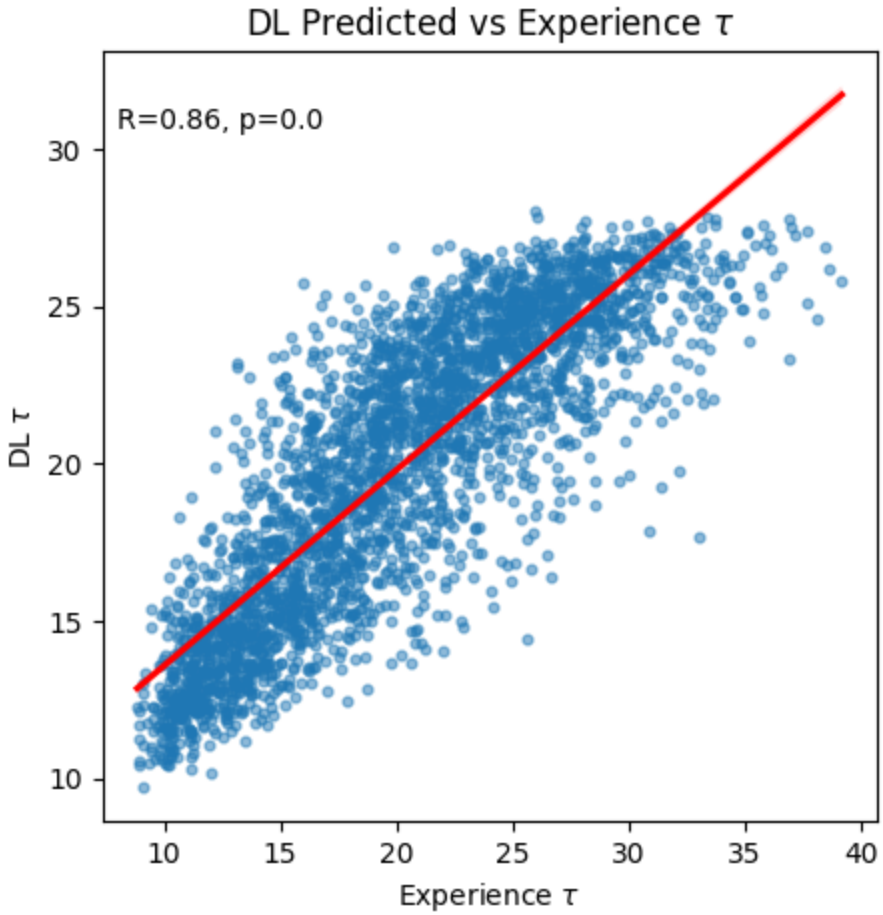


Figure 4.12: Correlation between the average experienced time intervals in attentive state and the τ parameter in RL-LAS model that captures transition between disengaged/engaged attention states estimated by the ANN.

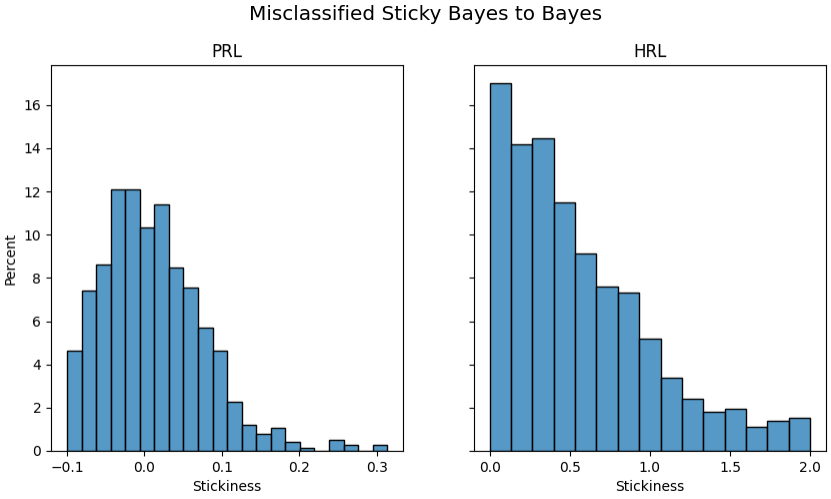


Figure 4.13: Misclassification of Bayes and sticky Bayes model is contingent on the value of the stickiness parameter κ. The misclassification percentage is higher at κ values closer to 0.

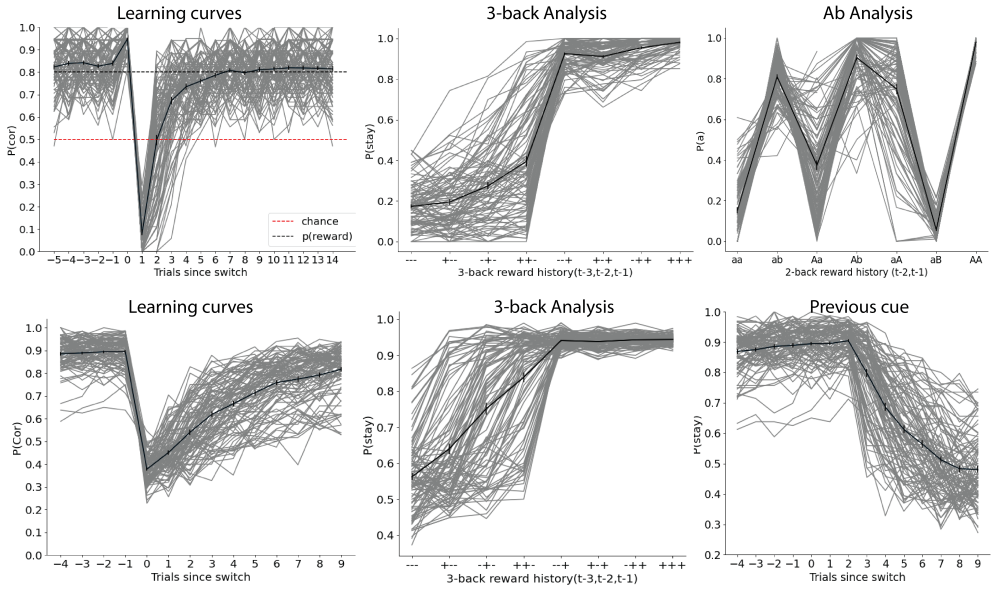


Figure 4.14: Summary statistics for Approximate Bayesian Computation (ABC). Top row shows summary statistics computed for all models simulated on a probabilistic reversal learning task; the figure only shows agents simulated using a 4-parameter RL model. Bottom row shows summary statistics computed for all models simulated on a hierarchical reversal learning task; the figure only shows performance of HRL model agents. Both rows depict 200 out of 3000 test set agents. Gray lines represent individual agents; black line represents an average across the agents.

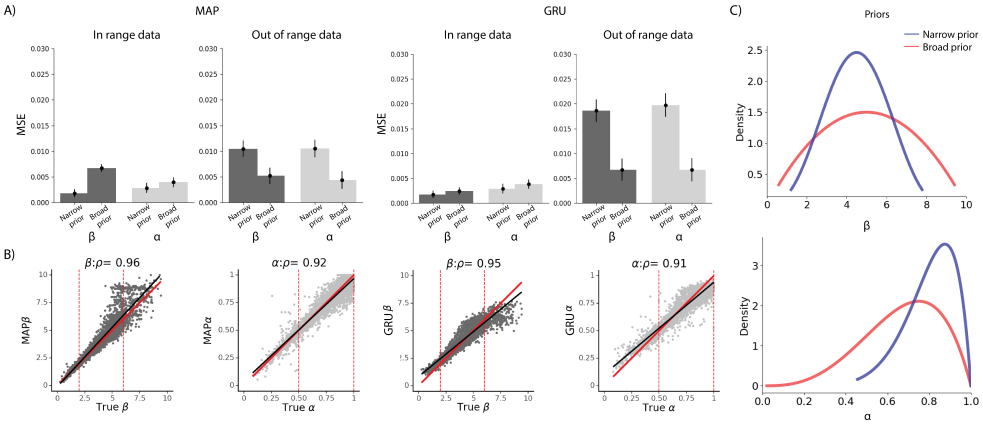


Figure 4.15: Effect of prior misspecification on parameter estimation in MAP and our ANN approach. A) Applying too narrow prior specification to the fitting procedure (prior in MAP, training samples in ANN) results in difficulty estimating out-of-range parameters for both MAP and ANN. Broader prior specification addresses this issue, with only a slight loss of precision in specific target ranges. Training the network with a broad range of parameters while oversampling parameters from regions of interest yields most robust results. B) Visualization of fitting with MAP and ANN with a wide prior, tested on a full range/wide range data set - training the network with broader range while oversampling from the most plausible range yields less noisy performance in the range compared to MAP. Red lines delineate the range of the narrow prior, which corresponds to the main text results. C) The broad prior was designed by sampling from the full broader range ($\beta \in [0, 10]$, $\alpha \in [0, 1]$), with the constraint that 70% of samples are in the expected narrow range ($\beta \in [2, 6]$, $\alpha \in [0.5, 1]$, and 30% outside.)

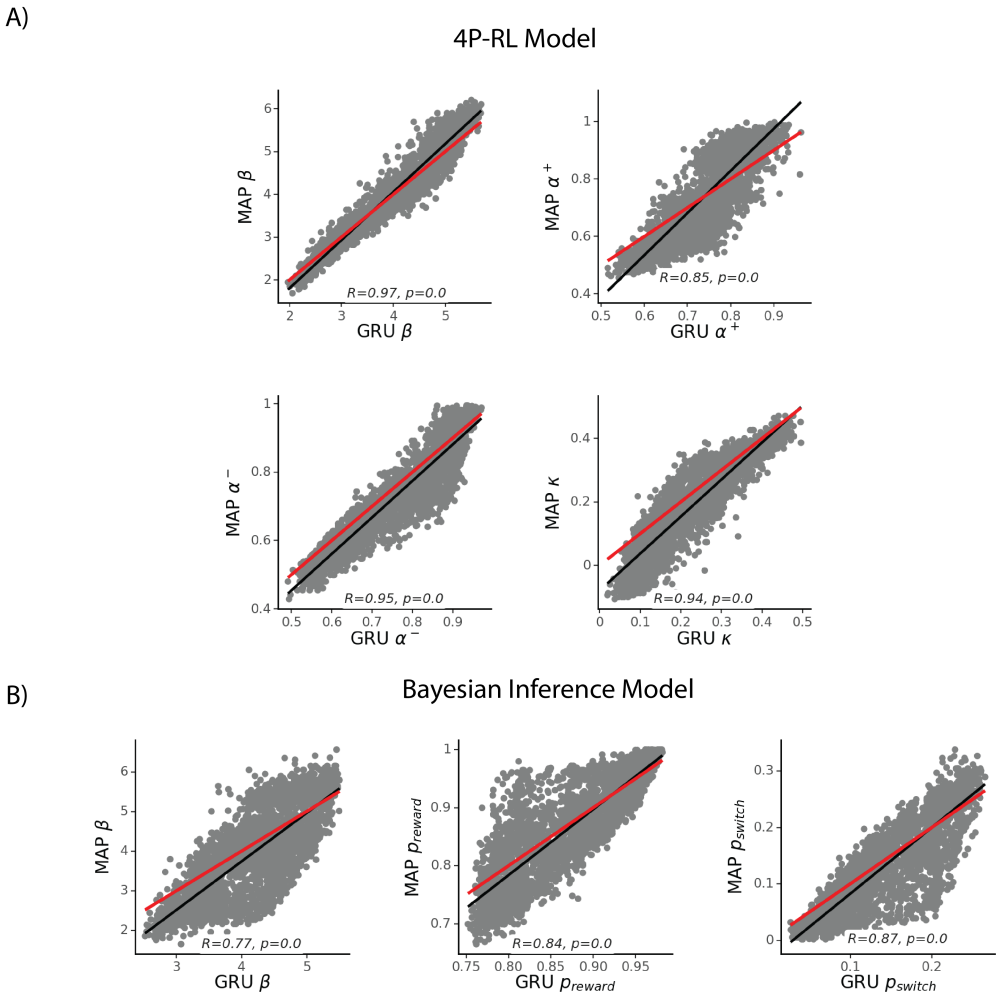


Figure 4.16: Consistency between methods for parameter estimation. A) The correlation between parameter estimates in the 4P-RL model derived using MAP and ANN, is high, and indeed stronger than the correlation between true and derived parameters (see Fig. 4.2), showing that both methods systematically misidentify some parameters similarly, likely due to specific data patterns. B) The correlation between parameter estimates in the Bayesian inference model derived using MAP and ANN shows similar results.

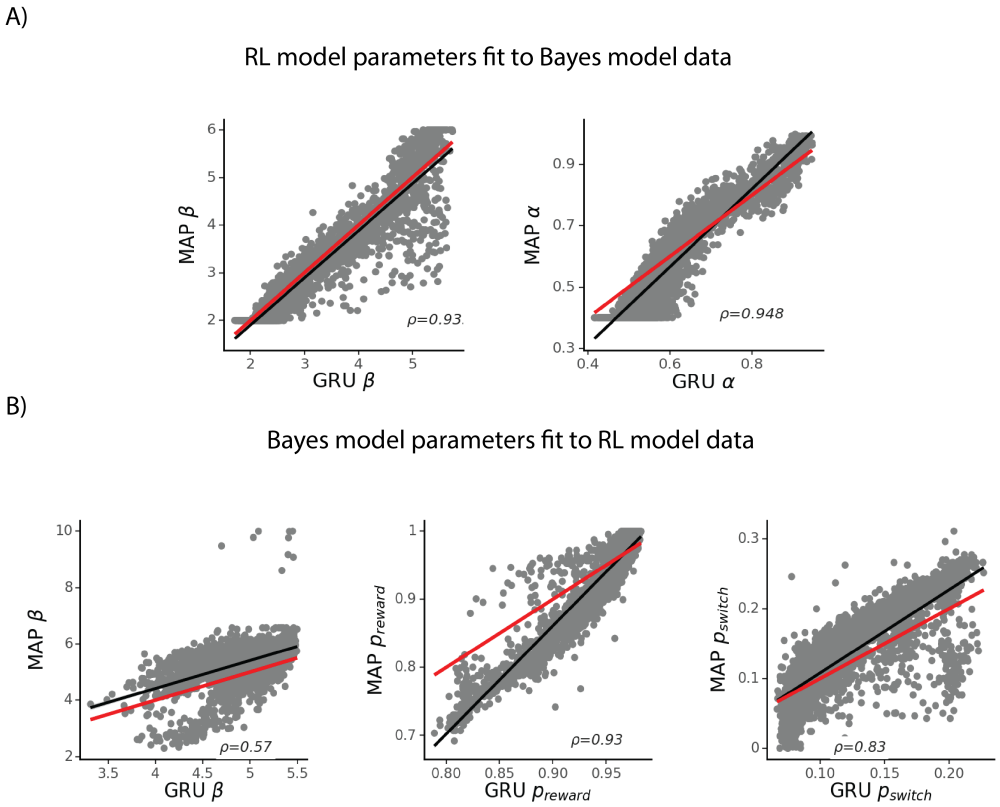


Figure 4.17: Consistency between methods for parameter estimation in two model misspecification cases. A) The correlation between MAP and GRU RL model parameter estimates, fit to data simulated from Bayesian Inference model. B) The correlation between MAP and GRU Bayes model parameter estimates, fit to data simulated from the RL model. High correlation would imply similarities in estimates between MAP and GRU, suggesting that ANNs are similarly impacted by model misspecification as traditional methods such as MAP.

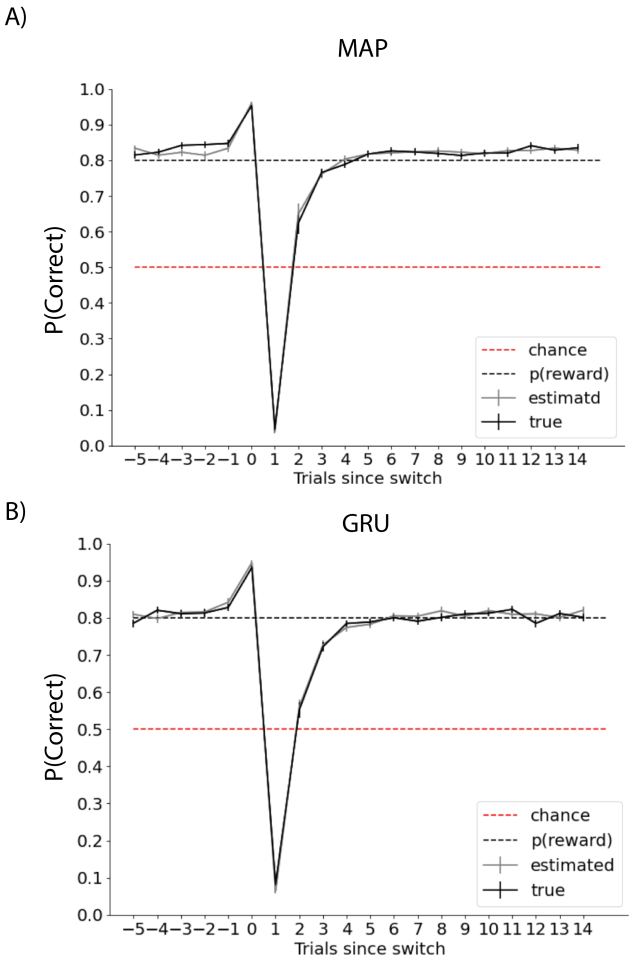


Figure 4.18: Comparison of model predictions of ground truth simulated behavior (black line) and choices simulated using A) MAP and B) GRU estimated parameters (gray line) of the 4P-RL model. We randomly sampled 100 agents from the test set, and the respective parameter estimates for each of the methods. We simulated data from the model and compared it to ground truth. Both methods successfully recover choices from the ground truth agents.

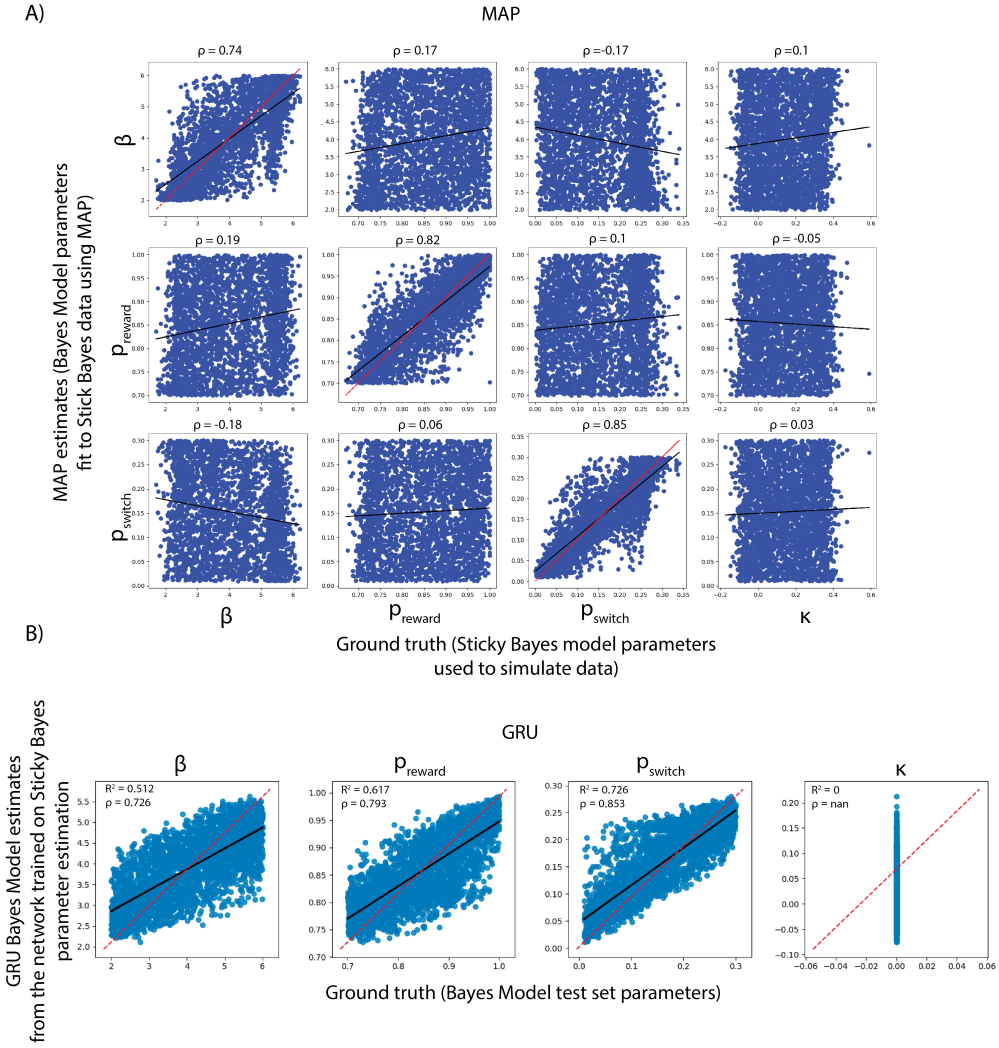


Figure 4.19: Effect of model misspecification on standard method and ANN performance. A) We fit the Bayesian Inference model (without stickiness) to the data simulated from the Bayesian Inference Model with stickiness using MAP. We correlated the estimated Bayesian inference model parameters (y axis) with the ground truth parameters from the model with stickiness (x-axis). B) For the ANN, we trained the neural network to estimate parameters of the Bayesian inference model, and tested it on the data simulated from the Bayesian inference model with stickiness. We looked at the correlation between the ground truth parameters (from a separate test set), and the predictions of the network trained on the model without stickiness. Both methods show that parameters shared between the misspecified models can be reasonably and similarly recovered. Both ANN and MAP generated some non-zero estimates of stickiness when data simulated from model without stickiness was fit using the model/network that assumes presence of stickiness in the model; however, these values were closely clustered around 0, to a similar degree between methods (Fig. 4.22).

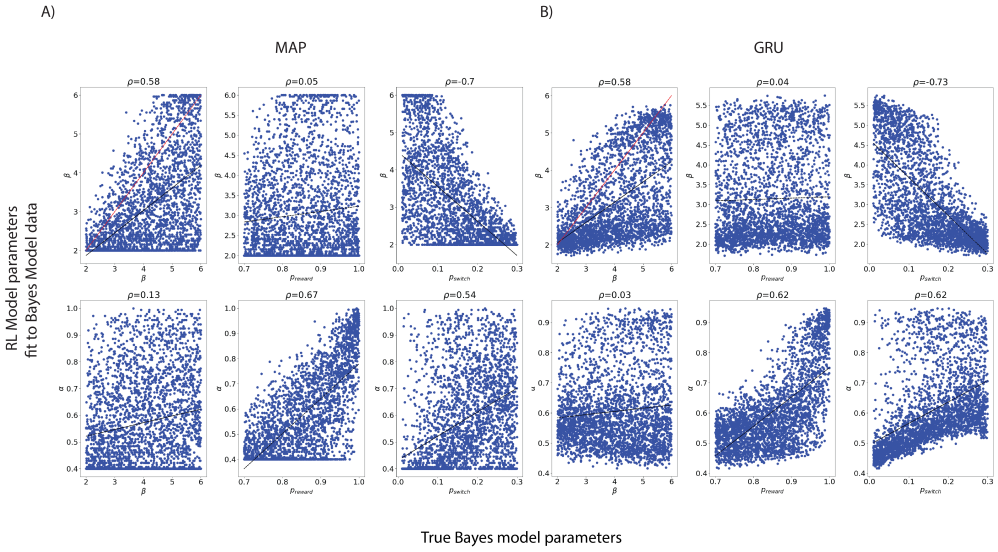


Figure 4.20: Effect of model class misspecification on standard method and ANN performance. We fit the RL model to the data simulated from the Bayesian Inference model in the probabilistic reversal learning task (see Methods section on Tasks and Cognitive Models) using A) MAP and B) GRU. We correlated the estimated RL model parameters (y axis) with the ground truth parameters from the Bayesian inference model (x-axis). MAP and GRU show similar patterns between estimated and true parameters, such that variance driven by true parameter β p_{switch} are both captured in the fit β parameters, while the fit learning rate parameter α captures behavioral variance driven by the Bayesian update parameters p_{reward} and p_{switch}.

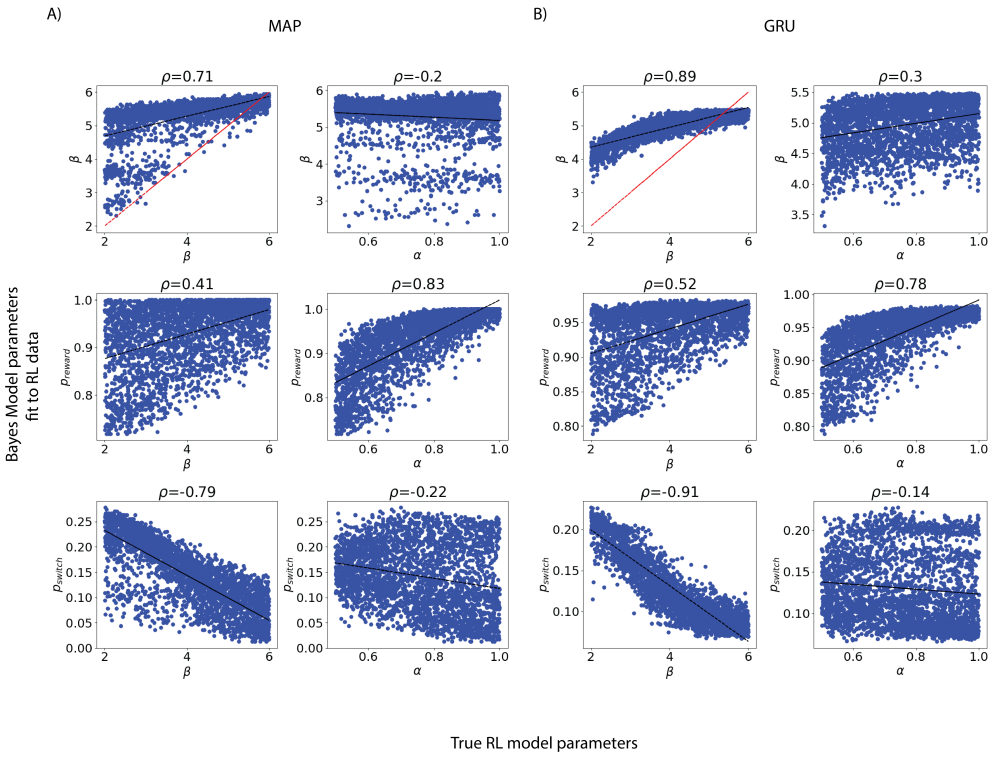


Figure 4.21: Effect of model class misspecification on standard method and ANN performance. We fit the Bayesian Inference model to the data simulated from the RL model in the probabilistic reversal learning task (see Methods section on Tasks and Cognitive Models) using A) MAP and B) GRU. We correlated the estimated Bayesian inference model parameters (y axis) with the ground truth parameters from the RL model (x-axis). MAP and GRU again show similar patterns between estimated and true parameters. In particular, we see that in both cases, noise in behavior due to β in the RL model tends to be attributed to the p_{switch} fit parameter rather than the fit Bayesian model β parameter. Effect of learning rate parameter α are attributed to p_{reward} by both methods.

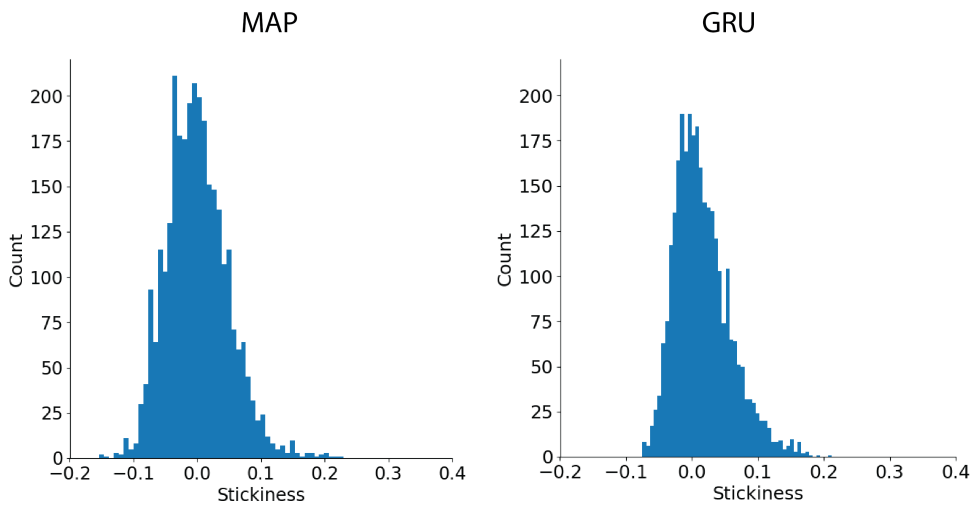


Figure 4.22: Stickiness parameter estimates from the data simulated from the Bayesian inference model without stickiness from A) fitting the Bayesian inference model with stickiness using MAP, and B) utilizing the ANN trained to estimate parameters of the model with stickiness. Despite both methods producing non-zero estimates of stickiness, they tend to cluster around the value of 0.

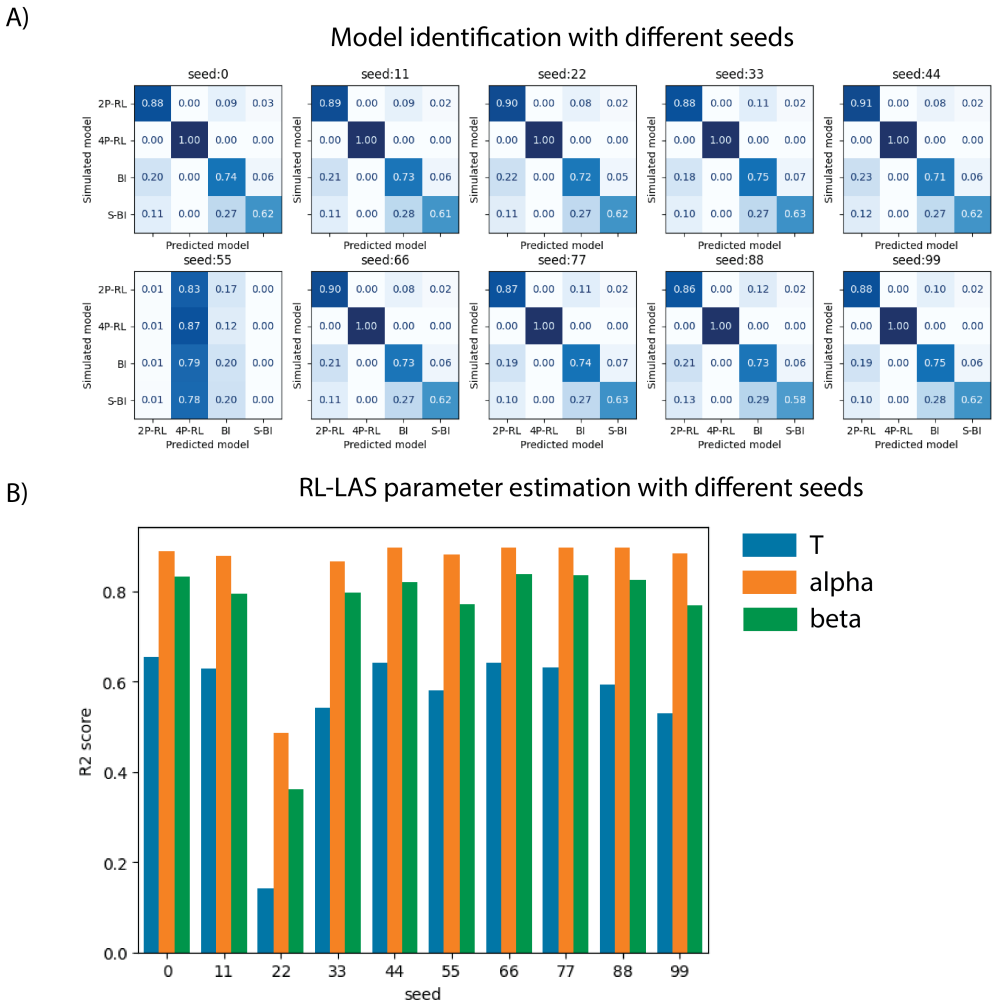


Figure 4.23: Neural network performance variability by different seeds for model identification and parameter estimation. For conciseness, we show tests from 10 different seeds on model identification with 4 models simulated on the PRL task (e.g. same as Fig. 4.5) and parameter estimation of one of the likelihood-intractable models (e.g. RL-LAS model). We found that overall both model identification and parameter estimation had relatively stable results across different seeds, with an exception of one seed value in both cases.

Tasks	Parameter Recovery				Model Identification		
Cognitive Models	2P-RL	4P-RL	RL-LAS	HRL	PRL	2P-RL & RL-LAS	HRL
Batch size	256	256	256	256	128	128	128
GRU units	128	90	256	256	151	151	151
1st dense units	64	45	128	128	75	75	75
2nd dense units	32	22	64	64	37	37	37
Dropout after RNN	0.2	0.3	0.2	0.2	0.187	0.187	0.187
2nd dropout	0.1	0.2	0.1	0.1	0.04	0.04	0.04
3rd dropout	0.01	0.05	0.02	0.01	0.02	0.02	0.02

Table 4.1: Summary of hyper-parameter values selected from Bayesian optimization algorithms.

Chapter 5

Conclusions

5.1 Effect of choice abstraction on reinforcement learning and working memory

Reinforcement learning algorithms applied in cognitive models often assume a strictly defined action space, and a direct mapping between stimuli and simple actions. However, choice spaces are often more complex and often involve layered execution, such as choosing the high level, goal option (e.g. which yogurt color to select) which constrains the motor action execution (reaching in the direction of chosen color). Oftentimes, choices at these different levels are treated the same in computational models. The project outlined in chapter 2 tested variability in computational mechanisms of reinforcement learning in conditions with more/less abstract choice spaces, and found that 1) less abstract choices (e.g. motor actions) tend to interfere with more abstract choices (e.g. choice of a goal stimulus), and 2) a model that accounts for working memory contribution to reinforcement learning indicated that working memory deployment (quantified by the working memory weight paramter) is reduced in more abstract choice condition. This has implications for how we understand choice spaces as an integral part of reinforcement learning algorithms. First, different choice types are dissociable, and may recruit RL mechanisms differently, and this should merit more careful consideration when defining how the choice process is modeled in RL algorithms. Second, working memory resources might be used up to constrain a more abstract choice space, which results in reduced contribution of working memory to the actual choice process. Our work, however, did not identify RL or WM as the source of interference of motor actions with stimulus goal selection, with lack of evidence in favor of models that place policy mixture parameter selectively in RL or WM module. Future work is required to investigate this in closer detail.

Identification of relevant features of the choice space is a critical building block of correct credit assignment and adaptive behavior. Failure to identify what feature of choice space defines rewarding responses would likely result in suboptimal and maladaptive behav-

ioral patterns. The finding that the certain types of choice interfere with others, leading to increased rate of interference errors and erroneous selections may be characteristic of perseverative responses and inability to disengage from certain actions commonly observed in obsessive-compulsive disorder. Future work could implement this design, or its variation, to explore the effect of choice abstraction and relevant WM/RL mechanisms in choice process in clinical populations.

In addition to explaining human behavior, this paradigm could be used to explore the differences in strategies human and artificial agents implement in service of goal-directed behavior. We recently explored how a large language model (GPT 3.5) solves the task comparable to position condition administered in experiment 2 (e.g. simple actions with varying set size). We observed that the large language model exploited associations irrelevant to the task structure when making choices. For instance, if the instructions stated to choose one of the actions labeled by a digit (action 1, action 2 or action 3) in responses to each of the stimuli also labeled by digits (e.g. stimulus 1, stimulus 2, stimulus 3), it tended to match actions and stimuli based on their digit labels (e.g. choosing action 1 for stimulus 1, action 2 for stimulus 2, etc.) even when this was not a correct mapping, occasionally requiring a lot of training to override this bias. Updated version of the model (GPT 4) did not show this tendency, but the overall result may have a potential implication for the extent to which artificial agents can be likened to humans in ways they solve problems, especially with the rising popularity in applications of artificial intelligence and large language models.

This work represents an addition to the line of research that attempts to outline a more holistic picture of reinforcement learning in cognition, including how the computational mechanisms of RL trades off with different cognitive processes, including the executive function. More specific applications (e.g. with regards to clinical populations and artificial agents) may be possible.

5.2 Do we need subgoals for generalization?

Chapter 3 presented work looking at how subgoals are used to hierarchically organize simple action sequences into complex policies in service of reward collection in the context of hierarchical reinforcement learning. We have defined subgoals in a way that decouples them from reward values and internal motivators (e.g. curiosity drive from novelty/surprise). We found that participants showed evidence of using subgoals to solve the task, and through the regression analysis we found that tokens used to represent subgoals impacted participants' accuracy, and their tendency to repeat action sequences. This implies that subgoals can exert pseudo-reinforcing effects on performance and choices, independently of factors we controlled for. We also tested if the subgoals are successfully generalized to separate tasks. We found that only a subset of our participant sample (one third) showed evidence of subgoal generalization, under our subgoal specifications. Participants who did show evidence of subgoal generalization were able to identify the subgoal features when explicitly probed

to do so; they also displayed preference for subgoal over non-subgoal features, which led us to conclude that subgoal generalization, under our definition of subgoals, is plausible but predicated on explicit recognition of subgoal features.

We explored the role of subgoals in hierarchy in the context of learning; however, hierarchical organization is present in other domains - such as language. Language can be viewed as hierarchically organized due to its compositional structure (e.g. letters construct words, words construct sentences, sentences construct paragraphs, etc.). Therefore, subgoals may be present in the language domain as well, in form of a type of word, or its position in the sentence, that may signal termination of one syntactic/semantic unit and initiation of a new one (much like action sub-sequences). These insights could prove to be valuable in streamlining language instructions for artificial agents (e.g. in large language models, where identifying proper ways of instructing the model remains a challenge), where the structure organized around subgoals may improve comprehension of instructions, and better translation into direct output.

5.3 The future of cognitive modeling

Chapter 4 focused on method development aimed to construct a tool that will enable fitting computational cognitive models with intractable likelihoods and strong sequential dependencies by leveraging the power of artificial neural networks. This approach is based on feeding a large set of simulated data sequences to and training the neural network to estimate model parameters, or identify which model the data was simulated from without the need to compute model likelihood. The simple neural network architecture consisting of an RNN (recurrent neural network) and densely connected layers was successfully applied to model parameter recovery and model identification for various models - both tractable and intractable. Furthermore, our tool allows for a simple add-on of evidential learning that permits quantification of parameter estimate uncertainty. In addition, the neural network approach is not computationally expensive and provides a fast way to relate cognitive models to the data.

In our project we focused on basic properties of cognitive modeling - parameter estimation and model identification. However, with tools such as neural networks, alternative uses in service of cognitive modeling may be possible. For instance, instead of model parameter estimation/model identification, future projects could focus on extracting agent's latent task-solving strategies from data sequences which may evolve throughout the task. Such application may be possible with an alternative neural network architecture that permits a sequence-to-sequence translation, whereby instead of single parameter estimate or model label for the agent we obtain a sequence of strategies the agent deployed to solve the task across trials.

Recently, neural networks have been applied to cognitive modeling in a different way, departing from fitting hand-crafted cognitive models to the data and instead using ANNs

as hybrid cognitive model "replacements" (Eckstein et al., 2023). The motivation behind this approach is to exploit advantages of both cognitive modeling and ANNs (interpretable outputs and flexibility respectively), while mitigating their limitations (potential poor fit to the data, and opaque, black-box mechanisms that are not interpretable). This approach essentially permits substitution of model functions with ANNs, and extraction of information (e.g. learning rules) beyond what is specified by the cognitive model (e.g. by inspecting neural network weights).

It is evident that, whether through merely substituting model-fitting tools or augmenting cognitive models by offering additional insights into processes underlying the behavioral data, artificial neural networks are increasingly taking an important place in cognitive modeling research. The properties of ANNs, specifically their flexibility and capacity for capturing complex data patterns, make them an attractive tool for cognitive modeling that many researchers might gravitate towards. While it may be expected that the literature leveraging ANNs for cognitive models may expand in the future, it will be important for the researchers to exercise caution when applying ANNs, be cognizant of their limitations, as well interpret the results with caution (considering both the practices of applying the ANNs, and the assumptions about tested cognitive theories).

5.4 Summary

We completed three projects with the goal of outlining computational mechanisms of reinforcement learning in human cognition - in the context of hierarchy and interaction with other learning systems in settings that challenge basic RL premises. We also devoted a project to developing tools that will help us, and other cognitive researchers, test a broader range of cognitive models by replacing traditional model-fitting techniques with neural networks. Results combined across the first 2 projects imply that while basic RL algorithms have high utility and have been of immense importance in cognitive science, they pose restrictive assumptions and fail to account for many robust learning patterns - some of which could be accounted for by modeling an interaction with other learning mechanisms (including that of working memory) and leveraging hierarchical representations. Our third project offers a simple and light-weight approach to relating cognitive models with complex intractable likelihoods to behavioral data through neural networks. This project aims to broaden the range of testable cognitive theories, including those previously deemed infeasible due to intractable likelihood, that might offer better explanations of learning processes.

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). {Tensorflow}: A system for {large-scale} machine learning. *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 265–283.

Abbeel, P., Coates, A., Quigley, M., & Ng, A. (2006). An application of reinforcement learning to aerobatic helicopter flight. *Advances in neural information processing systems*, *19*.

Acerbi, L., & Ma, W. J. (2017). Practical bayesian optimization for model fitting with bayesian adaptive direct search. *Advances in neural information processing systems*, *30*.

Adams, R. A., Huys, Q. J., & Roiser, J. P. (2016). Computational psychiatry: Towards a mathematically informed understanding of mental illness. *Journal of Neurology, Neurosurgery & Psychiatry*, *87*(1), 53–63.

Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. *Selected papers of hirotugu akaike*, 199–213.

Amini, A., Schwarting, W., Soleimany, A., & Rus, D. (2020). Deep evidential regression. *Advances in Neural Information Processing Systems*, *33*, 14927–14937.

Ashwood, Z. C., Roy, N. A., Stone, I. R., Laboratory, I. B., Urai, A. E., Churchland, A. K., Pouget, A., & Pillow, J. W. (2022). Mice alternate between discrete strategies during perceptual decision-making. *Nature Neuroscience*, *25*(2), 201–212.

Atallah, H. E., Frank, M. J., & O'Reilly, R. C. (2004). Hippocampus, cortex, and basal ganglia: Insights from computational models of complementary learning systems. *Neurobiology of learning and memory*, *82*(3), 253–267.

Badre, D., & D'esposito, M. (2009). Is the rostro-caudal axis of the frontal lobe hierarchical? *Nature reviews neuroscience*, *10*(9), 659–669.

Badre, D., & Nee, D. E. (2018). Frontal cortex and the hierarchical control of behavior. *Trends in cognitive sciences*, *22*(2), 170–188.

Baldassarre, G., & Mirolli, M. (2013). Intrinsically motivated learning systems: An overview. *Intrinsically motivated learning in natural and artificial systems*, 1–14.

Ballard, I., Miller, E. M., Piantadosi, S. T., Goodman, N. D., & McClure, S. M. (2018). Beyond reward prediction errors: Human striatum updates rule values during learning. *Cerebral Cortex*, *28*(11), 3965–3975.

Baribault, B., & Collins, A. G. (2023). Troubleshooting bayesian cognitive models. *Psychological Methods*.

Bergstra, J., Yamins, D., & Cox, D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *International conference on machine learning*, 115–123.

Beron, C., Neufeld, S., Linderman, S., & Sabatini, B. (2021). Efficient and stochastic mouse action switching during probabilistic decision making. *Neuroscience*, *10*(2021.05), 13–444094.

Boelts, J., Lueckmann, J.-M., Gao, R., & Macke, J. H. (2022). Flexible and efficient simulation-based inference for models of decision-making. *Elife*, *11*, e77220.

Bornstein, A. M., & Daw, N. D. (2013). Cortical and hippocampal correlates of deliberation during model-based decisions for rewards in humans. *PLoS computational biology*, *9*(12), e1003387.

Bornstein, A. M., Khaw, M. W., Shohamy, D., & Daw, N. D. (2017). Reminders of past choices bias decisions for reward in humans. *Nature Communications*, *8*(1), 15958.

Botvinick, M. M. (2008). Hierarchical models of behavior and prefrontal function. *Trends in cognitive sciences*, *12*(5), 201–208.

Botvinick, M. M., Niv, Y., & Barto, A. G. (2009). Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *cognition*, *113*(3), 262–280.

Bromberg-Martin, E. S., Matsumoto, M., & Hikosaka, O. (2010). Dopamine in motivational control: Rewarding, aversive, and alerting. *Neuron*, *68*(5), 815–834.

Bunge, S. A. (2024). How should we slice up the executive function pie? striving toward an ontology of cognitive control processes. *Mind, Brain, and Education*.

Chen, C., Takahashi, T., Nakagawa, S., Inoue, T., & Kusumi, I. (2015). Reinforcement learning in depression: A review of computational research. *Neuroscience & Biobehavioral Reviews*, *55*, 247–267.

Chen, Y., Zhang, D., Gutmann, M., Courville, A., & Zhu, Z. (2020). Neural approximate sufficient statistics for implicit models. *arXiv preprint arXiv:2010.10079*.

Chentanez, N., Barto, A., & Singh, S. (2004). Intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, *17*.

Collins, A. G. (2018). The tortoise and the hare: Interactions between reinforcement learning and working memory. *Journal of cognitive neuroscience*, *30*(10), 1422–1432.

Collins, A. G., Brown, J. K., Gold, J. M., Waltz, J. A., & Frank, M. J. (2014). Working memory contributions to reinforcement learning impairments in schizophrenia. *Journal of Neuroscience*, *34*(41), 13747–13756.

Collins, A. G., Ciullo, B., Frank, M. J., & Badre, D. (2017). Working memory load strengthens reward prediction errors. *Journal of Neuroscience*, *37*(16), 4332–4342.

Collins, A. G., & Cockburn, J. (n.d.). Beyond simple dichotomies in reinforcement learning.

Collins, A. G., & Frank, M. J. (2012). How much of reinforcement learning is working memory, not reinforcement learning? a behavioral, computational, and neurogenetic analysis. *European Journal of Neuroscience*, *35*(7), 1024–1035.

Collins, A. G., & Frank, M. J. (2013). Cognitive control over learning: Creating, clustering, and generalizing task-set structure. *Psychological review*, *120*(1), 190.

Collins, A. G., & Frank, M. J. (2018). Within-and across-trial dynamics of human eeg reveal cooperative interplay between reinforcement learning and working memory. *Proceedings of the National Academy of Sciences*, *115*(10), 2502–2507.

Cools, R., Barker, R. A., Sahakian, B. J., & Robbins, T. W. (2001). Enhanced or impaired cognitive function in parkinson's disease as a function of dopaminergic medication and task demands. *Cerebral cortex*, *11*(12), 1136–1143.

Cools, R., Clark, L., Owen, A. M., & Robbins, T. W. (2002). Defining the neural mechanisms of probabilistic reversal learning using event-related functional magnetic resonance imaging. *Journal of Neuroscience*, *22*(11), 4563–4567.

Costa, V. D., Tran, V. L., Turchi, J., & Averbeck, B. B. (2015). Reversal learning and dopamine: A bayesian perspective. *Journal of Neuroscience*, *35*(6), 2407–2416.

Cousineau, D., & Helie, S. (2013). Improving maximum likelihood estimation using prior probabilities: A tutorial on maximum a posteriori estimation and an examination of the weibull distribution. *Tutorials in Quantitative Methods for Psychology*, *9*(2), 61–71.

Cranmer, K., Brehmer, J., & Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, *117*(48), 30055–30062.

Daw, N. D., Gershman, S. J., Seymour, B., Dayan, P., & Dolan, R. J. (2011). Model-based influences on humans' choices and striatal prediction errors. *Neuron*, *69*(6), 1204–1215.

Dayan, P., & Daw, N. D. (2008). Decision theory, reinforcement learning, and the brain. *Cognitive, Affective, & Behavioral Neuroscience*, *8*(4), 429–453.

Dayan, P., & Hinton, G. E. (1992). Feudal reinforcement learning. *Advances in neural information processing systems*, *5*.

De Leeuw, J. R. (2015). Jspsych: A javascript library for creating behavioral experiments in a web browser. *Behavior research methods*, *47*, 1–12.

Decker, J. H., Otto, A. R., Daw, N. D., & Hartley, C. A. (2016). From creatures of habit to goal-directed learners: Tracking the developmental emergence of model-based reinforcement learning. *Psychological science*, *27*(6), 848–858.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dezfouli, A., Ashtiani, H., Ghattas, O., Nock, R., Dayan, P., & Ong, C. S. (2019). Disentangled behavioural representations. *Advances in neural information processing systems*, *32*.

Diuk, C., Schapiro, A., Córdova, N., Ribas-Fernandes, J., Niv, Y., & Botvinick, M. (2013). Divide and conquer: Hierarchical reinforcement learning and task decomposition in

humans. *Computational and robotic models of the hierarchical organization of behavior*, 271–291.

Djuric, P. M., Kotecha, J. H., Zhang, J., Huang, Y., Ghirmai, T., Bugallo, M. F., & Miguez, J. (2003). Particle filtering. *IEEE signal processing magazine*, *20*(5), 19–38.

Eckstein, M. K., & Collins, A. G. (2020). Computational evidence for hierarchically structured reinforcement learning in humans. *Proceedings of the National Academy of Sciences*, *117*(47), 29381–29389.

Eckstein, M. K., & Collins, A. G. (2021). How the mind creates structure: Hierarchical learning of action sequences. *CogSci... Annual Conference of the Cognitive Science Society. Cognitive Science Society (US). Conference*, *43*, 618.

Eckstein, M. K., Master, S. L., Dahl, R. E., Wilbrecht, L., & Collins, A. G. (2022). Reinforcement learning and bayesian inference provide complementary models for the unique advantage of adolescents in stochastic reversal. *Developmental Cognitive Neuroscience*, *55*, 101106.

Eckstein, M. K., Starr, A., & Bunge, S. A. (2019). How the inference of hierarchical rules unfolds over time. *Cognition*, *185*, 151–162.

Eckstein, M. K., Summerfield, C., Daw, N. D., & Miller, K. J. (2023). Predictive and interpretable: Combining artificial neural networks and classic cognitive models to understand human learning and decision making. *bioRxiv*, 2023–05.

Eckstein, M. K., Wilbrecht, L., & Collins, A. G. (2021). What do reinforcement learning models measure? interpreting model parameters in cognition and neuroscience. *Current Opinion in Behavioral Sciences*, *41*, 128–137.

Eckstein, M. K., Master, S. L., Xia, L., Dahl, R. E., Wilbrecht, L., & Collins, A. G. (2022). The interpretation of computational model parameters depends on the context. *Elife*, *11*, e75474.

Eppinger, B., Walter, M., Heekeren, H. R., & Li, S.-C. (2013). Of goals and habits: Age-related and individual differences in goal-directed decision-making. *Frontiers in neuroscience*, *7*, 253.

Farashahi, S., Rowe, K., Aslami, Z., Lee, D., & Soltani, A. (2017). Feature-based learning improves adaptability without compromising precision. *Nature communications*, *8*(1), 1768.

Fearnhead, P., & Prangle, D. (2012). Constructing summary statistics for approximate bayesian computation: Semi-automatic approximate bayesian computation. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *74*(3), 419–474.

Fengler, A., Govindarajan, L. N., Chen, T., & Frank, M. J. (2021). Likelihood approximation networks (lans) for fast inference of simulation models in cognitive neuroscience. *Elife*, *10*, e65074.

Findling, C., Chopin, N., & Koechlin, E. (2021). Imprecise neural computations as a source of adaptive behaviour in volatile environments. *Nature Human Behaviour*, *5*(1), 99–112.

Fleuret, F., & Geman, D. (2001). Coarse-to-fine face detection. *International Journal of computer vision*, *41*, 85–107.

Foerde, K., & Shohamy, D. (2011). The role of the basal ganglia in learning and memory: Insight from parkinson's disease. *Neurobiology of learning and memory*, *96*(4), 624–636.

Frank, M. J., & Badre, D. (2012). Mechanisms of hierarchical reinforcement learning in corticostriatal circuits 1: Computational analysis. *Cerebral cortex*, *22*(3), 509–526.

Frank, M. J., Moustafa, A. A., Haughey, H. M., Curran, T., & Hutchison, K. E. (2007). Genetic triple dissociation reveals multiple roles for dopamine in reinforcement learning. *Proceedings of the National Academy of Sciences*, *104*(41), 16311–16316.

Friedman, N. P., & Robbins, T. W. (2022). The role of prefrontal cortex in cognitive control and executive function. *Neuropsychopharmacology*, *47*(1), 72–89.

Ger, Y., Nachmani, E., Wolf, L., & Shahar, N. (2023). Harnessing the flexibility of neural networks to predict dynamic theoretical parameters underlying human choice behavior. *bioRxiv*, 2023–04.

Gershman, S. J. (2015). Do learning rates adapt to the distribution of rewards? *Psychonomic bulletin & review*, *22*, 1320–1327.

Gershman, S. J., & Daw, N. D. (2017). Reinforcement learning and episodic memory in humans and animals: An integrative framework. *Annual review of psychology*, *68*, 101–128.

Ghaderi-Kangavari, A., Rad, J. A., & Nunez, M. D. (2023). A general integrative neurocognitive modeling framework to jointly describe eeg and decision-making on single trials. *Computational Brain & Behavior*, *6*(3), 317–376.

Gilbert, S. J., & Burgess, P. W. (2008). Executive function. *Current biology*, *18*(3), R110–R114.

Gillan, C. M., Kosinski, M., Whelan, R., Phelps, E. A., & Daw, N. D. (2016). Characterizing a psychiatric symptom dimension related to deficits in goal-directed control. *elife*, *5*, e11305.

Gläscher, J., Hampton, A. N., & O'Doherty, J. P. (2009). Determining a role for ventromedial prefrontal cortex in encoding action-based value signals during reward-related decision making. *Cerebral cortex*, *19*(2), 483–495.

Gutnisky, D. A., Hansen, B. J., Iliescu, B. F., & Dragoi, V. (2009). Attention alters visual plasticity during exposure-based learning. *Current Biology*, *19*(7), 555–560.

Hampton, A. N., Bossaerts, P., & O'doherty, J. P. (2006). The role of the ventromedial prefrontal cortex in abstract state-based inference during decision making in humans. *Journal of Neuroscience*, *26*(32), 8360–8367.

Hauser, T. U., Iannaccone, R., Walitza, S., Brandeis, D., & Brem, S. (2015). Cognitive flexibility in adolescence: Neural and behavioral mechanisms of reward prediction error processing in adaptive decision making during development. *NeuroImage*, *104*, 347–354.

Hazy, T. E., Frank, M. J., & O'reilly, R. C. (2007). Towards an executive without a homunculus: Computational models of the prefrontal cortex/basal ganglia system. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *362*(1485), 1601–1613.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

Hegdé, J. (2008). Time course of visual perception: Coarse-to-fine processing and beyond. *Progress in neurobiology*, *84*(4), 405–439.

Huys, Q. J., Maia, T. V., & Frank, M. J. (2016). Computational psychiatry as a bridge from neuroscience to clinical applications. *Nature neuroscience*, *19*(3), 404–413.

Iwana, B. K., & Uchida, S. (2021). An empirical survey of data augmentation for time series classification with neural networks. *PLOS ONE*, *16*(7), 1–32. https://doi.org/10.1371/journal.pone.0254841

Ji-An, L., Benna, M. K., & Mattar, M. G. (2023). Automatic discovery of cognitive strategies with tiny recurrent neural networks. *bioRxiv*, 2023–04.

Jiang, B., Wu, T.-y., Zheng, C., & Wong, W. H. (2017). Learning summary statistic for approximate bayesian computation via deep neural network. *Statistica Sinica*, 1595–1618.

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.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, *4*, 237–285.

Karlsson, J. (1994). Task decomposition in reinforcement learning. *Proceedings of the AAAI Spring Symposium on Goal-Driven Learning, Stanford, CA*.

Katahira, K. (2018). The statistical structures of reinforcement learning with asymmetric value updates. *Journal of Mathematical Psychology*, *87*, 31–45.

Koechlin, E., Ody, C., & Kouneiher, F. (2003). The architecture of cognitive control in the human prefrontal cortex. *Science*, *302*(5648), 1181–1185.

Koechlin, E., & Summerfield, C. (2007). An information theoretical approach to prefrontal executive function. *Trends in cognitive sciences*, *11*(6), 229–235.

Konidaris, G. D., & Barto, A. G. (2007). Building portable options: Skill transfer in reinforcement learning. *Ijcai*, *7*, 895–900.

Lashley, K. S., et al. (1951). *The problem of serial order in behavior* (Vol. 21). Bobbs-Merrill Oxford.

Lavin, A., Krakauer, D., Zenil, H., Gottschlich, J., Mattson, T., Brehmer, J., Anandkumar, A., Choudry, S., Rocki, K., Baydin, A. G., et al. (2021). Simulation intelligence: Towards a new generation of scientific methods. *arXiv preprint arXiv:2112.03235*.

Lawrence, A. D., Sahakian, B., Rogers, R., Hodges, J., & Robbins, T. (1999). Discrimination, reversal, and shift learning in huntington's disease: Mechanisms of impaired response selection. *Neuropsychologia*, *37*(12), 1359–1374.

Le, N., Rathour, V. S., Yamazaki, K., Luu, K., & Savvides, M. (2022). Deep reinforcement learning in computer vision: A comprehensive survey. *Artificial Intelligence Review*, 1–87.

Lee, M. D. (2011). How cognitive modeling can benefit from hierarchical bayesian models. *Journal of Mathematical Psychology*, *55*(1), 1–7.

Lee, M. D., & Webb, M. R. (2005). Modeling individual differences in cognition. *Psychonomic Bulletin & Review*, *12*(4), 605–621.

Lenzi, A., Bessac, J., Rudi, J., & Stein, M. L. (2023). Neural networks for parameter estimation in intractable models. *Computational Statistics & Data Analysis*, *185*, 107762.

Li, J.-J., Shi, C., Li, L., & Collins, A. (2023). Dynamic noise estimation: A generalized method for modeling noise in sequential decision-making behavior. *bioRxiv*, 2023–06.

Li, J.-J., Shi, C., Li, L., & Collins, A. G. (2023). A generalized method for dynamic noise inference in modeling sequential decision-making. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *45*(45).

Liang, S., Li, Y., & Srikant, R. (2017). Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.

Lintusaari, J., Gutmann, M. U., Dutta, R., Kaski, S., & Corander, J. (2017). Fundamentals and recent developments in approximate bayesian computation. *Systematic biology*, *66*(1), e66–e82.

Lueckmann, J.-M., Goncalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., & Macke, J. H. (2017). Flexible statistical inference for mechanistic models of neural dynamics. *Advances in neural information processing systems*, *30*.

Luk, C.-H., & Wallis, J. D. (2013). Choice coding in frontal cortex during stimulus-guided or action-guided decision-making. *Journal of Neuroscience*, *33*(5), 1864–1871.

Maia, T. V., & Frank, M. J. (2011). From reinforcement learning models to psychiatric and neurological disorders. *Nature neuroscience*, *14*(2), 154–162.

Master, S. L., Eckstein, M. K., Gotlieb, N., Dahl, R., Wilbrecht, L., & Collins, A. G. (2020). Disentangling the systems contributing to changes in learning during adolescence. *Developmental cognitive neuroscience*, *41*, 100732.

McDougle, S. D., Ballard, I. C., Baribault, B., Bishop, S. J., & Collins, A. G. (2022). Executive function assigns value to novel goal-congruent outcomes. *Cerebral Cortex*, *32*(1), 231–247.

McDougle, S. D., Ivry, R. B., & Taylor, J. A. (2016). Taking aim at the cognitive side of learning in sensorimotor adaptation tasks. *Trends in cognitive sciences*, *20*(7), 535–544.

McGovern, A., & Barto, A. G. (2001). Automatic discovery of subgoals in reinforcement learning using diverse density.

Middleton, F. A., & Strick, P. L. (2000). Basal ganglia and cerebellar loops: Motor and cognitive circuits. *Brain research reviews*, *31*(2-3), 236–250.

Miller, E. K., & Cohen, J. D. (2001). An integrative theory of prefrontal cortex function. *Annual review of neuroscience*, *24*(1), 167–202.

Miller, G. A., Eugene, G., & Pribram, K. H. (2017). Plans and the structure of behaviour. In *Systems research for behavioral science* (pp. 369–382). Routledge.

Miller, K., Eckstein, M., Botvinick, M., & Kurth-Nelson, Z. (2024). Cognitive model discovery via disentangled rnns. *Advances in Neural Information Processing Systems*, *36*.

Minka, T. P. (2013). Expectation propagation for approximate bayesian inference. *arXiv preprint arXiv:1301.2294*.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Montague, P. R., Dolan, R. J., Friston, K. J., & Dayan, P. (2012). Computational psychiatry. *Trends in cognitive sciences*, *16*(1), 72–80.

Moosavi-Dezfooli, S.-M., & Alhussein Fawzi, O. F. (2017). Pascal frossard." *Universal adversarial perturbations." 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, *47*(1), 90–100.

Nassar, M. R., & Frank, M. J. (2016). Taming the beast: Extracting generalizable knowledge from computational models of cognition. *Current opinion in behavioral sciences*, *11*, 49–54.

Nguyen, A., Yosinski, J., & Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 427–436.

Niv, Y. (2019). Learning task-state representations. *Nature neuroscience*, *22*(10), 1544–1553.

Niv, Y., Daniel, R., Geana, A., Gershman, S. J., Leong, Y. C., Radulescu, A., & Wilson, R. C. (2015). Reinforcement learning in multidimensional environments relies on attention mechanisms. *Journal of Neuroscience*, *35*(21), 8145–8157.

Niv, Y., Edlund, J. A., Dayan, P., & O'Doherty, J. P. (2012). Neural prediction errors reveal a risk-sensitive reinforcement-learning process in the human brain. *Journal of Neuroscience*, *32*(2), 551–562.

Nussenbaum, K., & Hartley, C. A. (2019). Reinforcement learning across development: What insights can we draw from a decade of research? *Developmental cognitive neuroscience*, *40*, 100733.

Nussenbaum, K., Velez, J. A., Washington, B. T., Hamling, H. E., & Hartley, C. A. (2022). Flexibility in valenced reinforcement learning computations across development. *Child development*, *93*(5), 1601–1615.

O'Doherty, J. P., Dayan, P., Friston, K., Critchley, H., & Dolan, R. J. (2003). Temporal difference models and reward-related learning in the human brain. *Neuron*, *38*(2), 329–337.

O'Reilly, R. C., & Frank, M. J. (2006). Making working memory work: A computational model of learning in the prefrontal cortex and basal ganglia. *Neural computation*, *18*(2), 283–328.

Packard, M. G., & Knowlton, B. J. (2002). Learning and memory functions of the basal ganglia. *Annual review of neuroscience*, *25*(1), 563–593.

Palestro, J. J., Sederberg, P. B., Osth, A. F., Van Zandt, T., & Turner, B. M. (2018). *Likelihood-free methods for cognitive science*. Springer.

Palminteri, S., Kilford, E. J., Coricelli, G., & Blakemore, S.-J. (2016). The computational development of reinforcement learning during adolescence. *PLoS computational biology*, *12*(6), e1004953.

Palminteri, S., Wyart, V., & Koechlin, E. (2017). The importance of falsification in computational cognitive modeling. *Trends in cognitive sciences*, *21*(6), 425–433.

Perfors, A., Tenenbaum, J. B., Griffiths, T. L., & Xu, F. (2011). A tutorial introduction to bayesian models of cognitive development. *Cognition*, *120*(3), 302–321.

Peterson, D. A., Elliott, C., Song, D. D., Makeig, S., Sejnowski, T. J., & Poizner, H. (2009). Probabilistic reversal learning is impaired in parkinson's disease. *Neuroscience*, *163*(4), 1092–1101.

Piray, P., Dezfouli, A., Heskes, T., Frank, M. J., & Daw, N. D. (2019). Hierarchical bayesian inference for concurrent model fitting and comparison for group studies. *PLoS computational biology*, *15*(6), e1007043.

Poldrack, R. A., Clark, J., Pare-Blagoev, E., Shohamy, D., Creso Moyano, J., Myers, C., & Gluck, M. A. (2001). Interactive memory systems in the human brain. *Nature*, *414*(6863), 546–550.

Qiang, W., & Zhongli, Z. (2011). Reinforcement learning model, algorithms and its application. *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, 1143–1146.

Radev, S. T., D'Alessandro, M., Mertens, U. K., Voss, A., Köthe, U., & Bürkner, P.-C. (2021). Amortized bayesian model comparison with evidential deep learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., & Köthe, U. (2020). Bayesflow: Learning complex stochastic models with invertible neural networks. *IEEE transactions on neural networks and learning systems*, *33*(4), 1452–1466.

Radev, S. T., Voss, A., Wieschen, E. M., & Bürkner, P.-C. (2020). Amortized bayesian inference for models of cognition. *arXiv preprint arXiv:2005.03899*.

Radulescu, A., Daniel, R., & Niv, Y. (2016). The effects of aging on the interaction between reinforcement learning and attention. *Psychology and aging*, *31*(7), 747.

Radulescu, A., Niv, Y., & Ballard, I. (2019). Holistic reinforcement learning: The role of structure and attention. *Trends in cognitive sciences*, *23*(4), 278–292.

Rescorla, R. A., & Solomon, R. L. (1967). Two-process learning theory: Relationships between pavlovian conditioning and instrumental learning. *Psychological review*, *74*(3), 151.

Ribas-Fernandes, J. J., Solway, A., Diuk, C., McGuire, J. T., Barto, A. G., Niv, Y., & Botvinick, M. M. (2011). A neural signature of hierarchical reinforcement learning. *Neuron*, *71*(2), 370–379.

Ribas-Fernandes, J. J., Shahnazian, D., Holroyd, C. B., & Botvinick, M. M. (2019). Subgoal- and goal-related reward prediction errors in medial prefrontal cortex. *Journal of cognitive neuroscience*, *31*(1), 8–23.

Rigoux, L., Stephan, K. E., Friston, K. J., & Daunizeau, J. (2014). Bayesian model selection for group studies—revisited. *Neuroimage*, *84*, 971–985.

Rmus, M., He, M., Baribault, B., Walsh, E. G., Festa, E. K., Collins, A. G., & Nassar, M. R. (2023). Age-related differences in prefrontal glutamate are associated with increased working memory decay that gives the appearance of learning deficits. *Elife*, *12*, e85243.

Rmus, M., McDougle, S. D., & Collins, A. G. (2021). The role of executive function in shaping reinforcement learning. *Current Opinion in Behavioral Sciences*, *38*, 66–73.

Rosa-Alcázar, Á., Olivares-Olivares, P. J., Martínez-Esparza, I. C., Parada-Navas, J. L., Rosa-Alcázar, A. I., & Olivares-Rodríguez, J. (2020). Cognitive flexibility and response inhibition in patients with obsessive-compulsive disorder and generalized anxiety disorder. *International Journal of Clinical and Health Psychology*, *20*(1), 20–28.

Rothenhoefer, K. M., Costa, V. D., Bartolo, R., Vicario-Feliciano, R., Murray, E. A., & Averbeck, B. B. (2017). Effects of ventral striatum lesions on stimulus-based versus action-based reinforcement learning. *Journal of Neuroscience*, *37*(29), 6902–6914.

Särkkä, S., & Svensson, L. (2023). *Bayesian filtering and smoothing* (Vol. 17). Cambridge university press.

Sasaki, Y., Nanez, J. E., & Watanabe, T. (2010). Advances in visual perceptual learning and plasticity. *Nature Reviews Neuroscience*, *11*(1), 53–60.

Schmidhuber, J. (1991). A possibility for implementing curiosity and boredom in model-building neural controllers.

Schmitt, M., Bürkner, P.-C., Köthe, U., & Radev, S. T. (2021). Detecting model misspecification in amortized bayesian inference with neural networks. *arXiv preprint arXiv:2112.08866*.

Schultz, W., Dayan, P., & Montague, P. R. (1997). A neural substrate of prediction and reward. *Science*, *275*(5306), 1593–1599.

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 461–464.

Shahar, N., Moran, R., Hauser, T. U., Kievit, R. A., McNamee, D., Moutoussis, M., Consortium, N., & Dolan, R. J. (2019). Credit assignment to state-independent task representations and its relationship with model-based decision making. *Proceedings of the National Academy of Sciences*, *116*(32), 15871–15876.

Sheynin, J., Moustafa, A. A., Beck, K. D., Servatius, R. J., & Myers, C. E. (2015). Testing the role of reward and punishment sensitivity in avoidance behavior: A computational modeling approach. *Behavioural Brain Research*, *283*, 121–138.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, *6*(1), 1–48.

Shultz, T. R. (2003). *Computational developmental psychology*. Mit Press.

Şimşek, Ö., & Barto, A. (2008). Skill characterization based on betweenness. *Advances in neural information processing systems*, *21*.

Singh, S., Lewis, R. L., Barto, A. G., & Sorg, J. (2010). Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, *2*(2), 70–82.

Sisson, S. A., Fan, Y., & Beaumont, M. (2018). *Handbook of approximate bayesian computation*. CRC Press.

Sokratous, K., Fitch, A. K., & Kvam, P. D. (2023). How to ask twenty questions and win: Machine learning tools for assessing preferences from small samples of willingness-to-pay prices. *Journal of choice modelling*, *48*, 100418.

Solway, A., Diuk, C., Córdova, N., Yee, D., Barto, A. G., Niv, Y., & Botvinick, M. M. (2014). Optimal behavioral hierarchy. *PLoS computational biology*, *10*(8), e1003779.

Stolle, M., & Precup, D. (2002). Learning options in reinforcement learning. *Abstraction, Reformulation, and Approximation: 5th International Symposium, SARA 2002 Kananaskis, Alberta, Canada August 2–4, 2002 Proceedings 5*, 212–223.

Sunnåker, M., Busetto, A. G., Numminen, E., Corander, J., Foll, M., & Dessimoz, C. (2013). Approximate bayesian computation. *PLoS computational biology*, *9*(1), e1002803.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine learning*, *3*, 9–44.

Sutton, R. S., & Barto, A. G. (1990). Time-derivative models of pavlovian reinforcement.

Sutton, R. S., & Barto, A. G. (1999). Reinforcement learning: An introduction. *Robotica*, *17*(2), 229–235.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S., Precup, D., & Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, *112*(1-2), 181–211.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Tai, L.-H., Lee, A. M., Benavidez, N., Bonci, A., & Wilbrecht, L. (2012). Transient stimulation of distinct subpopulations of striatal neurons mimics changes in action value. *Nature neuroscience*, *15*(9), 1281–1289.

Thompson, B., Van Opheusden, B., Sumers, T., & Griffiths, T. (2022). Complex cognitive algorithms preserved by selective social learning in experimental populations. *Science*, *376*(6588), 95–98.

Todd, M., Niv, Y., & Cohen, J. D. (2008). Learning to use working memory in partially observable environments through dopaminergic reinforcement. *Advances in neural information processing systems*, *21*.

Tomov, M. S., Schulz, E., & Gershman, S. J. (2021). Multi-task reinforcement learning in humans. *Nature Human Behaviour*, *5*(6), 764–773.

Turner, B. M., Forstmann, B. U., Wagenmakers, E.-J., Brown, S. D., Sederberg, P. B., & Steyvers, M. (2013). A bayesian framework for simultaneously modeling neural and behavioral data. *NeuroImage*, *72*, 193–206.

Turner, B. M., & Sederberg, P. B. (2014). A generalized, likelihood-free method for posterior estimation. *Psychonomic bulletin & review*, *21*, 227–250.

van Opheusden, B., Acerbi, L., & Ma, W. J. (2020). Unbiased and efficient log-likelihood estimation with inverse binomial sampling. *PLoS computational biology*, *16*(12), e1008483.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., & Kavukcuoglu, K. (2017). Feudal networks for hierarchical reinforcement learning. *International conference on machine learning*, 3540–3549.

Vikbladh, O. M., Meager, M. R., King, J., Blackmon, K., Devinsky, O., Shohamy, D., Burgess, N., & Daw, N. D. (2019). Hippocampal contributions to model-based planning and spatial memory. *Neuron*, *102*(3), 683–693.

Wagenmakers, E.-J., & Farrell, S. (2004). Aic model selection using akaike weights. *Psychonomic bulletin & review*, *11*, 192–196.

Wagner, A. R., & Rescorla, R. A. (1972). Inhibition in pavlovian conditioning: Application of a theory. *Inhibition and learning*, 301–336.

Weber, I., Zorowitz, S., Niv, Y., & Bennett, D. (2022). The effects of induced positive and negative affect on pavlovian-instrumental interactions. *Cognition and Emotion*, *36*(7), 1343–1360.

Wei, Y., & Jiang, Z. (2022). Estimating parameters of structural models using neural networks. *USC Marshall School of Business Research Paper*.

Wilson, R. C., & Collins, A. G. (2019). Ten simple rules for the computational modeling of behavioral data. *Elife*, *8*, e49547.

Wimmer, G. E., & Shohamy, D. (2012). Preference by association: How memory mechanisms in the hippocampus bias decisions. *Science*, *338*(6104), 270–273.

Xia, L., & Collins, A. G. (2021). Temporal and state abstractions for efficient learning, transfer, and composition in humans. *Psychological review*, *128*(4), 643.

Yoo, A. H., & Collins, A. G. (2022). How working memory and reinforcement learning are intertwined: A cognitive, neural, and computational perspective. *Journal of cognitive neuroscience*, *34*(4), 551–568.

Zhao, F., Zeng, Y., Wang, G., Bai, J., & Xu, B. (2018). A brain-inspired decision making model based on top-down biasing of prefrontal cortex to basal ganglia and its application in autonomous uav explorations. *Cognitive Computation*, *10*, 296–306.

Zou, A. R., Muñoz Lopez, D. E., Johnson, S. L., & Collins, A. G. (2022). Impulsivity relates to multi-trial choice strategy in probabilistic reversal learning. *Frontiers in Psychiatry*, *13*, 800290.