

# COGNITIVE MECHANISMS OF COMPLEX PLANNING

by

Ionatan Kuperwajs

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

CENTER FOR NEURAL SCIENCE

NEW YORK UNIVERSITY

JANUARY 2024

---

Wei Ji Ma

© IONATAN KUPERWAJS

ALL RIGHTS RESERVED, 2024

*Most of our distress begins when we imagine how different life should have been. Entrusting your hopes to something as flimsy as your possibilities is the root of all evil. You must accept this you of here and now; you are incapable of becoming anyone else. There's no way you could live a so-called rose-colored campus life to the fullest. I guarantee it, so best dig in your heels.*

Tomihiko Morimi, *The Tatami Galaxy*

# ACKNOWLEDGMENTS

This dissertation represents not only the research that I've done in graduate school, but rather the culmination of my entire academic trajectory. I'd like to first thank my undergraduate advisor at Macalester College, Andrew Beveridge, for instilling an appreciation for mathematics in me in a way that only the best teachers can. I wouldn't be where I am without your encouragement to pursue my collective interests in neuroscience, computer science, and mathematics.

I want to especially thank my advisor, Wei Ji Ma. Whenever people ask me for advice about graduate school, the first thing that comes to mind is to prioritize mentorship above all else. I consider myself extremely fortunate to have had an advisor who is not only a brilliant scientist, but a mentor who genuinely cares about his trainees as people, prioritizes fostering a welcoming lab environment, and is committed to advocating for social good both within and outside of academia. Thank you for being an inspiration to many, and providing an example of how an academic can be so much more than just their research.

I would like to thank everyone else that I have had the privilege of collaborating with over the past few years. In particular, Bas van Opheusden for welcoming me into the line of work that he started, Heiko Schütt for more often than not acting as my second advisor, and Mark Ho for serving as an academic role model. I would be remiss if I didn't mention the wonderful members of the lab that I overlapped with: Hsin-Hung Lee, Dongjae Kim, Aspen Yoo, Jennifer Laura Lee, Peiyuan Zhang, Xiang Li, Daisy Lin, Nastaran Arfaei, Jeroen Olieslagers, and Jordan Lei. Thank you for being the most prominent part of my daily life as a graduate student.

I want to thank the members of my dissertation committee, Christine Constantinople, Catherine Hartley, Todd Gureckis, and Tom Griffiths, for their guidance and comments on the work that is presented here. At New York University, I would like to thank administrative and IT staff in the Neural Science and Psychology departments for their assistance on everything from reimbursements to debugging code on the cluster. I would also like to thank the people involved with the Scientist Action and Advocacy Network for helping enable meaningful work on a variety of advocacy related projects. I also have to mention my cohort, who made moving to a completely new city incredibly easy by providing an immediate circle of close friends that many are not as lucky to have in graduate school.

I want to thank my friends for their continued support throughout my time in graduate school. I apologize in advance as I cannot possibly do everyone justice in the limited space I have. Matthias, you have consistently remained one of my closest friends since high school. Max, Jake, Charlie, and others from the Macalester Men's Soccer Team, you are a testament to how lifelong friendships can be built from sports. Camille, I don't know how I would've managed in graduate school without a roommate to vent to and share that experience with. Ravi, we've somehow started a podcast that has proven to be one of the most enjoyable and fulfilling projects I've ever worked on. Hope, I love the life we're building together and that we push each other to be the best versions of ourselves that we can be.

Finally, I would like to thank my family, many of whom are spread out across the world. Living far apart from my extended family hasn't always been easy, but I am indebted to the experiences we've shared that have shaped who I am. Most important to me are my father, mother, and sister, who are my greatest support system. Mario, perhaps the greatest indication of how close we are is that we still text each other the same message before every Real Madrid game all these years later. Flaminia, you are a shining example of how to be caring, kind, and compassionate. Anael, I could not ask for a better best friend. I hope you know how much I love you.

# ABSTRACT

Planning is a hallmark of human intelligence that involves the mental simulation of futures and their consequences in order to make a decision. Spatial navigation, scheduling, and strategy games are all examples of ecologically relevant planning tasks. Artificial intelligence has fully embraced the challenge of developing powerful algorithms to solve a wide array of problems in large state spaces. Meanwhile, despite the ubiquity of sequential decision-making in naturalistic behavior, the study of the cognitive mechanisms underlying human planning has been primarily limited to relatively simple tasks.

In this dissertation, I will outline a framework for studying the cognitive science of complex planning. This is founded upon a combinatorial game where participants think multiple steps into the future as well as a large-scale data set consisting of millions of games. I will first show how human gameplay can be characterized by a computational cognitive model based on heuristic search, and how that model can provide evidence for increased planning depth with expertise. Then, I will highlight how this task and data set can be leveraged to investigate a diverse set of research directions, from using deep neural networks for guided model improvement to building normative theories of meta-planning to analyzing the relationship between task performance and engagement. Together, my work exemplifies a broad approach to understanding the algorithms that people use to plan and reason in complex environments.

# CONTENTS

<b>Acknowledgments</b>	<b>iv</b>
<b>Abstract</b>	<b>vi</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xii</b>
<b>List of Appendices</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Artificial intelligence . . . . .	2
1.2 Multi-step planning . . . . .	10
1.3 Tasks . . . . .	19
1.4 Dissertation outline . . . . .	26
<b>2 Framework</b>	<b>29</b>
2.1 Task and data . . . . .	30
2.2 Model . . . . .	32
2.3 Results . . . . .	38
2.4 Discussion . . . . .	42

<b>3 Neural networks</b>	<b>44</b>
3.1 Methods . . . . .	47
3.2 Results . . . . .	50
3.3 Discussion . . . . .	58
<b>4 Theory</b>	<b>62</b>
4.1 Model . . . . .	65
4.2 Results . . . . .	69
4.3 Discussion . . . . .	78
<b>5 Learning and motivation</b>	<b>82</b>
5.1 Results . . . . .	85
5.2 Model . . . . .	96
5.3 Discussion . . . . .	98
<b>6 Conclusion</b>	<b>102</b>
6.1 Dissertation summary . . . . .	102
6.2 Limitations and future directions . . . . .	104
6.3 Parting words . . . . .	115
<b>Appendices</b>	<b>118</b>
<b>References</b>	<b>166</b>

# LIST OF FIGURES

1.1	Agent-environment interface in a Markov decision process . . . . .	4
1.2	Heuristic tree search . . . . .	6
1.3	Artificial neural network . . . . .	9
1.4	Tasks for studying human planning . . . . .	11
1.5	Comparing lab-based and game-based tasks . . . . .	23
2.1	4-in-a-row . . . . .	31
2.2	Computational cognitive model for 4-in-a-row . . . . .	34
2.3	Model accounts for human choices and response times . . . . .	37
2.4	Model comparison . . . . .	38
2.5	Effects of expertise on planning . . . . .	40
2.6	Response times in mobile data . . . . .	41
3.1	Neural network architecture . . . . .	48
3.2	Scaling up the neural network . . . . .	49
3.3	Summary statistics for the neural network . . . . .	51
3.4	Iterating over cognitive model extensions . . . . .	53
3.5	Representative residuals for the baseline model . . . . .	54
3.6	Representative residuals for the model extensions . . . . .	57

4.1	Comparing planning and meta-planning . . . . .	64
4.2	Formalizing Bayesian meta-planning . . . . .	66
4.3	Meta-planner simulations . . . . .	70
4.4	Comparing the meta-planner with canonical search algorithms . . . . .	72
4.5	Human prospecton is driven by the action gap . . . . .	74
4.6	Human response times are driven by retrospection and uncertainty . . . . .	76
5.1	Relationship between task performance and total experience . . . . .	87
5.2	Playing strength increases during learning . . . . .	89
5.3	Dropout behavior is driven by recent gameplay . . . . .	91
5.4	Physical time as a factor in gameplay . . . . .	92
5.5	Effect of opponent playing strength on learning and dropout . . . . .	95
5.6	Graphical model of performance and engagement . . . . .	98
6.1	Empirical extensions of 4-in-a-row . . . . .	105
B.1	Neural network training procedure . . . . .	133
B.2	Example high and low accuracy board positions . . . . .	137
B.3	Example high and low entropy board positions . . . . .	138
B.4	Example board positions played by stronger and weaker users . . . . .	139
B.5	Neural network validation . . . . .	141
B.6	Summary statistics for the neural network and the baseline model . . . . .	142
C.1	2-dimensional representations of sampling probability . . . . .	152
C.2	Progression of the action gap . . . . .	153
C.3	Evidence for retrospective response times in 4-in-a-row . . . . .	155
C.4	Evidence for retrospective decision-making in 4-in-a-row . . . . .	156
D.1	Distribution per analysis . . . . .	158

D.2	Validation of Elo ratings as a measure of task performance	162
D.3	Control for edge effects in dropout	163
D.4	Physical time model simulations	164

## LIST OF TABLES

A.1	Number of users per country . . . . .	120
A.2	Per turn scoring . . . . .	121
A.3	End bonus scoring . . . . .	122
B.1	Trained neural networks . . . . .	134
B.2	Tested cognitive models . . . . .	135
D.1	Number of users and games played per analysis . . . . .	159

# LIST OF APPENDICES

<b>A Appendix for Chapter 2</b>	<b>118</b>
A.1 Human-versus-human experiment . . . . .	118
A.2 Large-scale mobile data . . . . .	119
A.3 Detailed model specification . . . . .	124
A.4 Stopping rule . . . . .	128
A.5 Model comparison . . . . .	129
<b>B Appendix for Chapter 3</b>	<b>132</b>
B.1 Neural network training and testing . . . . .	132
B.2 Model extension specification . . . . .	133
B.3 Model fitting . . . . .	136
B.4 Example board positions . . . . .	138
B.5 Neural network validation . . . . .	140
<b>C Appendix for Chapter 4</b>	<b>143</b>
C.1 Detailed model derivation . . . . .	143
C.2 Model simulations . . . . .	152
C.3 Additional behavioral analyses . . . . .	154
<b>D Appendix for Chapter 5</b>	<b>157</b>

D.1	Methods	157
D.2	Control and validation analyses	161
D.3	Model simulations	163

# 1 | INTRODUCTION

Planning is a hallmark of human intelligence. In our everyday lives, we must constantly make decisions by considering the future consequences of our actions. This is made all the more challenging by the fact that we live in a complex world, one in which various events are interconnected throughout time and the outcomes of our sequential choices are difficult to predict.

To be more concrete, imagine a recent college graduate interested in pursuing an academic career. First, they must deliberate between applying to graduate school right away or gathering more research experience. Then, if they eventually decide on the former, they must choose which programs they find appealing and which advisors they might want to work with. This decision can be influenced by numerous factors including scientific interests, mentorship needs, institutional reputation, friends and family, and location, among others. These same considerations can be applied to all positions that they will pursue after completing their doctorate. Even then, they must weigh the likelihood and potential benefits of a tenure-track position against alternate career options to judge whether the payoff is worthwhile. Additionally, every step of this process is not only subject to prospective thinking, but also influenced by prior experiences such as the student's relationship with their advisor or if they spent time in industry completing an internship. In sum, decisions in the real world are often complex, intertwined with other experiences and decisions, and come with substantial uncertainty.

The goal of this chapter is to bring together perspectives, both computational and experimental, from artificial intelligence and cognitive science to support a research program aimed at

understanding the algorithms by which people think ahead in complex environments. This will outline the literature upon which this dissertation is built as well as provide a comprehensive background of the material required to fully contextualize each chapter. The structure of the chapter is as follows: in Section 1.1 I will recap the methods employed in artificial intelligence to create planning algorithms that outperform human abilities, in Section 1.2 I will cover the trajectory of multi-step planning studies in both neuroscience and cognitive science that have shaped our understanding of how people plan, and in Section 1.3 I will focus on tasks, including the psychology of chess as a case study in human planning as well as the movement towards using games and large-scale data sets in tandem to characterize complex behavior.

## 1.1 ARTIFICIAL INTELLIGENCE

### 1.1.1 REINFORCEMENT LEARNING

Reinforcement learning (RL) is arguably the most successful theoretical framework available for explaining goal-directed learning and decision-making [Sutton and Barto 2018]. RL uses the formal framework of a Markov decision process (MDP) to define how an agent interacts with and learns from their environment (Figure 1.1). Beyond the agent and environment, there are four main elements common to RL systems that guide the selection of an action  $a$  in state  $s$ : (1) a policy  $\pi$ , which is a relationship between states in the environment and actions that can be taken in each state, (2) a reward signal  $R$ , which provides feedback contingent upon given events and which the agent must maximize in the long run, (3) a value function  $V$ , which specifies what is good for the agent in the long run, and (4) a model, which mimics the behavior of the environment and allows the agent to make inferences about how the environment will behave. The last component is key to planning, which can be defined as deciding on a course of action by considering possible future situations before they are actually experienced. RL spans the spectrum from model-free methods

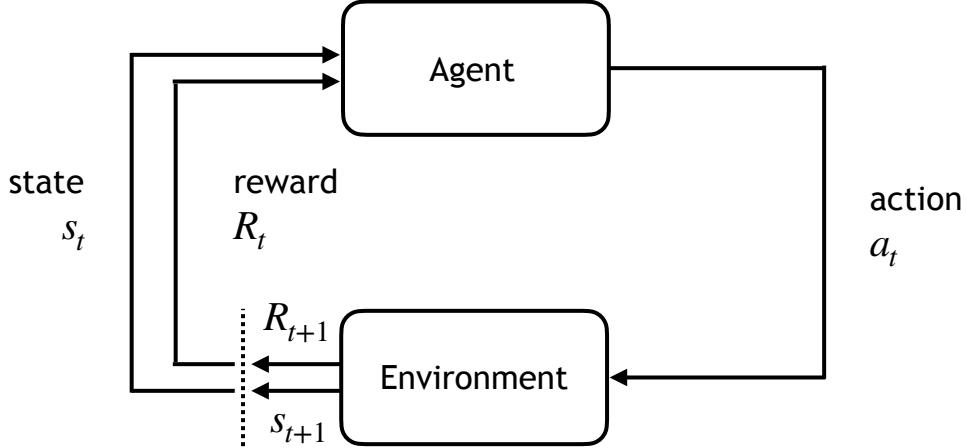
that are explicitly low-level trial-and-error learners to model-based methods that make use of simulations to engage in high-level deliberative planning. Throughout its history, RL has had a particularly strong interaction with psychology and neuroscience, as many of these algorithms were inspired by biological learning systems [Bengio et al. 2009]. In turn, RL has generated computational models that can be tested with behavioral and neural data [Montague et al. 1996].

This raises the question: is all of the RL literature relevant to this dissertation? Undoubtedly, the interplay between model-free and model-based approaches have direct correlates in psychology and neuroscience studies of sequential decision-making [Dickinson 1985; Daw et al. 2005]. Additionally, heuristic search and deep neural networks both intersect with RL and human planning [Silver et al. 2016; Huys et al. 2012]. However, many of the most famous approaches in RL are not directly applicable to the study of the cognitive algorithms underlying planning in complex environments. This is primarily because such algorithms are constructed to derive an optimal value function known as the Bellman equation:

$$V(s) = \max (R(s, a) + \gamma V(s')) \quad (1.1)$$

where  $\gamma$  is a discount factor that determines how much the agent cares about potential rewards in the distant future, denoted by  $V(s')$ , relative to those in the immediate future, denoted by  $R(s, a)$ . Meanwhile, I am interested in large state space settings where approaches to solving the Bellman equation are simply intractable. Modern RL algorithms are now explicitly designed to address this challenge, but optimal value functions are typically not representative of how people cognitively reason in complex tasks.

Nonetheless, there are a number of model-free and model-based RL methods that will hopefully serve to contextualize the experimental and modeling work that I will describe later in this chapter. In model-free RL, an agent does not have access to a transition rule, which predicts the next state after taking an action, and therefore learns purely by trial-and-error. These methods



**Figure 1.1:** A standard agent-environment interface at time  $t$  in a Markov decision process (MDP). Adapted from [Sutton and Barto 2018].

implicitly approximate a value function by averaging over experienced rewards and state transitions. Examples of model-free RL algorithms are dynamic programming (DP), which break down an MDP into smaller problems to perform updates based on current value estimates [Bellman 1966], Monte Carlo methods, which average over sample returns from the environment [Rubinstein and Kroese 2016], and temporal-difference (TD) learning, which combines DP and Monte Carlo methods by bootstrapping from the current estimate of the value function [Sutton 1988; Watkins and Dayan 1992]. Among approaches to model-based RL, one subset of algorithms are concerned with background planning, or using a model to replay or simulate experience offline or as part of a learning rule. A model can be formally defined as a belief state over a transition rule. Planning here is not focused on the current state, but rather on gradually improving a policy or value function that results in better tabular state-action estimates for lookup. Dyna is one relevant example where real experience, passed back and forth between the environment and the policy, affects the policy and value function in much the same way as simulated experience generated by the model of the environment [Sutton 1991]. In this scheme, learning and planning are deeply integrated in the sense that they share almost all of the same machinery. Another extension, called prioritized sweeping, works backwards from goal states to prioritize

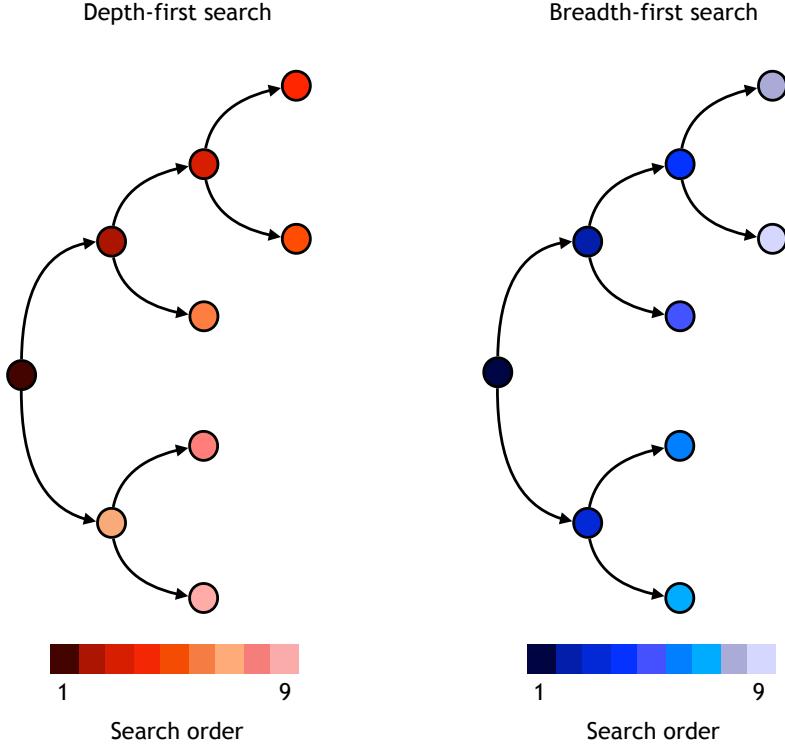
updates according to a measure of their urgency [Moore and Atkeson 1993]. In general, both model-free and model-based RL methods are based on looking ahead to future events, computing a backed-up value, and then using it as an update target for an approximate value function.

### 1.1.2 HEURISTIC SEARCH

The most important technical approach in the context of the relationship between artificial intelligence and this dissertation, and the one I would like to expand on the most, is heuristic search. Heuristic search differs from the previous descriptions of background planning and model-based RL in that it is the classical example of decision-time planning. In decision-time planning, simulated experience is used to select an action for the current state rather than gradually improve a policy or value function. Broadly speaking, planning algorithms in this space can search much deeper ahead than a single step by constructing a decision tree that evaluates many actions leading to different state and reward trajectories (Figure 1.2). The size of this decision tree is exponential in the number of choice points. For example, if an agent has to make a sequence of  $N$  decisions with  $K$  options at each step, then the total number of sequences is  $K^N$ . Heuristic search algorithms deal with this combinatorial explosion by introducing an approximate value function:

$$H(s) = \sum_i w_i f_i(s) \quad (1.2)$$

where  $f_i(s)$  are features of the state and  $w_i$  are feature weights. The weights are learned over experience such that the discrepancy between  $H(s)$  and the true or optimal value of the state  $V(s)$  is minimized [Pearl 1984]. Using this approximation, an agent builds a partial decision tree starting from the current state as the root node, eventually selecting the best action available according to the heuristic function. There are many approaches to constructing this partial tree, one of which is best-first search [Dechter and Pearl 1985]. In best-first search, the agent iteratively selects a sequence of the most promising actions leading to a leaf node, expands the tree by



**Figure 1.2:** A decision tree implementing heuristic search. Two possible sequence of nodes used to construct the tree are numbered, one for depth-first search (red) and another for breadth-first search (blue). The search order of each sequence is shaded from dark to light in increasing order from 1 to 9.

evaluating candidate actions with  $H(s)$ , and backpropagates the information to the root node.

In the history of artificial intelligence, heuristic search has been implemented in various forms, often to play zero-sum, two-player games like tic-tac-toe, chess, and Go. In 1950, Claude Shannon published a groundbreaking paper describing how a machine or computer could be designed to play a reasonable game of chess [Shannon 1950]. His algorithm was based on a minimax procedure, which used an evaluation function of chess positions to select the best move for both players. Later, TD-Gammon was developed as the first program to play backgammon at human master level [Tesauro et al. 1995]. TD-Gammon used TD learning to compute an afterstate value through many games of self-play, using a form of heuristic search to make its moves. Backgammon has a large branching factor, and even selectively searching ahead a few steps drastically improved action selection. Finally, the chess-playing computer that defeated reigning world

champion Gary Kasparov, DeepBlue, made use of alpha-beta pruning to decrease the number of nodes evaluated by its minimax algorithm [Campbell et al. 2002]. This variation on heuristic search stops evaluating a move when at least one possibility has been found that proves the move to be worse than a previously examined move. In this way, the algorithm returns the same move as a standard implementation of minimax would, but prunes away branches that cannot possibly influence the final decision to reduce the size of the search tree. Modern neural networks have further augmented the capabilities of these search methods in combinatorial games [Silver et al. 2016], and there is a rich history investigating the psychology of chess [De Groot 2014; Chase and Simon 1973]. These are important topics to this dissertation, and I cover both in dedicated sections later in this chapter.

One final and highly successful example of decision-time planning is Monte Carlo tree search (MCTS). MCTS is a rollout algorithm, meaning that it estimates action values for a given policy by averaging the returns of many simulated trajectories that start with each possible action and then follow a given policy [Browne et al. 2012]. More specifically, a tree is built by iteratively employing a tree policy, which attempts to balance exploration, or searching in areas of the state space that have not been extensively sampled yet, and exploitation, or searching in areas of the state space that appear to be promising. A simulation is then run from the selected node, with moves made according to some default policy, which in the simplest case is to make uniform random moves. The search tree is then updated with a new child node as well as the result of the simulation that is backpropagated to the root node. The most popular tree policy is an application of the multi-armed bandit (MAB) algorithm UCB1 (Upper Confidence Bound 1) called UCT (Upper Confidence Bound 1 applied to Trees). UCB1 assigns scores to child nodes  $j$  using their expected value  $X_j$ , the number of visits to the parent  $n$  and child  $n_j$ , and an exploration bonus  $C_p$ . UCT then applies this recursively to action selection in decision trees:

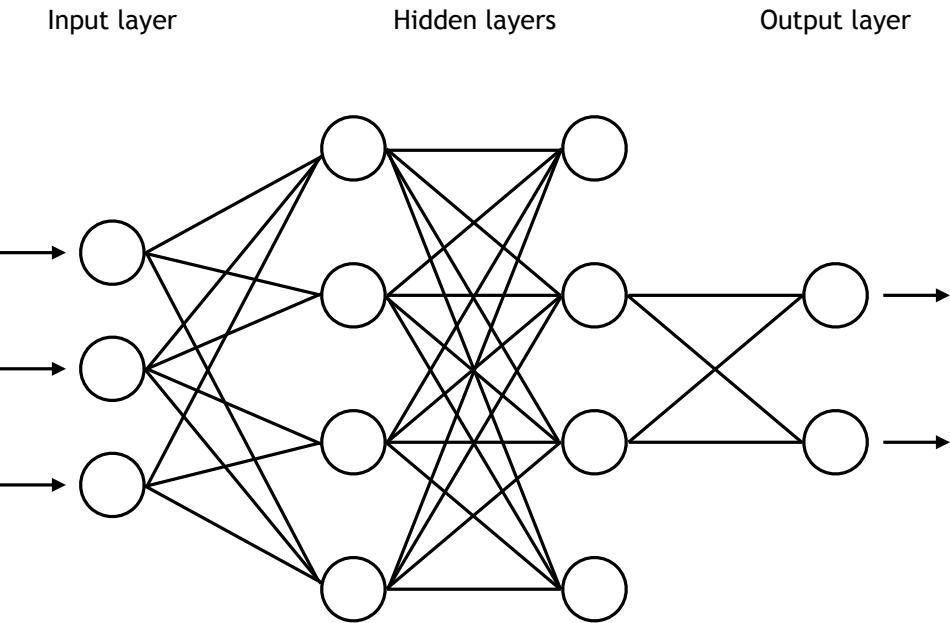
$$UCT = \bar{X}_j + 2C_p \sqrt{\frac{2 \ln n}{n_j}}. \quad (1.3)$$

The first and second terms reflect the balance between exploitation and exploration, and  $C_p$  can be adjusted for the amount of exploration that is desired. MCTS has been deeply influential in artificial intelligence, leading the paradigm shift from chess to Go in the field [Lee et al. 2010] and having broader applications to general game playing [Finnsson and Björnsson 2008]. However, it is important to note that MCTS, and the majority of heuristic search methods in artificial intelligence, are engineered for maximal performance rather than to be human-like in their behavior.

### 1.1.3 ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANNs) are another widely used class of models that typically consist of an architecture, which describes how different units are connected, and a learning algorithm, which is used to learn the appropriate connection weights for the model’s parameters (Figure 1.3). ANNs themselves have a long history that I unfortunately lack the space to delve into, but these networks have the general capacity to represent any computational process [Siegelmann and Sontag 1995; LeCun et al. 2015]. In other words, given enough training time and data, ANNs are nonlinear function approximators that can fit data equally well or better than any other model. ANNs, however, are notoriously difficult to train to generalize across tasks, are high-dimensional and hard to interpret by construction, and the representations and computations underlying their behavior are typically obscure. That being said, they are increasingly becoming a common tool for research, both within the context of artificial intelligence and in application to cognitive science.

Combining prior work on MCTS with ANNs, a team at DeepMind developed AlphaGo in 2016, the first artificial agent to achieve superhuman performance in Go with a series of stunning victories against world champion Lee Sedol [Silver et al. 2016]. The main innovation behind AlphaGo is that it selected moves using a novel version of MCTS that was guided by both a policy and a value function learned by RL with function approximation provided by deep convolutional ANNs. Additionally, instead of starting from random network weights, it started from weights that were pretrained on human experts as a starting point, iterating on previous work that aimed to predict



**Figure 1.3:** A fully-connected feedforward artificial neural network (ANN) with three input units, two output units, and two hidden layers with four units each.

human moves in large Go databases [Stern et al. 2006; Clark and Storkey 2015]. Impressively, AlphaGo was improved on to develop AlphaGo Zero, which used no human data or guidance beyond the basic rules of the game [Silver et al. 2017], and AlphaZero, which does not even incorporate knowledge of Go and subsequently outperforms the world’s best programs not only in Go, but also in chess and shogi [Silver et al. 2018]. These represent state-of-the art advances in artificial intelligence, namely those that utilize self-play reinforcement learning to create computer agents that solve complex planning problems at a level beyond human capabilities.

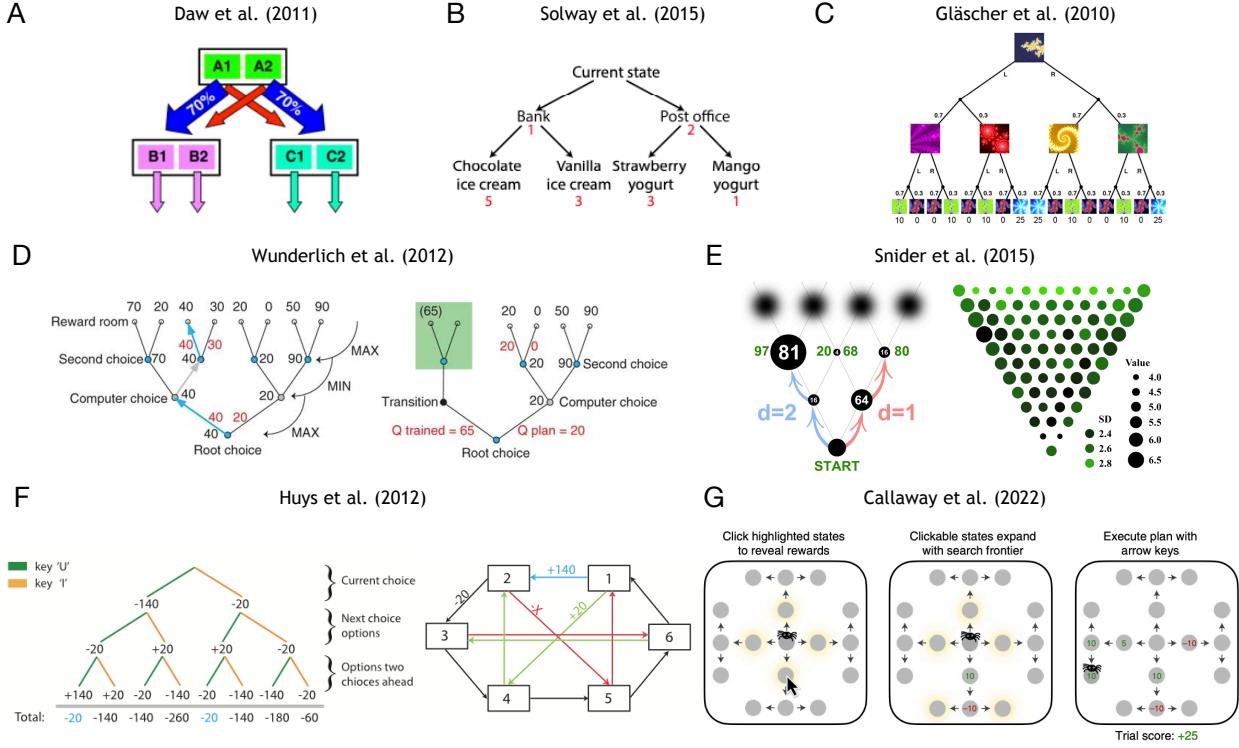
Meanwhile, a related emerging field in cognitive science has leveraged the unbounded expressivity of ANNs to improve our understanding of the algorithms underlying human behavior. Rather than training on the task itself and optimizing for performance, these ANNs are trained to predict human behavior. Then, they are systematically investigated as an upper bound for prediction where any generalizable aspects of the human data are captured by the network and can be extracted for use in interpretable models. In complex tasks where large-scale data has

been collected, relatively simple, feedforward network architectures have been pioneered to discover algorithms underlying human decision-making [Agrawal et al. 2020; Peterson et al. 2021] and categorization [Battleday et al. 2020]. This is done by concurrently examining the deviations from the neural network’s predictions and a cognitive model that is meant to be improved. In simpler RL contexts, recurrent neural networks have been the primary architectural choice [Dezfouli et al. 2019; Eckstein et al. 2023; Miller et al. 2023; Ji-An et al. 2023]. These come with the tradeoff that they are more difficult to train and analyze, but have the added benefit of being useful for more smaller, more common data sets. In conclusion, ANNs are being applied broadly across domains, two specific instances of which are to develop better algorithms within artificial intelligence itself and as a tool for creating better computational models of human behavior.

## 1.2 MULTI-STEP PLANNING

### 1.2.1 EXPERIMENTAL FINDINGS

Recent years have seen a surge in experimental work aimed at understanding the neural mechanisms of multi-step planning in the brain [Miller and Venditto 2021; Mattar and Lengyel 2022]. Here I review the literature that is most important for transitioning to studies that investigate how humans engage in forward thinking, but a wide range of neural structures that contribute to associative learning, food foraging, and spatial navigation have been implicated in planning as well. A cornerstone of the empirical study of sequential decision-making was Tolman’s finding that rats navigating in a maze use data gathered from free exploration to build a mental map of their environment that they can subsequently use for efficient, goal-directed planning [Tolman 1948]. Following this, the activity of hippocampal place cells was decoded to determine what part of space is being representing on a moment-to-moment basis. When a rat pauses at a choice point in a maze, these representations sweep forward along the possible paths that the



**Figure 1.4:** Decision trees in tasks used to study human planning, roughly sorted by state space complexity. Adapted from [van Opheusden and Ma 2019].

animal can take [Johnson and Redish 2007]. Furthermore, the spatial trajectories represented by these sweeps closely correspond to the rat's subsequent navigational behavior [Pfeiffer and Foster 2013]. Hippocampal neural activity has also been associated with both background and decision-time planning [Pezzulo et al. 2019], and more recently it was determined that rats can efficiently navigate or direct objects to arbitrary goal locations solely by activating and sustaining appropriate hippocampal representations of remote places [Lai et al. 2023].

Perhaps the most influential study of sequential decision-making employs the two-step task, originally in animals [Daw et al. 2005] and extended to humans [Daw et al. 2011b]. In this task, participants make a sequence of two binary choices between states (Figure 1.4A). In the first decision stage, the participant chooses between stimuli A1 and A2. On 70% of trials, choosing A1 leads to the set of B states and choosing A2 leads to the set of C states. However, on 30% of trials, the opposite transitions occur. In the second stage, participants make another choice between

two stimuli, which yields a monetary reward with some probability. The reward probabilities fluctuate slowly, so participants have to constantly adapt the values they associate with the stimuli and adjust their decisions accordingly. Notably, this is the simplest task in which model-free and model-based RL make different behavioral predictions. When a reward is received, a model-based agent has the capacity to take into account whether it arrived there through a common or rare transition, whereas a model-free learner does not. In the seminal work in which the task was introduced, it was found that people use a mixture of model-based and model-free learning. Additionally, this task has been utilized to demonstrate that the relative usage of each system depends on the reliability of their respective predictions [Lee et al. 2014] or on an online cost-benefit analysis [Kool et al. 2016], and that people's arbitration between these systems changes under cognitive load [Otto et al. 2013] or when they receive a dopamine precursor [Wunderlich et al. 2012b]. More recently, it has been argued that sophisticated model-free learning can masquerade as model-based learning in the two-step task [Akam et al. 2015] and that more detailed task instructions lead participants to make primarily model-based choices that have little model-free influence [Feher da Silva and Hare 2020]. Moreover, further evidence for the neural substrates of planning in animals has been found in different adaptations of the task [Miller et al. 2017; Groman et al. 2018; Akam et al. 2021].

Arbitration between model-based and model-free RL, as contextualized by the two-step task, provides a direct link to artificial intelligence. To choose an action in a given state, a model-based system mentally simulates the consequences of possible actions multiple steps into the future, whereas the model-free system considers the outcome of actions taken in the same or similar states in past experience. These dual systems have been discussed under various names and implementations, with the most standard mapping to RL being habitual and goal-directed control of learned behavioral patterns [Dickinson 1985; Dolan and Dayan 2013]. The model-based system is slow and computationally expensive, but can determine high-value actions from any state, including ones that the agent has never previously encountered. On the other hand, the model-

free system is fast but needs previous experience to inform its policy. While this section has thus far focused on model-based RL, there is a long history of findings about model-free RL in animals and humans. These results range from the Rescorla-Wagner model of the circumstances under which Pavlovian conditioning occurs [Rescorla 1972] to the reward prediction error hypothesis of dopamine neuron activity delivering error signals between old and a new estimates of expected future reward to target areas throughout the brain [Schultz et al. 1997]. One question of particular interest in the field is how people combine information from these systems. For example, this decision may be based on uncertainty estimates provided by both systems [Daw et al. 2005] or the historical accuracy of their predictions [Kool et al. 2017]. A related problem is how these systems can benefit from each other’s computations, for which a candidate framework is amortization [Dasgupta et al. 2018], in which the agent re-uses simulated experience from the model-based system as additional training data for the model-free system. While arbitration between dual systems is not a focus of this dissertation, it is closely related to a topic that is, namely deciding whether prospective planning is worthwhile in terms of time and computational resources.

### 1.2.2 HUMAN PLANNING IN A COMPUTATIONAL FRAMEWORK

Beyond the two-step task, the study of human planning is rich and has employed an entire suite of experimental tasks. A core challenge is that researchers can only gather data about people’s decisions, but planning is an internal, cognitive process that is inherently not observable. Thus, one method for inferring planning algorithms is to fit a computational model to human behavior and evaluate how closely the model predicts people’s decisions. While studies in this regime primarily rely on planning tasks of limited complexity compared to those used in artificial intelligence, the more traditional approach to science comes with the advantage that behavior can be precisely explained with process-level models.

This computational framework has led to substantial progress in understanding the algorithmic mechanisms that people use to plan, specifically in the context of navigating and constructing

decision trees. One idea that emerged in a series of papers was that planning can be conceptualized as probabilistic inference [Botvinick and Toussaint 2012; Solway and Botvinick 2012]. In a real-world decision task where participants had to choose between items that they had previously ranked by desirability (Figure 1.4B), behavior was captured by noisy evidence integration, which treats each path through the decision tree as a competitor in a bounded accumulation process [Solway and Botvinick 2015]. Another study dissociated neural correlates of reward prediction errors and state prediction errors by extending the two-step task (Figure 1.4C) to include more second-level states and introducing third-level states that deterministically lead to reward [Gläscher et al. 2010]. In a two-player variant of the two-step task, the transitions from the first to second level states were made by an adversarial computer agent (Figure 1.4D). This allowed for the identification of neural correlates of the values of individual branching steps in a minimax decision tree [Wunderlich et al. 2012a]. Human planning has also been studied in a fast-paced, dynamic environment where participants watched a triangular lattice of disks of different sizes scroll down a touchscreen and traced the most rewarding path (Figure 1.4E). Participants received a reward proportional to the size of all disks on that trajectory, and human behavior was found to be consistent with planning several steps into the future [Snider et al. 2015]. In another goal-directed decision-making task, participants were asked to make a sequence of multiple two-alternative choices by which they traversed a graph (Figure 1.4F). Each transition incurred a reward which could be either positive or negative, and the task was designed such that the optimal policy requires taking large negative rewards to obtain positive future rewards. This revealed that people plan along multiple branches in a decision tree, but eliminate unpromising branches by pruning [Huys et al. 2012] and decompose the task into a hierarchy of subtasks [Huys et al. 2015].

To reiterate the aforementioned point that experimental studies in neuroscience as well as the computational literature in cognitive science have primarily concerned themselves with limited complexity tasks to study planning, the state space of these tasks range from a lower bound of 3 in the two-step task and [Solway and Botvinick 2015] to an upper bound of 128 in [Huys et al. 2012].

By contrast, chess has approximately  $10^{47}$  states [Chinchalkar 1996] and Go has approximately  $2.1 \cdot 10^{170}$  states [Tromp 2016]. Simpler tasks inherently impose a ceiling for the depth of planning, and therefore characterizing people's planning strategies in complex environments that resemble more naturalistic behavior remains an open problem. That being said, these results introduce important concepts that a successful model of sequential decision-making would need to take into account such as relying on uncertainty, integrating information from various sources, and pruning certain courses of action.

### 1.2.3 OPTIMALITY AND RECENT ADVANCES

An outstanding problem in the study of human planning that has not been directly addressed by the studies in the previous section is how people are able to solve novel problems when their actions have long-reaching consequences given the huge number of actions and outcomes that could be considered. Indeed, exhaustive planning, or traversing a full decision tree, is often intractable since the size of the tree grows exponentially with the number of steps that one looks ahead. One way to frame the results covered thus far is in terms of heuristics, with researchers proposing and testing different possible methods that people could be using to reduce the costs associated with planning. This dependence of human planners on heuristics has been cited as far back as Newell and Simon in one of the earliest attempts to replicate human-like intelligence in a computer [Newell and Simon 1956; Newell et al. 1959]. In many domains, progress has been made by analyzing optimal solutions to a problem that a cognitive system is meant to solve [Marr 2010; Anderson 2013]. Normative approaches to modeling human planning are fairly sparse, although there have been a number of recent attempts. One is the plan-until-habit scheme, which executes forward planning up to some depth and then exploits heuristic values from a habitual system as proxies for consequences that may arise further into the future [Sezener et al. 2019]. This framework is designed to optimally trade off speed and accuracy under the assumption that deeper planning leads to more accurate evaluations, but at the cost of slower decision-making.

The critical value to be computed when deciding if to expand the decision tree in a certain trajectory is the value of uncertainty reduction (VUR). VUR computation examines whether a new piece of information, possibly provided by a further expansion of the tree along a trajectory, could change the agent’s decision about what action to take and how much extra value is expected to be gained by that policy improvement. Formally, VUR is defined as the difference between the amount of future rewards that are expected to be gained with the expansion of strategy  $A_i$  and without the expansion of  $A_i$ :

$$VUR(A_i|F) = \mathbb{E}_{\mu_i^*} [\max \left( \mu_i^*, \max_{A \in F - A_i} \mathbb{E}[V(A)] \right)] - \max_{A \in F} E[V(A)]. \quad (1.4)$$

Here,  $F - A_i$  is the frontier set  $F$  of all strategies excluding  $A_i$ , and  $\mu_i^*$  is the expected mean of strategy  $A_i$  after the potential expansion. However, this variable is computed before expansion using a discount factor as well as the mean and the variance of the model-free value distribution for the last action made using  $A_i$ . This algorithm is close in form to arbitration between model-based and model-free RL, and can reproduce several behavioral patterns in grid-world environments and the [Huys et al. 2012] task, namely the effect of time pressure on the depth of planning, the effect of reward magnitudes on the direction of planning, and the gradual shift from goal-directed to habitual behavior during training.

A second normative approach to modeling human planning has been to cast the problem in terms of resource rationality. Resource-rational analysis also strives towards optimality by deriving models of human behavior that take into account which cognitive operations are available to people, how long they take, and how costly they are [Russell and Wefald 1991; Griffiths et al. 2015; Lieder and Griffiths 2017]. Applied to planning, the problem is formalized as a sequential decision problem in which an agent executes a sequence of cognitive operations to construct a decision tree [Callaway et al. 2022b]. More specifically, optimal solutions can be derived as a function of the conceptual and technical tools for metalevel MDPs, which, in contrast to standard

MDPs, dictate the interaction between an agent and its internal, computational environment. This creates a metalevel question similar to the one tackled in [Sezener et al. 2019]: which states should the agent consider to achieve the best tradeoff between the costs and benefits of planning? The key observation underlying the resource rational model in this context is that the basic and metalevel problems are both sequential decision problems. That is, they require the agent to make a sequence of choices in which the outcome of each choice depends on which choices were made previously. While the basic problem is defined by states of the world, physical actions, and external rewards, the metalevel problem is defined by decision trees and the mental operations that build them. Thinking about planning from this perspective enables both the identification of the optimal strategy as the one that maximizes the expected utility of executing a plan minus the cost of each cognitive operation required to make that plan as well as a flexible framework for testing heuristic planning strategies. In addition, a process-tracing paradigm that externalizes the cognitive operations underlying planning as mouse clicks was developed (Figure 1.4G) and used to investigate the predictions of different models. This is an extension of the Mouse-lab paradigm [Payne 1976], where participants navigate a directed graph in which each node is associated with a reward that is only revealed when the participant selects the corresponding node. The sequence in which participants choose to reveal rewards provides insight in the cognitive process by which participants plan their actions. In a series of four experiments, participants were found to use planning strategies that are largely consistent with optimal planning strategies, using previously proposed heuristics when they are adaptive.

Finally, I would like to briefly mention a few recent advances in the study of human planning that I believe are relevant to this dissertation and combine distinct elements of the work on planning I have reviewed thus far. In terms of the neural basis of planning, it has been demonstrated that people use prospective information to guide current choices, and located the representation of prospective information in cingulate and prefrontal cortices [Kolling et al. 2018]. Another study developed a normative theory, based on Dyna, to predict not just whether but which memories

should be accessed to enable the most rewarding future decisions [Mattar and Daw 2018]. This theory conceptualizes planning as learning about values from remembered experiences, generalizing work on tradeoffs between model-based and model-free controllers with a gain term that prioritizes states behind the agent when an unexpected outcome is encountered and a need term that prioritizes states ahead of the agent that are imminently relevant. Together, this unifies various functions of hippocampal replay including planning, learning, and consolidation. One aspect of human planning which I have neglected to mention until now is that most studies assume that task representations are complete and fixed. However, efficient and flexible planning might also need to control these representations in order to quickly simplify and more easily reason about problems. One model formalizes how specific task decomposition strategies towards subgoals reflect resource rational tradeoffs [Correa et al. 2023], while another characterizes how an ideal, cognitively limited decision-maker forms value-guided construals that balance the complexity of a representation and its use for planning and acting [Ho et al. 2022a]. There have also been extensions in the realm of heuristic and optimal algorithms for human planning. One line of work has tackled the breadth-depth dilemma in large decision trees, accounting for people's policies that trade off between evaluating many options and gaining more information about a smaller number of options [Moreno-Bote et al. 2020; Mastrogiosseppe and Moreno-Bote 2022]. Another model makes use of previously exercised action sequences to make planning faster and more accurate by focusing expansion of the search tree on paths that have been frequently used in the past [Éltető and Dayan 2023]. This effectively reduces deep planning problems to shallow ones via multi-step jumps in the tree and can be embedded into an MCTS planner. Overall, the computational approach to studying multi-step planning in humans continues to be an active area of research, with contributions guided by disparate task environments, the interplay between heuristics and optimality, and results in neuroscience and RL.

## 1.3 TASKS

### 1.3.1 CHESS

Chess presents an intriguing case study in multi-step planning applied to human cognition. As I've previously outlined, computers have already far surpassed human abilities in chess, primarily by relying on increased computational resources to evaluate orders of magnitude more moves. Top players can arrive at decisions that are nearly as good as those selected by computers despite having vastly greater resource limitations. Therefore, the cognitive science of chess has historically been of a topic of much interest. In fact, chess was referred to by Chase and Simon as the "Drosophila of psychology," or a standard task environment around which knowledge and understanding can accumulate much like model organisms in biology. Seminal work in this field dates back to Adriaan de Groot, who in 1946 proposed that strong chess players make moves by constructing a decision tree through an iterative deepening algorithm [De Groot 2014]. Experimentally, De Groot studied gifted chess players, conducting experiments in which he presented players with pre-configured board positions and asked them to freely narrate their thought process while selecting a move, finding no differences between stronger and weaker players. In another experiment, de Groot instructed players to memorize and reconstruct given chess positions, this time finding that stronger players were able to place more pieces correctly. In 1973, Chase and Simon repeated the reconstruction experiment, but added a control condition in which players were provided with scrambled and often illegal chess positions [Chase and Simon 1973]. They found that players were better at reconstructing legal positions, suggesting that their representations in memory were somehow aided by commonly occurring board states. As a mechanism, they hypothesized that people represent chess positions with an array of small patterns called chunks, allowing them to compress information and avoid capacity limits.

Since these experiments, a growing body of literature has investigated the nature of expertise

in planning by studying how expert chess players differ from less skilled counterparts. Over the years, the explanation for the superior performance of experts in chess has been hotly contested. One line of thought is that this difference is primarily due to better pattern recognition. To support this hypothesis, another replication of the reconstruction experiment analyzed which specific features of a chess position players remember incorrectly [Linhares et al. 2012]. Other studies found no difference in search between experts and novices [Gobet and Simon 1998] or used eye movements and visual search tasks to further validate that experts possess chess-specific improvements in performance [Bilalić et al. 2010]. Conversely, some experiments have shown that deeper search is a key factor for improved play in chess. In a set of papers, players were asked to play chess under time pressure [Holding 1992] or while counting backwards [Holding 1989a] in order to tax their working memory. Both manipulations were designed to selectively impair search while leaving pattern recognition abilities intact, and affected experts more than novices. In another experiment, players evaluated chess positions taken from Grandmaster games, after which they were shown the next few moves in the game and asked to re-evaluate [Holding 1989b]. Here, weaker players were more likely than stronger players to change their evaluation after witnessing the Grandmaster’s moves. There have also been studies that directly investigated differences in search between experts and novices, finding that they do exist [Saariluoma 1992; Campitelli and Gobet 2004]. At least one intermediate proposal has been suggested, which is that improved search may be responsible for the development from novice to expert, but the step from expert to Grandmaster level relies on pattern recognition [Ericsson and Smith 1991].

However, developing computational cognitive models that accurately predict the moves of individual chess players has proven to be difficult [Gobet and Jansen 1994; Gobet 1997]. Instead, studies implicitly assume models of the cognitive processes by which chess players arrive at their decisions and often rely on clever experimental manipulations or verbal reports to answer questions about human reasoning during play [Holding 1989a; Charness 1989]. While there is still no process-level theory of human planning in chess, recent progress in understanding people’s

strategies during gameplay has been made due to technological advancements in computer hardware as well as the availability of online chess data. One computational approach in this domain tested the hypothesis that people intelligently select the situations in which computational resources are spent [Russek et al. 2022]. Specifically, players seemed to spend more time thinking in board positions where planning was more beneficial, and this effect was greater in stronger players. This article combined the Stockfish chess engine (<https://stockfishchess.org>) to estimate the benefit of applying planning computations for each board position occurring in 12.5 million games from the Lichess database (<https://lichess.org>) as follows:

$$\Delta U_c = U_c(m_c) - U_c(m_{-c}). \quad (1.5)$$

The benefit of computation is the increase in board position advantage  $\Delta U_c$ , where players can make the maximum utility move  $m_{-c}$  with no planning or perform a planning computation which leads to a more accurate utility function  $m_c$  and then select the new maximum utility move. Meanwhile, another study utilized the same chess database to develop an algorithm for calculating the riskiness of each move in a chess game [Holdaway and Vul 2021]. This showed that players exhibit state-dependent risk preferences, change their risk-taking strategy depending on their opponent, and that these effect differs in experts when compared to novices. A third study investigated the kinds of decisions where people are likely to make errors in chess [Anderson et al. 2017]. After conducting this analysis at a large scale, features describing the inherent difficulty of a move in a game turned out to be significantly more powerful than features based on skill or time in terms of predicting errors in human play. This collection of results suggest that, despite the state space complexity of chess, it is possible to use quantitative methods to investigate human cognition in a paradigm for complex planning.

### 1.3.2 GAMES AS COGNITIVE TASKS

Complementary to studies that use chess, there has been a noticeable trend in cognitive science to break the traditional practice of experimental paradigms that are designed to isolate a single variable while carefully controlling for sources of variation. Much like the standard tasks used to study human planning, this reductionist approach permits the field at large to conduct precise statistical and computational modeling. Despite this, it also restricts the set of answerable questions that can be posed by researchers, and many groups have expanded their repertoire of tasks to include games (Figure 1.5). Games can be loosely defined as "facilitators that structure player behavior and whose main purpose is enjoyment" [Aarseth 2014; Allen et al. 2023] and have been proposed as candidate environments to study the mind since the days of Newell and Simon [Newell and Simon 1956; Gobet et al. 2004]. The benefits of using games in psychology are twofold: (1) they can clarify whether results or models derived from constrained laboratory tasks generalize, and (2) they can open new research directions for understanding human cognition. In contrast to conventional paradigms, which tend to involve abstract, arbitrary rules along with explicit instructions to guide behavior and require monetary rewards to incentivize participation, tasks that are game-like are importantly both intuitive and enjoyable. In the remainder of this section, I explain these two components in more detail.

Games are designed to produce behavior approaching the complexity of the real world by being intuitive to players. In other words, games reflect the assumptions that people have a priori, or their inductive biases, which constrain and guide a learner to prefer one hypothesis over another. One successful game for studying inductive biases in a more complex environment required participants to select one of three tool objects to interact with in a scene as well as the precise location to place it [Allen et al. 2020]. In this setting, people represented actions relationally to compress the space of actions to consider, and these actions were learned via limited amounts of trial-and-error experience. Researchers have also used existing games such as the

	Planning	Memory	Exploration	Problem solving	Multi-agent
Lab-based tasks					
Game-based tasks					

**Figure 1.5:** Comparing lab-based and game-based tasks developed to investigate different cognitive functions. **Top row:** the two-step task [Daw et al. 2011b], an *n*-back memory task [Kirchner 1958], a multi-armed bandit task [Gershman 2019], the Towers of London task [Shallice 1982], and a matrix-form social coordination task [Costa-Gomes et al. 2001]. **Bottom row:** a programmatically generated video game [Tsividis et al. 2021], Sea Hero Quest [Coutrot et al. 2018], Little Alchemy [Brändle et al. 2022], the Virtual Tools game [Allen et al. 2020], and Overcooked [Wang et al. 2020]. Adapted from [Allen et al. 2023].

Atari video game suite, where each state is an image shown on the screen. Here, people critically made use of the existence of objects to play the game and build relational theories about how these objects should behave [Dubey et al. 2018; Tsividis et al. 2021]. This extends to multiplayer games, which necessitate shared inductive biases and have been used to study social behaviors that rely on cultural transmission, collective search, and other large-scale social phenomena [Carroll et al. 2019; Wang et al. 2020; Kumar et al. 2021].

Games also provide a mechanism for engagement by making participation itself naturally rewarding. Previous findings on curiosity and boredom argue that people prefer to participate in challenging tasks that provide new information while avoiding excessively simple or difficult tasks [Schmidhuber 2010; Geana et al. 2016; Ten et al. 2020]. However, designing experiments to study intrinsic motivation, meaning that they can track all the potential kinds of exploration people can do in the real world and avoid explicit rewards, is nontrivial. This makes it hard to

investigate questions such as why people explore new systems [Brändle et al. 2021] or persist in the presence of repeated failure [Leonard et al. 2017]. One related study adapted a game in which players have to combine elements to create new elements in order to understand how people navigate environments where there are no explicit goals [Brändle et al. 2022]. By using data from an original mobile game that people elected to play, empowerment was identified as a key factor for continued play in a truly intrinsically motivating setting. Another experiment investigated player enjoyment in games with a specific goal [Pedersen et al. 2021]. Seemingly, compensating players via classical experimental platforms led to players exploiting subtle flaws in the game’s mechanics to rapidly complete the task in an unintended way. However, when the concept of citizen science was introduced, where participation from the general public was framed as collectively working towards a common goal, players behaved much more conscientiously. In short, games are engineered to be both intuitive and enjoyable, and thus make viable candidate tasks in cognitive science.

### 1.3.3 LARGE-SCALE DATA SETS

The renewed interest in games has coincided with another trend in cognitive science, which is analyzing massive data sets collected through online experiments. The purpose of this methodological shift is to obtain rich data in participants’ real-world environments, and when combined with game-based studies can lead to invaluable accounts of naturalistic behavior. Indeed, these two approaches are almost inseparable, as complex tasks simply require more data to be properly studied from a computational perspective. Newly available pipelines have undoubtedly encouraged researchers to begin implementing large-scale studies, from the ease of developing games with platforms like Unity to rapid data collection platforms such as Prolific or Amazon Mechanical Turk to database storage systems like Structured Query Language (SQL). Further, partnering with developers and other companies to make custom games or to gamify classical experimental paradigms has been a fruitful approach that makes use of existing platforms and userbases.

In turn, this data can be used to interrogate a wide array of cognitive mechanisms, and virtually every study mentioned in the previous section was facilitated by a large-scale data set. One of the first such online experiments required players to guide a neuron from connection to connection by quickly clicking on potential targets [Stafford and Dewar 2014]. This allowed analyses to be conducted for the full time course of learning, revealing that while practice improved all players' performance, it did not affect all players equally. Another challenge that these kind of data sets pose relates back to motivation, since participants now have complete autonomy over when and for how long to engage in the given task. That is to say, when individuals drop out for reasons that are related to their current or future performance, their learning functions are directly biased. The existing literature on human learning often ignores this effect, with a notable exception extrapolating group learning policies for age-related differences in dropout in a large-scale learning study [Steyvers and Benjamin 2019]. In general, massive data sets allow researchers to test theories over many more data points and participant characteristics than was previously possible with laboratory-based samples. For example, a virtual navigation task collected data across 4 million participants in 195 countries gave insight into why some nations have better navigators [Coutrot et al. 2018], how environment can shape spatial skills [Coutrot et al. 2018], and personalized diagnostics for individuals at genetic risk of Alzheimer's disease [Coughlan et al. 2019]. Otherwise, various experiments have made use of such data to uncover naturalistic behavior in visual search [Mitroff et al. 2015], exploration in real-world choices [Schulz et al. 2019], and individual differences associated with learning [Steyvers et al. 2019; Steyvers and Schafer 2020]. The argument for studying more naturalistic tasks, which is closely related to the collection of large-scale data sets, has also been made in the context of RL [Wise et al. 2023].

Finally, machine learning methods require large amounts of training data. As an example, the approaches that leverage ANNs to construct computational models are primarily enabled by recent large-scale sources of human behavioral data. This is similarly true for many cognitive studies that have a neural network component, such as the recent wave of large language model

(LLM) experiments. LLMs are transformers that recognize, translate, predict, or generate text or other content, and have been used to study human cognition globally as well as in specific domains such as problem solving [Yao et al. 2023] and game playing [Akata et al. 2023]. Moving forward, the shift towards large-scale data will continue to be crucial in developing naturalistic theories of human behavior and for bridging the gap with artificial intelligence.

## 1.4 DISSERTATION OUTLINE

In this dissertation, I will explore human decision-making in a task where the primary difficulty is the requirement to plan ahead. This task is a two-player combinatorial game which has similar properties to chess, but lies at an intermediate level of complexity where behavior is rich yet still amenable to cognitive modeling. In Chapter 2, I will introduce the task, which we call 4-in-a-row, as well as a large-scale mobile data set consisting of over 10 million games from over 1.2 million unique users. I will then present a heuristic search model that combines a value function and best-first search to predict people’s choices in the game, as well as results showing that the model can reveal insights on the nature of expertise while planning. These are meant to serve as foundational components upon which the rest of the dissertation was built. The work described in Chapter 2 was led by Bas van Opheusden with contributions from Gianni Galbiati, Zahy Bnaya, and Yunqi Li and published in *Nature* [van Opheusden et al. 2023].

In Chapter 3, I will train deep neural networks on this data set, showing that they accurately predict human moves while capturing meaningful patterns in the data. Then, I will use deviations between the heuristic search model and the best network to identify opportunities for model improvement. Based on this analysis, I will implement three extensions to the model that range from a simple opening bias to specific adjustments regarding endgame planning. This chapter is meant to demonstrate the advantages of model comparison with a high-performance deep neural network as well as the feasibility of scaling cognitive models to massive data sets for

systematically investigating the processes underlying human sequential decision-making. The work described in Chapter 3 was done in collaboration with Heiko H. Schütt and published in *Scientific Reports* [Kuperwajs et al. 2023]. A preliminary version was published as a conference paper in the *Proceedings of the 44th Annual Meeting of the Cognitive Science Society* [Kuperwajs et al. 2022].

In Chapter 4, I will derive a principled framework for meta-planning, or determining which action to plan for. Specifically, the model is an abstracted metacognitive process where evaluating candidate actions via simulation is viewed as gaining noisy measurements of the value of each action. This statistical estimate is then combined with prior experience to decide whether and in which direction to plan. I will highlight how the model produces intuitive simulation results across a range of parameters and acts as a more valuable, informed method for guiding planning when compared to best-first and breadth-first search. Additionally, I will link this normative approach to meta-planning back to 4-in-a-row by showcasing that it produces hypotheses that account for various human response time trends. The main contribution of the model is to treat policy evaluation as an inference problem, where a distributional representation of value is used to compute information gain and direct simulations towards the most promising actions. The work described in Chapter 4 was done in collaboration with Mark K. Ho and will be submitted for publication after this dissertation is complete. A preliminary version was published as a conference paper in the *Proceedings of the 43rd Annual Meeting of the Cognitive Science Society* [Kuperwajs and Ma 2021].

Finally, in Chapter 5, I will pivot away from planning to show that this complex task and data set provide opportunities to ask a wide variety of scientific questions. Namely, I will investigate the relationship between learning and motivation by establishing that there is a correlation between task performance and total experience. Then, I will present a series of analyses which aim to uncover the factors that influence this correlation, such as playing strength, the time interval between games, and opponent difficulty. I will conclude by presenting a dynamic model of the

nature of people's learning over time that replicates these empirical findings. The work described in Chapter 5 will be submitted for publication after this dissertation is complete. A preliminary version was published as a conference paper in the *Proceedings of the 44th Annual Meeting of the Cognitive Science Society* [Kuperwajs and Ma 2022].

## 2 | FRAMEWORK

What algorithms do people use to make decisions with future consequences in complex environments? Planning and sequential decision-making are ubiquitous in many aspects of daily life. A core difficulty of studying human planning from the perspective of a cognitive scientist is developing tasks and models that scale to the full complexity of real-world problems. Laboratory experiments that probe planning often restrict the size of the state space in order to preserve the tractability of behavioral modeling [Daw et al. 2005; Huys et al. 2012; Solway and Botvinick 2015]. Meanwhile, studies across other domains such as expertise in chess [Chase and Simon 1973; De Groot 2014], player modeling [Yannakakis and Togelius 2018], and artificial intelligence [Silver et al. 2016] embrace complexity but compromise on the level of detail in their methods, or are not concerned with modeling planning in a manner that is human-like. As such, the approach outlined in this chapter and maintained throughout this dissertation is to compromise between these two extremes by designing a task that is complex enough that it allows for rich data on the computational principles underlying human sequential-decision making, but at the same time simple enough to allow for precise behavioral modeling.

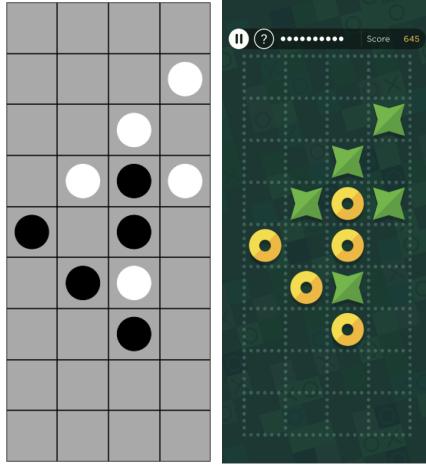
In this chapter, I provide a description of the proposed framework for studying complex planning. Namely, I introduce a task and human behavioral data set as well as a heuristic search model, all of which will be reused throughout later chapters. Then, I summarize the key methods and results from the publication in which these were introduced [van Opheusden et al. 2023], validating the model itself and investigating the nature of expertise in planning.

## 2.1 TASK AND DATA

The perfect task for studying complex planning needs to be difficult enough that strong play requires thinking multiple steps ahead, but tractable for computational modeling. Furthermore, to encourage learning, it should be novel, have simple rules, and be engaging. To satisfy these competing desiderata, we use an instance of an *mnk*-game. An *mnk*-game is adversarial, where two players alternate placing tokens on an  $m$ -by- $n$  board and the objective for both players is to get  $k$  of their own tokens in a row horizontally, vertically, or diagonally. As a reference, tic-tac-toe is a 3,3,3 variant of *mnk*, and we use 4,9,4. The game, which we call 4-in-a-row, is played on a 4-by-9 board with the aim of connecting four tokens (Figure 2.1). Since 4-in-a-row is winner-takes-all and without hidden information, it is solvable [Uiterwijk 2019]. That is to say, there is a weak solution where the first player always wins assuming perfect play from both sides. 4-in-a-row can be played online (<https://weijimalab.github.io>).

With approximately  $1.2 \cdot 10^{16}$  non-terminal states, 4-in-a-row has a state space complexity far beyond common cognitive science tasks [van Opheusden and Ma 2019]. This level of complexity prevents any exhaustive search or brute force algorithms from being successful. Therefore, people as well as artificial agents who play the game need to address the challenge of efficient search, which aims to find promising moves without expending excessive computational resources. Understanding the algorithms that people employ to solve this computational problem provides insight into human cognition, specifically in the domain of sequential decision-making.

We conducted several laboratory experiments and analyses to investigate human behavior in 4-in-a-row. In our first experiment, 40 human participants played games against other human players without any time pressure. This is the sole laboratory experiment that I include in this dissertation, since it was used to validate the heuristic search model, conduct model comparison, generate computer opponents for the large-scale implementation of the task, and compute playing strengths as well as the model’s derived metrics. We also fit behavioral data across a



**Figure 2.1:** An example board position in 4-in-a-row in the laboratory version of the task (left) and the gamified version used on the mobile platform (right). Two players, black and white or yellow circles and green stars, alternate placing pieces on the board, and the first player to connect four pieces in any orientation wins the game.

wide array of other experiments, namely generalization, Turing test, eye tracking, time pressure, learning, and memory and reconstruction. The methods and results for each are available in [van Opheusden et al. 2023].

We also partnered with Peak, a mobile app company, to implement a visually enriched version of 4-in-a-row on their platform (<https://www.peak.net>), which users play at their leisure in their daily environment. We are currently collecting data at a rate of approximately 1.5 million games per month, and throughout this dissertation we used a subset consisting of 82,761,594 moves from 10,874,547 games and 1,234,844 unique users collected between September 2018 and April 2019. In this version of the task, users always move first against an AI agent implementing the model of human planning that we describe in the next section, with parameters adapted from fits on the previously collected human-versus-human games [van Opheusden et al. 2023]. In the context of this chapter, this data was used to investigate the generalizability of the results from the laboratory learning experiment. In all other chapters, this is the primary data set used.

## 2.2 MODEL

The computational cognitive model for 4-in-a-row is adapted from the artificial intelligence literature, in particular heuristic search [Sutton and Barto 2018]. We assume that people’s choices on each move are independent and generated by the same decision-making process with the same parameters. Our model consists of two components: a value function and a tree search algorithm. Furthermore, we include sources of noise to capture variability in human play and human-like mistakes. Here we define the model formally, and we provide further details in Appendix A.3.

### 2.2.1 VALUE FUNCTION

The core component of the model is an evaluation function  $V(s, w)$  which assigns values to board states  $s$  [Sutton and Barto 2018; Bonet and Geffner 2001; Campbell et al. 2002]. The higher this value, the more likely the player is to win from that state. We assume that people use value function approximation [Sutton et al. 1999] such that their value function is a weighted linear sum of features. To provide an example, a common heuristic in chess is to count pieces for both players, with different point values for different pieces (pawns, knights, rooks, and so on). Similarly, our heuristic function counts how often particular features appear on the board (Figure 2.2A). It weighs those counts by feature weights, resulting in a quick-to-compute but approximate value estimate. For this sum, we used the following 5 features: center, connected 2-in-a-row, unconnected 2-in-a-row, 3-in-a-row and 4-in-a-row. The center feature assigns a value to each square corresponding to inverse Euclidean distance from the board center, and sums up the values of all squares occupied by the player’s pieces. The other 4 features count how often the associated pattern occurs on the board horizontally, vertically, or diagonally:

Connected 2-in-a-row: two adjacent pieces which lie on a line of four contiguous squares, with the remaining two squares empty.

Unconnected 2-in-a-row: two non-adjacent pieces which lie on a line of four contiguous squares, with the remaining two squares empty.

3-in-a-row: three pieces which lie on a line of four contiguous squares, with the remaining square empty. This pattern represents an immediate winning threat.

4-in-a-row: four pieces which lie on a line of four contiguous squares. This pattern appears only in board states where a player has already won the game.

We associate weights  $w_i$  to these features, and define the value function as follows:

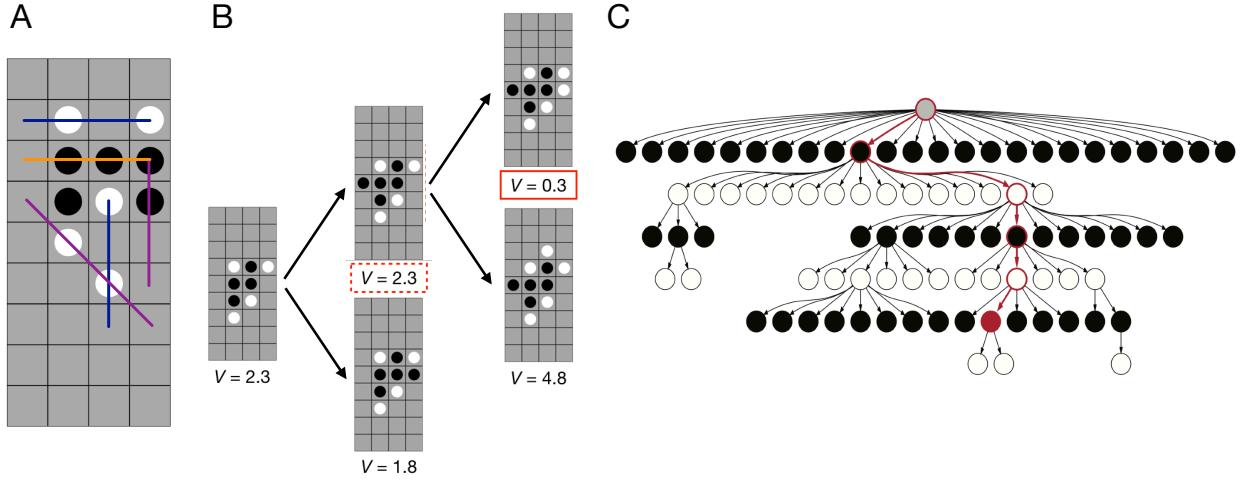
$$V(s, w) = \sum_{i=0}^4 w_i f_i(s, \text{self}) - \sum_{i=0}^4 w_i f_i(s, \text{opponent}). \quad (2.1)$$

In the following, we omit the dependence of  $V(s, w)$  on  $w$  for brevity.

Whenever the model evaluates a state, the weights of features belonging to the active player are multiplied by a scaling constant  $C$ . This captures value differences between active and passive features. For example, a 3-in-a-row feature signals an immediate win on the active player's move but not the opponent's because it can be blocked. We do not scale the center feature.

### 2.2.2 TREE SEARCH

To refine the value estimate, the model explores a decision tree of possible continuations (Figure 2.2B-C). The evaluation function guides the construction of this decision tree with an iterative best-first search algorithm [Dechter and Pearl 1985]. Each iteration, the algorithm chooses a board position to explore, evaluates the positions resulting from each legal move, and prunes all moves with value below that of the best move minus a threshold  $\theta$ . As such, the model expands nodes on the principal variation, or the sequence of best moves for both players given the current decision tree. The model's mechanisms for pruning branches in the decision tree with low heuristic value was inspired by previous studies [Huys et al. 2012, 2015]. This improves the effi-



**Figure 2.2:** Computational cognitive model for 4-in-a-row. **(A)** Features used in the heuristic function of the cognitive model, which are intermediate patterns to winning the game. Features with identical colors are constrained to the same weights, and the heuristic evaluation is a sum over the counts of these features. The model also includes a central tendency feature and a 4-in-a-row feature. **(B)** Illustration of the heuristic search algorithm. In the root position, black is to move. After expanding the root node with two candidate moves for black and evaluating the resulting positions using the heuristic function, the algorithm selects the highest value node ( $V = 2.3$ ) on the second iteration and expands it with two candidate moves for white. The algorithm evaluates the resulting positions, and backpropagates the lowest value ( $V = 0.3$ ), since white is the opponent, meaning that the value in the red solid box replaces the one in the red dashed box and the root node is updated to the highest value among its children ( $V = 1.8$ ). On the next iteration, the algorithm will again expand the child node with the highest value. **(C)** Decision tree built by the model. The red nodes indicate the sequence of highest-value moves for both players. Note that different branches of the tree are evaluated to different depths

ciency of search, but the model may not spot winning sequences. The algorithm has a stopping probability  $\gamma$ , resulting in a geometric distribution over the number of iterations.

### 2.2.3 NOISE

To account for variability in people's choices and enable the model to make human-like mistakes, we added three sources of noise. Before constructing the decision tree, we randomly dropped features at specific locations and orientations to model selective attention, which are omitted during the calculation of  $V(s)$ . We interpret these feature omissions cognitively as lapses of selective attention [Treisman and Gelade 1980]. During tree search, we added Gaussian noise to

$V(s)$  at each node. Finally, we included a lapse rate  $\lambda$ .

#### 2.2.4 MODEL FITTING

The model has 10 parameters: the 5 feature weights, the active-passive scaling constant  $C$ , the pruning threshold  $\theta$ , the stopping probability  $\gamma$ , the feature drop rate  $\delta$ , and the lapse rate  $\lambda$ . We infer these parameters for individual participants with maximum-likelihood estimation. Unfortunately, deriving the log-likelihood analytically requires marginalization of all latent variables (i.e. which features are dropped, the value at each node, and the number of iterations in the search algorithm), which is intractable. Restricting ourselves to only models with analytical likelihoods would limit the types of models that we can consider, particularly in regards to the noise structure. Instead, we estimated the log probability in a given board position with inverse binomial sampling (IBS) [van Opheusden et al. 2020], which compares the data to simulated data generated from the model. IBS is unbiased but its estimates are noisy. Additionally, we cannot calculate gradients of the log-likelihood, so we optimized the log-likelihood function with Bayesian adaptive direct search [Acerbi and Ma 2017]. To reduce overfitting, we compare models using 5-fold cross-validation.

This pipeline is computationally expensive, and fitting one participant’s data for a single model requires approximately 10<sup>14</sup> floating point operations. We perform the model fits on the NYU high-performance cluster (Intel Xeon E5-2690v2 CPUs 3.0 GHz) with a parallel implementation of IBS that uses 20 cores. On our hardware, fitting takes approximately 1 hour per set of parameters for a participant.

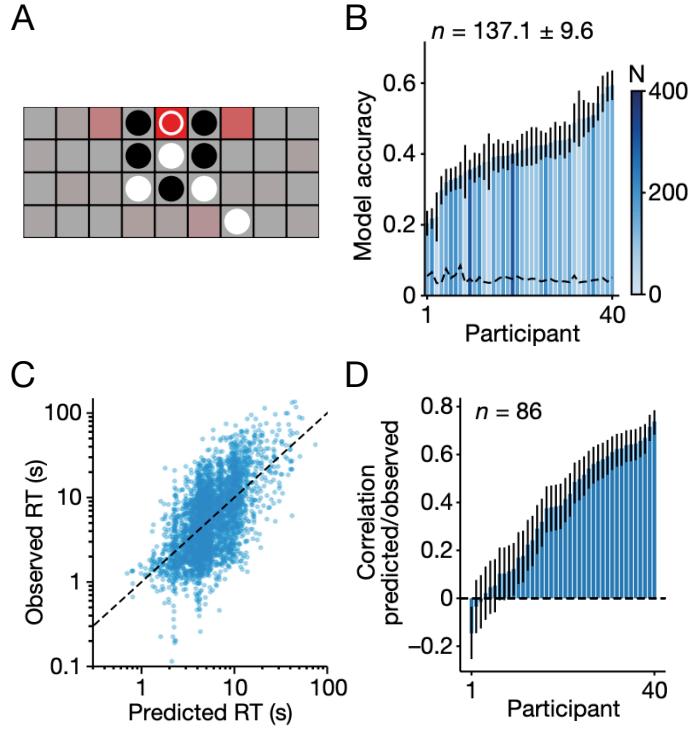
#### 2.2.5 ANALYSIS METHODS

To estimate a player’s playing strength from games against computer opponents, we use Elo ratings [Elo 1978], implemented using the publicly available program Bayeselo [Hunter 2004]. To

measure Elo ratings of all players in all experiments against a common baseline, we run Bayeselo on a database containing all games and a simulated computer-versus-computer tournament, in which each computer plays once against every other computer, including itself. In the computer-versus-computer tournament, we include all agents used across all laboratory experiments as well as the agents used in the mobile app.

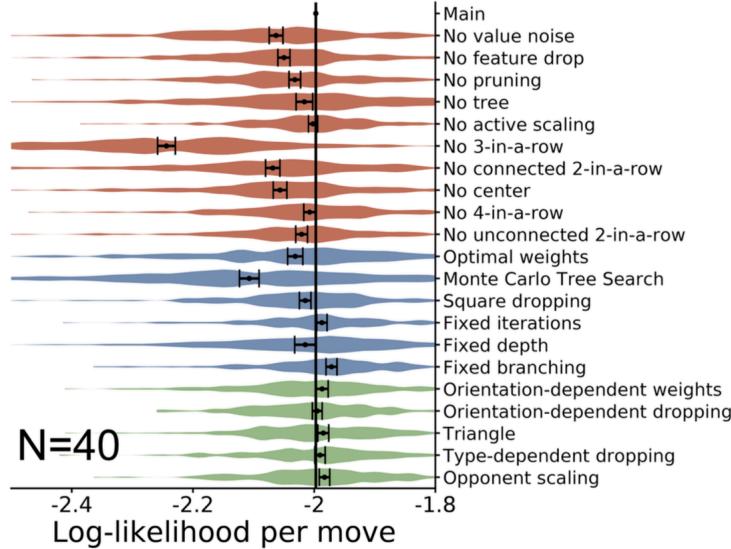
We can use the model to calculate the game-theoretic value  $\tilde{V}(s)$  of a position  $s$ , which is the outcome of a game starting from position  $s$  assuming perfect play from both sides. To compute the game-theoretic value, we execute the best-first search algorithm with default feature weights, no sources of noise, and no pruning. In the limit of infinitely many iterations, the value of the root node in the decision tree of best-first search is guaranteed to converge to the game-theoretic value. In practice we found that 200,000 search iterations was sufficient for almost all positions. For positions in which 200,000 iterations did not yield a determined result, we set the game-theoretical value to  $\tilde{V}(s) = 0$ , which is equivalent to a draw.

To analyze the nature of expertise, we convert the set of 10 parameters from the main model to 3 derived metrics: planning depth, feature drop rate, and heuristic quality. We define planning depth as the length of the principal variation in the model’s decision tree, averaged across simulations of the model with a given a parameter vector in a fixed set of probe positions. Specifically, we used all positions that occurred in the human-versus-human experiment (5,482 positions). As in algorithm 2, the principal variation is the sequence in which both players make the best move according to the values in the decision tree, from the root node to a leaf node. The length of this sequence is equal to the depth of that leaf node, and reflects how far into the future the model plans. We average this depth across 10 simulated moves, and across all probe positions. The feature drop rate is simply the parameter  $\delta$ . To define heuristic quality, we evaluate  $V(s)$  in all the probe positions, and compute the Pearson correlation between  $\tanh(V(s)/20)$  and the game-theoretic value  $\tilde{V}(s)$ . Note that the heuristic quality only depends on the feature weights  $\vec{w}$  and the active scaling constant  $C$ .



**Figure 2.3:** The model accounts for human choices and response times. **(A)** Example board position from a human-versus-human game. The open circle indicates the move that the active player (white) chose. The red shading indicates the probability distribution of that participant’s next move, as predicted by the model with parameters inferred for that participant using 5-fold cross-validation. **(B)** Model accuracy (percentage of correctly predicted moves) for each participant in human-versus-human games, ranked from worst to best predicted. Data are mean  $\pm$  s.e.m.  $n$  denotes the number of trials per participant. The dashed line represents the accuracy of a “chance” model, which assumes that people move onto a randomly selected unoccupied square. **(C)** Predicted and observed response times (RT) across all participants in the human-versus-human data. We exclude any positions with fewer than 6 or more than 30 pieces on the board. **(D)** The correlation between predicted and observed response times for each participant.

Because the probe positions are fixed in the definition of planning depth, it is purely a function of the model parameters. Planning depth depends primarily on the stopping probability (Spearman correlation:  $\rho = -0.87, p < 0.001$ ), and there is a minor dependence on the pruning threshold ( $\rho = -0.21, p < 0.001$ ). These correlations are computed across a range of parameter vectors taken from model fits to human data. The heuristic quality is a more complicated function of the feature weights and active scaling constant. For example, the heuristic quality correlates with  $w_{3\text{-in-a-row}}/w_{\text{connected 2-in-a-row}}$ , but the correlation is relatively weak ( $\rho = 0.55, p < 0.001$ ), and



**Figure 2.4:** Model comparison. We validate our main model specification by comparing to alternatives in three categories: lesions generated by removing model components (red), extensions generated by adding new model components (blue), and modifications generated by replacing a model component with a similar implementation (green). Cross-validated log-likelihood per move, across all participants in the human-versus-human experiment. Error bars indicate mean and s.e.m. of the difference in log-likelihood with the main model.

other feature weights influence the heuristic quality too. In other words, the derived metrics carve up the set of 10 parameters: planning depth primarily depends on pruning threshold and stopping probability, feature drop rate on the feature drop rate, and heuristic quality solely depends on feature weights. Together, the three metrics provide a reduced representation of the model parameters that is more interpretable, more reliably inferred, and sufficient to capture increases in performance.

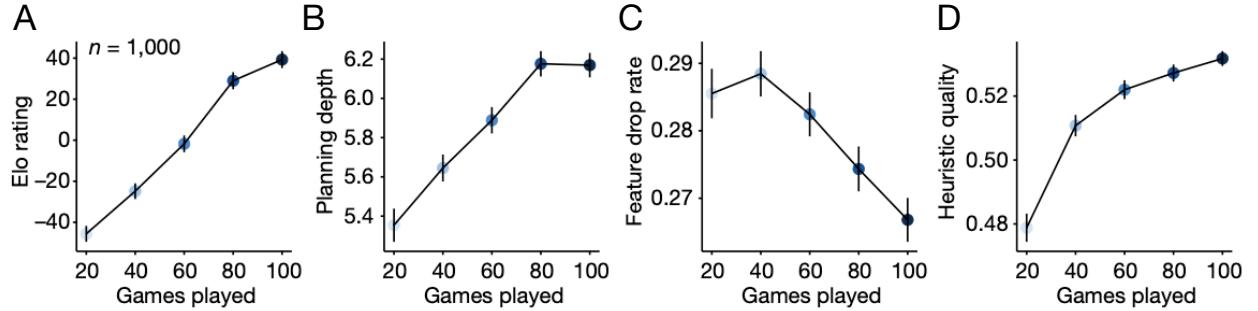
## 2.3 RESULTS

### 2.3.1 MODEL VALIDATION

In the human-versus-human experiment, the model predicts out-of-sample choices with  $40.8 \pm 1.4\%$  accuracy (mean and standard error across participants, two-sample T-test against chance:

$t(39) = 26, p < 0.001$ ). Figure 2.3A shows an example model prediction, and Figure 2.3B the model accuracy for each participant. We also tested the model’s ability to account for process data by analyzing response times. To predict response times, we estimate model parameters from choice data, and we extend the best-first search algorithm with an early stopping rule, which terminates search when the model’s decision is unlikely to change with more iterations (Appendix A.4). We then use the decision tree built by the model on each trial as a predictor for response time. Figure 2.3C shows the predicted and observed response times (in logarithmic space) across all participants in the human-versus-human experiment, and Figure 2.3D the Pearson correlation for each participant ( $\rho = 0.351 \pm 0.029, t(39) = 12, p < 0.001$ ). We can use the logarithm rather than raw response time values because human time perception and production approximately obey Weber’s law [Getty 1975; Grondin 1992; Bizo et al. 2006], but this decision does not affect the results. The ability of the model to predict response times on individual trials suggests that people plan their moves by building decision trees, using an algorithm similar to that in our computational cognitive model.

To validate the specification of our main model, we compare it to 22 alternatives (appendix A.5). We tested three categories of alternative models: lesions generated by removing model components, extensions generated by adding new model components, and modifications generated by replacing a model component with a similar implementation. In Figure 2.4, we show the cross-validated log-likelihood per move of each model averaged across all participants in the human-versus-human experiment. All lesion models fit worse than the main model, showing that all model components are necessary to capture human behavior. In particular, lesioning any feature, tree search, or feature dropping considerably worsens the model. Lesioning the active scaling constant also worsens the model, but its effect is small. Of the modifications, the optimal weights and Monte Carlo tree search models perform poorly. These results show that people’s choices are consistent with a broad class of planning algorithms, namely ones that contain a feature-based evaluation function, tree search, pruning, and a mechanism to capture attentional



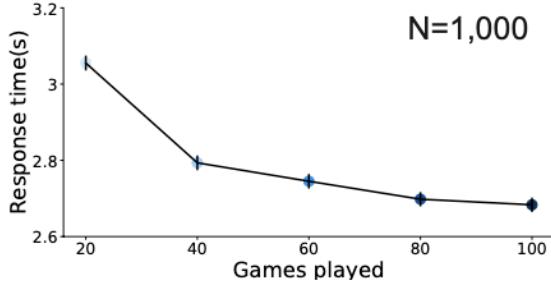
**Figure 2.5:** The effects of expertise on planning in mobile data. **(A)** The average Elo rating of users in the mobile app as a function of number of games played. Data are mean  $\pm$  s.e.m. across participants. **(B)** The average depth to which users plan, as estimated by the behavioral model. **(C)** The same as in (B), but for the feature drop rate. **(D)** The same as in (B), but for heuristic quality.

oversights. We select our main model as a representative of this class that balances parsimony with predictive power.

### 2.3.2 EXPERTISE

The model enables us to investigate how expert players differ from novices. In all of the laboratory experiments, we investigated expertise in participants recruited to perform a psychology experiment, and it was not necessarily clear whether those results would generalize to a more natural context for acquiring expertise. To address this issue, we conducted the same analysis as in the learning experiment using mobile data.

We analyze data from 1,000 randomly selected users who played at least 100 games, which approximately matches the total experience of participants in our learning experiment. For each user, we grouped their experience into 5 blocks of 20 games, and estimated model parameters for each block. Playing strength ( $\beta = 1.13 \pm 0.04, p < 0.001$ , Figure 2.5A) and depth of planning ( $\beta = 0.0108 \pm 0.0010, p < 0.001$ , Figure 2.5B) increase with experience, while feature drop rate decreases ( $\beta = -2.58 \cdot 10^{-4} \pm 4.7 \cdot 10^{-5}, p < 0.001$ , Figure 2.5C). We validate that the increase in users' planning depth is not a result of slower play (Figure 2.6), replicating the results from the laboratory experiment. In this experiment, we also observe a reliable increase in heuristic



**Figure 2.6:** Response time as a function of games played in the mobile data set. Error bars indicate s.e.m. across participants.

quality ( $6.12 \cdot 10^{-4} \pm 4.2 \cdot 10^{-5}$ ,  $p < 0.001$ , Figure 2.5D). However, heuristic quality in the first 20 games of the mobile app data was much lower than that in the first session of the laboratory data ( $0.5301 \pm 0.0098$  versus  $0.4788 \pm 0.0044$ ,  $t(999) = 4.7$ ,  $p < 0.001$ ). Therefore, users seem to have more opportunity to improve their feature weights, whereas heuristic quality in the laboratory data might already start at ceiling.

Our model best matches individual participants' choices with a planning depth of up to 6, which contradicts participants' anecdotal responses as well as previous studies that found lower numbers [Snider et al. 2015; Krusche et al. 2018]. We first note that these planning depth estimates do not imply that people's plan is equally concrete for each of their next 6 moves. Our model contains value noise that is summed along branches of the decision tree. Effectively, the model forms a concrete plan for the first few moves, and later moves are planned more loosely. Additionally, the model contains a sophisticated algorithm for deciding which nodes in the tree to explore, but its decision as to when to terminate this search is random. In practice, this leads the model to often continue search without changing its final decision. By contrast, people's termination rules are more strategic and approach optimality [Callaway et al. 2022b]. In practice, we can implement an early stopping rule that causes the model to estimate lower planning depth. Though the stopping threshold cannot be identified from choice data, the effect of expertise on planning depth is robust across a range of thresholds.

## 2.4 DISCUSSION

In this chapter, we introduced a two-player game of intermediate complexity that provides rich human behavior, but for which computational cognitive modeling is still tractable. We demonstrated that a computational model based on a heuristic value function and forward search algorithm predicts human choices as well as response times. Using this task and model, we showed robust evidence for increased planning and improved attention with expertise in large-scale mobile data. Importantly, our results are consistent with improved pattern recognition in experts [Huang et al. 2023], but highlight the underappreciated role of processing speed.

Would our results on the nature of expertise generalize to more complex games or natural planning tasks? We speculate that in more complex games, expertise will also improve attention and search. For the heuristic quality effect, we note that in the laboratory data, participants already start with approximately correct inductive biases [Dubey et al. 2018] about the relevant features and their relative values, and we observe no increase in heuristic quality with expertise. In the mobile app data, people’s feature weights are initially worse and we do observe an increase. Thus, the model reveals a difference between laboratory and mobile data not obvious from playing strength alone.

Complex games like chess or Go contain many non-obvious features which people can only learn through extensive experience or explicit instruction [Charness et al. 2005]. Therefore, we speculate that in such games, the superior performance of experts also involves domain-specific feature knowledge. We can straightforwardly adapt our model to test this hypothesis, given a procedure to generate sophisticated candidate features. For 4-in-a-row, we discovered a small set of simple features that allow the model to explain people’s choices through manual exploration and model comparison. A promising feature discovery approach for complex games would be to examine internal representations of neural networks trained to either play these games or predict human choices. Finally, our model only applies to deterministic two-player games, and human

behavior in stochastic or multi-player games [Brown and Sandholm 2019; FAIR] might involve additional computational mechanisms.

Our modeling results show how experts differ from novice players, but do not shed light on how those differences are shaped by their specific experience. A promising candidate for modeling the learning process is deep reinforcement learning, specifically algorithms such as AlphaZero [Silver et al. 2018] and SAVE [Hamrick et al. 2019], which combine learning from experience with forward planning at decision time. In future work, we aim to test these theories by analyzing games from all 1.2 million users in the mobile data set. Our work also opens the door to a precise understanding of human planning across development [Ma et al. 2022a] and in patient populations. It also raises the question of how the components of the model are represented neurally. A specific hypothesis is that the value of future states is correlated with the activity of neurons associated with reward-based decision-making, such as those in orbitofrontal cortex [Padoa-Schioppa and Assad 2006]. Additionally, we predict that the time course of neural activity while a player contemplates their move reflects the dynamics of the value of the root node over iterations of the search algorithm. In the following chapters, I will investigate distinct trajectories made possible by the foundational efforts detailed in this chapter.

## 3 | NEURAL NETWORKS

The standard approach to computational modeling in cognitive science involves handcrafting a model and specifying free parameters that are adjusted to produce behaviors consistent with empirical data [Busemeyer and Diederich 2010; Daw et al. 2011a]. Model predictions are then evaluated using the parameter values that achieve the best match to the data. Based on these evaluations, the model is iteratively amended to reduce remaining errors. Whether a specific change is accepted or not is usually based on model comparison techniques, balancing the trade-off between complexity and goodness of fit. This methodology yields interpretable models because all innovations are implemented by the researcher, but it provides no guidance for when to stop searching for candidate models or what changes to try. In this pipeline, there is no way to distinguish whether the unexplained variance represents natural variability in human behavior or could be explained by a crucial change to the model. Even if it can be determined that the model needs improvement, adjustments are usually based on intuition and manual engineering.

One method for addressing these limitations is to fit deep neural networks to behavioral data. Deep neural networks make minimal assumptions about underlying cognitive mechanisms and have sufficient capacity to represent virtually any computational process [Siegelmann and Sontag 1995; LeCun et al. 2015]. Training a network to predict human behavior in a particular task allows the network to detect patterns in the data without requiring human understanding of these patterns. An important step is then validating that the network is indeed accurately capturing human decisions. After validation, the predictions from the network can be compared against a

cognitive model’s predictions. Namely, deviations between the model and the network guide the model improvement process by highlighting situations in which the model requires novel mechanisms to explain human behavior. When there is no clear way of summarizing or pooling data across many trials, this method is more effective than simply investigating the model’s errors, which are often caused by noise that no model can explain. One potential problem with this approach is that neural networks are so flexible that they run the risk of overfitting. Regularization methods are a standard solution to overfitting in scenarios with limited data, while having access to a large data set for training can ameliorate this problem.

Consequently, neural network methods for guided model improvement have established themselves as an emerging field in cognitive science. The approach that we described in the previous paragraph is particularly useful in settings where the task is complex enough to extract additional meaningful information and when large-scale data exists to train relatively simple, feedforward network architectures. This method was pioneered to discover algorithms underlying human decision-making [Agrawal et al. 2020; Peterson et al. 2021] and categorization [Battleday et al. 2020]. A related line of work has started to develop recurrent neural networks for automated model discovery, thus far primarily in reinforcement learning environments [Dezfouli et al. 2019; Eckstein et al. 2023; Miller et al. 2023; Ji-An et al. 2023]. Recurrent neural networks are notoriously more difficult to train and analyze, but in turn can provide results for the simpler tasks and smaller data sets that are more ubiquitous throughout the field. Together, these approaches share the common goal of improving the process for developing cognitive models across a variety of domains.

Recently, a growing body of literature has started to examine the algorithms underlying human sequential decision-making [Daw et al. 2011b; Huys et al. 2012; Snider et al. 2015; Sezener et al. 2019; Kuperwajs and Ma 2021; Callaway et al. 2022b; Ho et al. 2022a]. Planning involves the mental simulation of future actions and their consequences in order to make a decision, but evaluating every possible course of action in real-world environments is simply intractable. Therefore,

a fruitful approach has been to employ tasks with larger state spaces than are typically used in cognitive science coupled with process-level models to investigate how people plan [van Opheusden and Ma 2019]. This combination of complex tasks and models in addition to the fact that planning is an unobservable internal process limits traditional model development frameworks and makes it an ideal domain for testing more powerful methods. One such example is 4-in-a-row, where human decisions have been well-described by a computational cognitive model in both laboratory and online experiments [van Opheusden et al. 2023]. These conditions make 4-in-a-row particularly fitting for an approach to model improvement driven by neural networks: a task with many different states where the key underlying features are hard to identify, a detailed model that is already informative about human planning but can be refined further, and a very large data set for training neural networks.

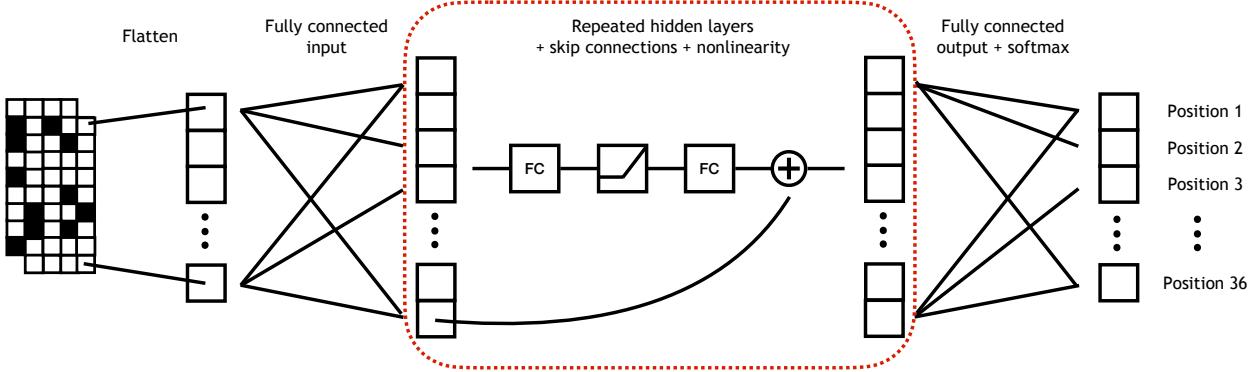
Here, our main contribution is to use deep neural networks to estimate the noise ceiling, or the best fit that can be achieved on the data, relative to a cognitive model and subsequently improve that model. We emphasize the methods used to fit the model and train the neural network such that they can be compared while making use of the entire data set. Then, we show that scaling up the size of the network approaches a satisfactory upper bound on the likelihood of predicting human moves, and that the best network matches human behavior well on a variety of quantitative and behavioral measures. We investigate the residuals between the model and the network, deriving various candidate model improvements from our analysis. Namely, we implement and test three distinct mechanisms that take into account early game biases, complex interactions between model features that result in overlooked moves, and novel features in the heuristic function. Taken together, this chapter highlights how deep neural networks and massive data sets can be leveraged to more systematically refine cognitive models.

## 3.1 METHODS

### 3.1.1 ADAPTING THE EXISTING DATA AND MODEL

Importantly, we can utilize the existing framework for studying complex planning as a starting point for further model development. We partitioned the large-scale 4-in-a-row data into three sets: 90% for training (9,787,093 games), 5% for validation (543,727 games), and 5% for testing (543,727 games). The training and testing sets were used for both the neural networks and the cognitive models, and the validation set was used to monitor learning and experiment with hyperparameters for the neural networks.

One of the main goals of this work is to iteratively improve the interpretable cognitive model of human planning introduced in the previous chapter by comparing its predictions with our best neural network and subsequently testing various mechanisms inspired by this analysis. To avoid confusion, the word model always signifies this cognitive model, while the deep neural network will be referenced as the network. When we implement extensions of this cognitive model later on, we will further delineate by labeling each model. A major technical challenge involves scaling up the fitting procedure for the cognitive model such that it makes use of the large-scale data set and is directly comparable to the neural network. To achieve this, we kept an identical implementation of the heuristic search model but fit parameters for the entire training set. On each model evaluation, we evaluate the log-likelihood on 100,000 trials. We found that this yields an unbiased and sufficiently precise estimate of overall performance. We then optimized this approximate likelihood for 20 different training runs and selected the best-performing parameter set for testing. On the test data set, we ran 100 repetitions per move to estimate a log-likelihood, followed by 200 simulations in each board position to get a probability distribution over move predictions.

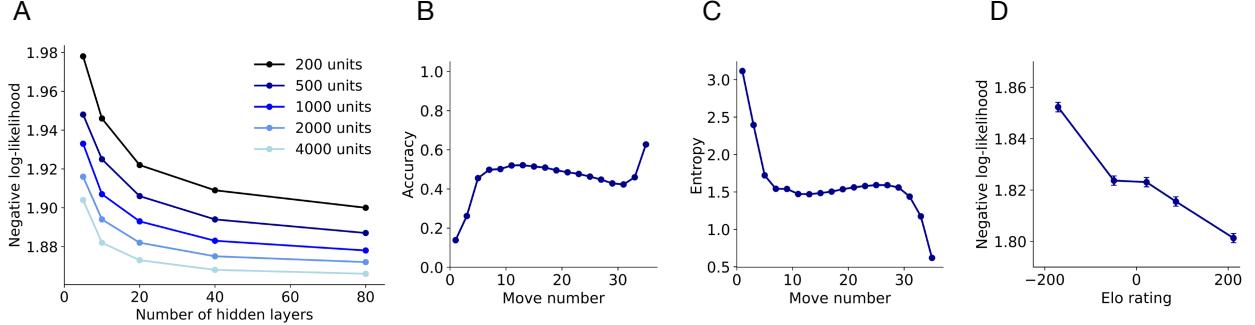


**Figure 3.1:** Neural network architecture. The board is represented as a  $2 \times 4 \times 9$  tensor filled with zeros where there are no pieces and ones where there are pieces. One matrix encodes the user’s pieces, and the second encodes the AI agent’s pieces. The board representation is flattened to a 72-dimensional vector, and then passed into a series of hidden layers. Each hidden layer contains a fully connected layer, a ReLU nonlinearity, another fully connected layer, and then adds the input from skip connections (red dashed box). Finally, the fully connected output layer has 36 units and is passed through a softmax function, which yields the probability that the model assigns to the human player selecting each position of the board. In addition to varying the number of hidden layers in the network, the number of units per fully connected layer is also varied when testing different networks.

### 3.1.2 NEURAL NETWORK TRAINING

To achieve sufficiently high performance on our data set, we constructed a deep neural network architecture that can be systematically scaled up. All of our networks take a tensor representation of the current board state and return a probability distribution for the next move over all board positions. The predictions for different board positions are independent of each other in order to match the cognitive model. We encode each board as two  $4 \times 9$  binary matrices. The first matrix has ones indicating the location of the user’s pieces, while the second  $4 \times 9$  matrix has ones marking where the AI agent’s pieces are located. Unoccupied locations contain a zero in both matrices. Thus, the input to each network is  $2 \times 4 \times 9$ , and the output of the network is a 36-dimensional vector, with each element representing a corresponding index of the board.

The architecture for our networks consists of an input layer that feeds into several hidden layers followed by an output layer (Figure 3.1). The input layer flattens the  $2 \times 4 \times 9$  board into a 72-dimensional vector and projects it to the number of dimensions used by the hidden layers



**Figure 3.2:** Scaling up the neural network achieves a satisfactory upper bound on goodness of fit. **(A)** Negative log-likelihood on the test data set as a function of the number of hidden layers and number of units per hidden layer in each network. **(B)** Accuracy as a function of move number for the best neural network, averaged across the test set. **(C)** Entropy of the best neural network’s output distribution as a function of move number, averaged across the test set. **(D)** Negative log-likelihood on the test data set as a function of playing strength, computed as an Elo rating (binned into quantiles).

with a fully connected layer. Each hidden layer consists of two fully connected layers with a rectified linear function between them and skip connections. These skip connections add the input of the hidden layer to its output without transformation, and aid in avoiding the vanishing gradient problem [He et al. 2016]. The output layer is a fully connected layer that projects from the dimensionality of the hidden layer to 36 units corresponding to the log probabilities for each board position. During training, we scaled the network architecture by varying the number of hidden layers as well as the number of units in each fully connected layer. In Figure B.1A, we show an example loss curve for the largest network that we trained. We observed nearly identical performance on validation and test data that we did not use for training, indicating that overfitting is not an issue for our data set.

To eliminate potential predictions at squares occupied by pieces already on the board, we subtracted a large value from the output at these locations. The final softmax operator always sets the corresponding outputs to exactly 0 and normalizes the probability distribution over all open positions. This also nulls all gradients for the occupied positions such that their values are ignored for gradient backpropagation and learning during training. Prior work used convolutional networks to predict human moves in Go [Sutskever and Nair 2008; Clark and Storkey

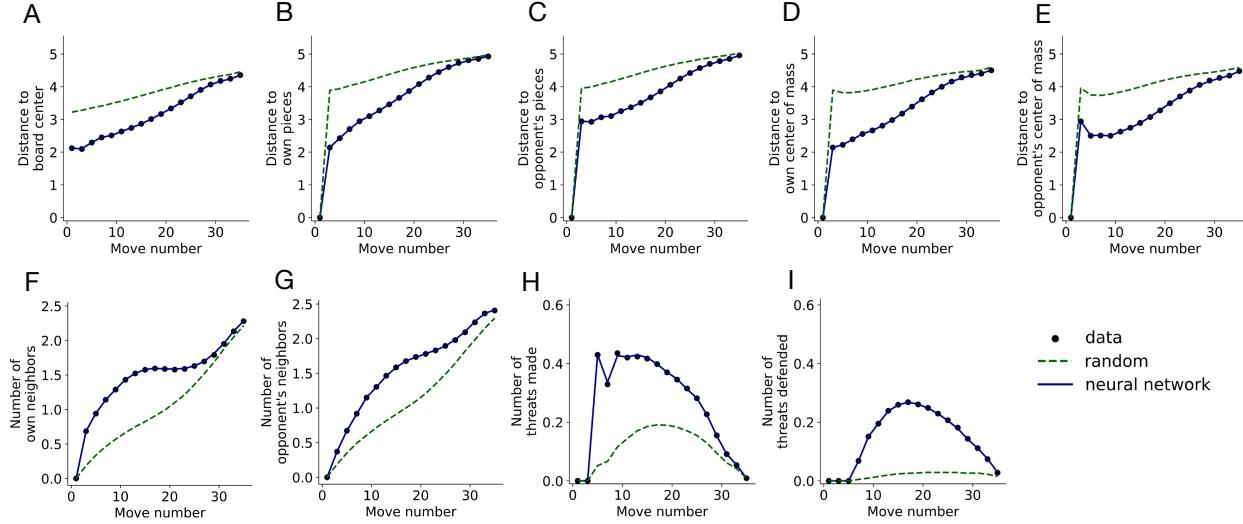
2015], and we initially tested similar architectures in our task. However, we consistently found that the convolutional networks performed worse than the fully connected layers in preliminary training runs. Therefore, we decided to move forward using only fully connected networks.

## 3.2 RESULTS

### 3.2.1 NEURAL NETWORK EVALUATION

In order to predict human behavior, we trained a total of 25 networks that varied along two dimensions: the number of hidden layers and the number of units per layer, spanning a range from 5 to 80 layers and 200 to 4000 units. We continued scaling up the networks until the log-likelihood on the test data reached a plateau, meaning that additional increases in either dimension would not lead to significant increases in performance (Figure 3.2A). The largest network achieved a negative log-likelihood of 1.87 per move and a prediction accuracy of 41.71% on the test data. Additionally, this network’s log-likelihoods per move were highly correlated with the networks that are one step smaller in either direction, further supporting our conclusion that our results would not radically change with larger networks (Figure B.1B-C). Therefore, we continue to analyze the largest network in the remainder of the chapter. A full specification of the networks that we trained and their performance is available in Table B.1.

We then assessed whether the network convincingly captures behavioral patterns. We first considered the accuracy of the network’s predictions (Figure 3.2B) and the entropy of the network’s output distributions (Figure 3.2C), both broken down by move number. Intuitively, positions in the early game are harder to predict because they consist of fairly empty boards where no player can immediately win the game, and therefore result in lower accuracy and higher entropy for the network’s output. Conversely, positions in the middle and late game are much easier to predict as there are fewer alternatives and more pieces to inform decision-making, leading



**Figure 3.3:** Summary statistics as validation that the neural network exhibits human-like behavior. Each statistic is averaged by move number for moves made by users (black circles), the neural network (blue lines), or a random model (green dashed lines).

to higher accuracy and lower entropy for the network’s output. These positions are also more likely to contain winning moves, which lead to more stereotyped decisions. We then investigated the negative log-likelihood for the network’s predictions as a function of playing strength, computed using Elo ratings [Elo 1978]. Our network is able to more successfully capture the moves of stronger players compared to weaker ones (Figure 3.2D), since unpredictable errors in gameplay are more common in the latter group. In Figures B.2-B.4, we show example board positions for each of the previous analyses, and in Figures B.5A-B we further analyze network accuracy as a function of number of guesses and log-likelihoods for players with varying levels of gameplay experience.

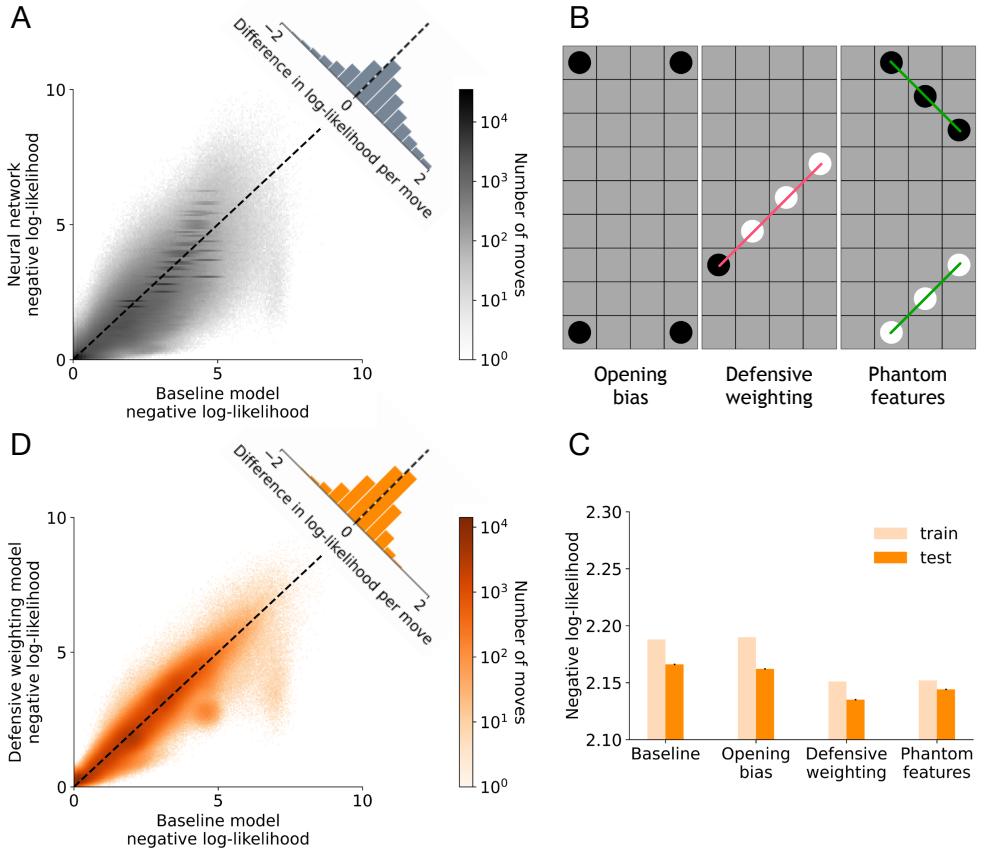
Next, we computed a set of summary statistics that characterize human play in 4-in-a-row. For each move made by each user, we calculated the distance from the chosen square to the center of the board, the distance to pieces owned by that user, the distance to pieces owned by the opponent, the distance to the center of mass of that user’s pieces, the distance to the center of mass of the opponent’s pieces, the number of that user’s pieces on the 8 squares neighboring the chosen square, and the number of opposing pieces on neighboring squares. We also indicated

whether with their chosen move, the user created a threat to win on the next move or parried a threat from their opponent. We computed these statistics for moves made by the network in the same positions encountered by human players and for random moves. Figure 3.3 shows the average of these summary statistics aggregated across all users in the test set as a function of move number. This analysis probes systematic patterns in people’s gameplay, for example a tendency to start playing near the center of the board and gradually expand outwards. For all summary statistics, people deviated considerably from random, and the neural network matched the data almost exactly. In sum, these results establish that the neural network accurately captures human decision-making in 4-in-a-row.

### 3.2.2 COMPARING THE COGNITIVE MODEL AND NEURAL NETWORK

In terms of overall performance, the cognitive model that we have discussed so far in this dissertation, which we subsequently refer to as the baseline model in this chapter, performed worse than the network on all measures that we tested. Specifically, the baseline model achieved a negative log-likelihood of 2.17 (0.30 more than the network) and prediction accuracy of 34.88% (6.83% less than the network) on the test data. Additionally, the network’s predicted log-likelihood per move was typically higher than that of the model ( $t = 322.86, p < 2 \cdot 10^{-308}$ , Figure 3.4A). The baseline model’s average accuracy per move was lower than the network’s throughout the course of gameplay (Figure B.5C), and on the summary statistics, the model deviated further from human data than the network (Figure B.6). Thus, there is room for improving the baseline model.

Having established that there exist mechanisms that describe aspects of human behavior but were overlooked in the construction of the baseline model, our goal is to identify and implement such mechanisms. An initial attempt at this might involve the traditional model development approach, namely directly comparing the baseline model to the data. However, even with the size of our data set, most board positions beyond the early game were only encountered once by human players. Therefore, many of the 2,698,483 board positions where the baseline model



**Figure 3.4:** Iterating over cognitive model extensions using the neural network. **(A)** Density plot of the difference in log-likelihood per move in the test set for the neural network and baseline model. Inset is the histogram version of the same log-likelihood difference (mean of baseline minus neural network:  $0.29993 \pm 0.00039$ ). **(B)** Model extensions derived from comparing the board positions that the neural network correctly predicted and the baseline model did not. **(C)** Negative log-likelihood for each model extension for the best set of parameters across 20 different fitting runs on the training data (light orange) as well as averaged across each move in the test set for the same parameters (dark orange). Error bars indicate the standard error of the mean for the test set. **(D)** Density plot of the difference in log-likelihood per move in the test set for the defensive weighting model and baseline model (mean of baseline minus defensive weighting:  $0.03097 \pm 0.00022$ ).

predicted that a different move was more likely than the one that humans actually made represent unpredictable random human behavior rather than a failure of the model. In fact, board positions that resulted in a low log-likelihood for the baseline model were often not predicted well by the network either, and largely seemed to be human errors in gameplay such as overlooking an immediate win or making a random move (Figure 3.5, first column). In short, direct comparisons between the model and data are not particularly informative.

	Large differences between the baseline model and the data			Large differences between the baseline model and the neural network		
Board position						
KL divergence	0.014	0.000	0.014	0.093	0.083	0.074
Baseline model	8.491	8.491	8.480	2.789	2.875	1.871
Neural network	10.489	11.893	5.731	0.068	0.332	0.054

**Figure 3.5:** Representative residuals between the baseline model and the data (first column) and the baseline model and the neural network (second column). For each board position, we report the KL divergence between the output distributions of the model and the network, as well as the negative log-likelihood of the human move for the model and the network. The user is playing black while the computer opponent is playing white. Additionally, the red shading indicates the probability distribution of the network’s move prediction, the open circle indicates the user’s selected move, and the dashed circle indicates the baseline model’s predicted move.

The neural network, however, provides a viable alternate to compare the baseline model against. To do so, we used the Kullback–Leibler (KL) divergence as a measure of the difference between the output distributions of the network and model on any given board position. By pooling information across board positions, the neural network can produce a better estimate of the difference between the model and the true human policy and can thus give better guidance for model improvements. Indeed, the largest differences between the baseline model and the neural network were more interpretable than the largest differences between the baseline model and the data (Figure 3.5, second column). After sorting the deviations, we manually inspected the board positions with the highest KL divergence and grouped positions together that shared identifiable features. Then, for each deviation, we implemented a change to the model to address the differences between the model and the network based on our understanding of the task and the model. We then validated that the change indeed altered the model’s predictions for the specific

board positions that prompted the change. For example, when we added a new component to the model’s heuristic value function, we passed a number of board positions from the subset of deviations as input, and compared across simulations that the model now predicts the correct move where the previous iteration did not. Only after this testing procedure did we fit the amended model to the data. More specifically, we identified three mechanisms that appeared to be shared across subsets of the largest deviations between the model and the network, each leading to a new iteration of the baseline model. We implemented these cumulatively: each model contains all features of the baseline model, any prior extensions, and a new extension.

### 3.2.3 TESTING CANDIDATE MODEL IMPROVEMENTS

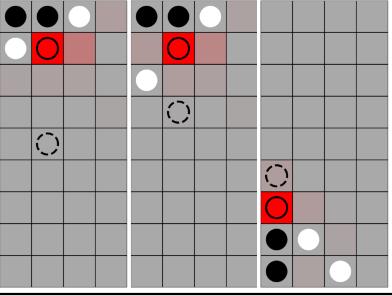
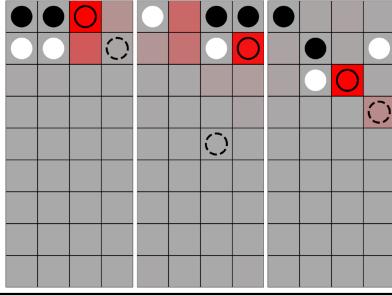
The first model extension consists of a corner bias for the opening move. The neural network comparison highlighted that users are quite likely to play in the corners in the opening, in particular in the upper left corner. There is no strategic reason for making these moves, but the network detects these preferences nonetheless. Since this pattern is especially prevalent on the first move (Figure B.5D), we added a set of parameters that can give higher value to moves being considered in each of the corners of the board (Figure 3.4B, left). In terms of implementation, this mirrors the central tendency feature that is already present in the baseline model. While this was a fairly incremental model improvement in terms of negative log-likelihood on the test data as compared to the baseline model ( $t = 4.24$ ,  $p = 2.28 \cdot 10^{-5}$ , Figure 3.4C), it serves as an initial proof of concept for our methodology.

The second model extension targets defending against opponent threats. In our analysis, we noticed that the model often overlooks immediate losses in favor of promising offensive moves elsewhere on the board, while both users and the network do not systematically make these errors. An example of this is shown in rightmost board in the second column of Figure 3.5. This is particularly prevalent when the defensive move creates no new features for the player and the player can create multiple features for themselves closer to the center of the board. The

explanation for the model’s behavior is that it assigns relatively high value to the offensive moves, causing the defensive moves to be pruned from the search tree. Thus, the defensive moves are never explored, even after the moves that the model expands preferentially are evaluated during tree search. To fix this deviation, we specify a weight in the heuristic function that explicitly recognizes immediate opponent threats (Figure 3.4B, middle). With this change, the defensive moves are now no longer overlooked, as they are almost always valued highly enough to avoid pruning. As such, the defensive weighting model significantly improved in terms of negative log-likelihood on the test data from the baseline ( $t = 33.48$ ,  $p = 8.85 \cdot 10^{-246}$ ) and opening bias models ( $t = 28.93$ ,  $p = 5.25 \cdot 10^{-184}$ , Figure 3.4C). This further validates our proposed approach, as we were able to account for an important, more complicated mechanism that we had no prior knowledge about beforehand. Without the neural network comparison, it would have been nearly impossible to detect this detrimental interaction between pruning and the heuristic evaluation in the end game.

The final model extension adds phantom features. These were inspired by positions in which users preferentially play to create or defend against features that are already part of the heuristic function but do not have empty squares contained within the feature to eventually win the game. An example of this is shown in leftmost and center boards in the second column of Figure 3.5. We define these in the 3-in-a-row case on the edges of the board, and include them in the model’s heuristic evaluation (Figure 3.4B, right). When looking at the log-likelihoods across training runs for the phantom features model, they are fairly similar overall to the defensive weighting model, with the best parameter set that we use for testing only resulting in a difference of 0.001. This final extension did not exceed the defensive weighting model’s performance on the test set ( $t = -9.87$ ,  $p = 5.54 \cdot 10^{-23}$ ), but still improved from the baseline ( $t = 24.13$ ,  $p = 1.32 \cdot 10^{-128}$ ) and opening bias models ( $t = 19.59$ ,  $p = 1.92 \cdot 10^{-85}$ , Figure 3.4C). Thus, we have no evidence that people use phantom features in 4-in-a-row.

Treating the defensive weighting model as our best model variant, we show that the model’s

	Large differences between the defensive weighting model and the neural network	Large differences between the phantom features model and the neural network
Board position		
KL divergence	0.094    0.093    0.090	0.087    0.086    0.082
Model extension	2.378    2.728    2.364	2.086    2.450    2.547
Neural network	0.317    0.332    0.162	0.404    0.814    0.224

**Figure 3.6:** Representative residuals between the defensive weighting model and the neural network (first column) and the phantom features model and the neural network (second column). The format for the board positions is the same as for Figure 3.5.

predicted log-likelihood per move is typically higher than that of the baseline model, and that this is particularly true in moves that had a high difference in terms of log-likelihood between the models (Figure 3.4D). This suggests that our added mechanisms are correctly accounting for the moves that the baseline model was initially worst at predicting. Finally, we repeated our analysis of the largest differences between the two best model extensions and the neural network (Figure 3.6). As expected, this revealed that the residuals for the defensive weighting model still contain the board positions that inspired our phantom features model, whereas the residuals for the phantom features model no longer contain these and instead highlight new deviations. This suggests that the lack of improvement shown by the phantom features model is due to a tradeoff between moves in which the phantom feature weights are helping and those in which they are not. In other words, despite accounting for the desired errors that the network is able to correctly predict, the phantom features model might require an alternate implementation. It is also possible that an entirely new mechanism altogether could account for these residual board positions. A full specification of the cognitive models that we tested and their performance is available in

Table B.2.

### 3.3 DISCUSSION

In this chapter, we trained deep neural networks to predict human moves in 4-in-a-row using a large-scale data set. We ensured that these networks estimate a reasonable upper bound on how well any model can explain human behavior by incrementally scaling up the networks and validating that any further scaling would result in marginal increases in performance. We then analyzed the best network, finding that the network captures general trends in human play. This provided us with a model that was able to predict human decisions more accurately than an interpretable cognitive model of human planning without requiring manual engineering. We then explored the positions in which the neural network was more accurate than the baseline model, leading to several candidate mechanisms for model improvement. Finally, we investigated the results from three new models that added an opening bias, defensive weighting, and phantom features, analyzing both overall goodness of fit and relative predictability compared to the neural network. Taken together, these results highlight the advantages of using deep neural networks as a guide for modeling human planning.

In comparing the neural network with our baseline model, our results suggested mechanisms that had not been previously considered and improved the model’s performance in 4-in-a-row. The defensive weighting discovery in particular is a combination of the model’s value function, forward search algorithm, and pruning mechanism that greatly affected its predictions in certain crucial positions, but would’ve been very difficult to detect without the neural network. Our findings also imply that further refinements for the model variants that we present here exist. For example, the opening bias that people display surely extends beyond just the first move and is more indicative of a faster, model-free process in the early game. Similarly, the phantom features could require a more sophisticated weighting of parameters or implementation altogether outside

of the heuristic function to avoid any tradeoffs with other moves. Additionally, our approach can facilitate the generation of completely new hypotheses to explain the remaining residuals. An example in this category is reconsidering the underlying mechanism for the moves that inspired the phantom features model. Another unaddressed but consistent residual appeared for situations when the players did not start playing in the center of the board. In these games, people and the network tended to continue to play away from the center of the board in close proximity to existing pieces, while the model preferred building new central features. In sum, our approach allows for continued discovery and testing of novel mechanisms in the cognitive model. However, for the purpose of this chapter, the existing set of extensions serve to primarily demonstrate the viability of guided model improvement via deep learning for complex models of human planning.

While our method provides a framework for guided model improvement with neural networks, it has several limitations. The first is that the effort of implementing, testing, and analyzing such networks might not be worth the effort if the task is simple enough. Many of the tasks that are utilized in psychology studies have a small enough state space that all substantially different situations can be investigated by hand and/or enough data is available for individual situations to make deviations between the raw data and model predictions meaningful. In such tasks, the utility of our approach is greatly reduced. Another limitation is that training neural networks requires large amounts of data. If less data is available, overfitting becomes a major concern for flexible architectures and the alternative of using less flexible network architectures implies biases in the network predictions that need to be justified. Additionally, such networks are inherently more challenging to train. Standardized tools for streamlining the neural network fitting process might alleviate these problems by reducing the burden on researchers to construct and analyze the networks. Collectively, such tool development might be worthwhile for cognitive science given the recent prominence of neural network-driven model improvement methods, but we do not provide such tools here. In terms of the method itself, a potential limitation is that we currently sort the trial-by-trial deviations and identify shared patterns manually. To automate

the pattern extraction process, we could apply clustering or other machine learning methods to the board positions showing deviations between the network and the model. This would result in groups of board positions that we then inspect more closely for shared features. Another alternative might be to have 4-in-a-row “experts” who have played many games and have high estimated Elo ratings interpret the deviations between the model and neural network to reduce investigator bias. This mirrors studies in the chess literature [Holding 1989a, 1992; Campitelli and Gobet 2004], and could generate new ideas for potential model improvements.

What do the mechanisms that we identify from the deviations between the model and neural network tell us about planning and human cognition? One takeaway is that people have inherent biases, meaning that they consistently prefer one out of many equivalent solutions to problems when there is no rational reason to do so. Humans display such systematic biases in many tasks [Griffiths et al. 2010], and the literature on these biases and how to model them may be informative to structure the biases players show in 4-in-a-row and while planning more generally. Our model extensions also suggest that people’s heuristic functions may be more sophisticated than a simple sum of features, accounting for complex tradeoffs between pieces on the board depending on the context of the board position. Further, we observed in earlier studies that individuals seem to evaluate positions differently, as feature weights vary when the cognitive model is fit to each participant. Adjusting the heuristic function to be more human-like and account for nuanced individual differences is a challenge, but the size of the data set paired with the neural network’s predictions can guide this process. While these specific features of gameplay are tied to 4-in-a-row, they point to the interaction between heuristic evaluations and forward search, and how either of these mechanisms may change depending on the individual and context they are placed in. These are fundamental aspects of human planning, and uncovering more nuanced intuitions for how the mind navigates this process may provide principles that generalize across planning tasks.

More broadly, our work provides a framework for model construction that makes use of both

deep neural networks and large-scale experiments. Cognitive science as a discipline has trended towards massive data sets collected via online studies, in part to obtain rich data in participants' real-world environments and clarify whether results derived from constrained laboratory tasks generalize. To this end, leveraging methods from machine learning to aid in model development is a particularly important undertaking for the field. Our approach is most useful in complex tasks where comparing a model directly with human decisions is noisy due to few repetitions of any particular state. In our case, both of the previous criteria were satisfied, albeit with a cognitive model that has already undergone rigorous testing against alternatives in previous work. It is reasonable to assume that the model refinement process would be greatly expedited in situations where tedious manual adjustments derived from intuition for the task can be avoided altogether. Therefore, we argue that this method will be valuable in the development of future cognitive models of planning as well as other complex human behavior.

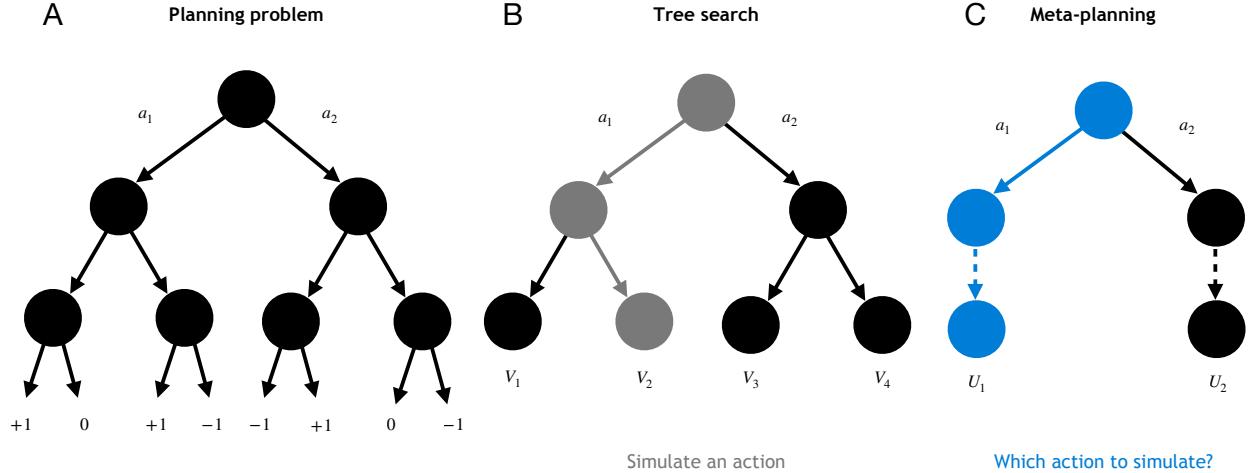
## 4 | THEORY

From spatial navigation to organizational strategy to playing games like chess and Go, planning is a fundamental mechanism underlying human intelligence that involves the mental simulation of future actions and their consequences in order to make a decision. Planning problems have typically been formalized as search over a decision tree in both cognitive science [Miller and Venditto 2021; Huys et al. 2015; van Opheusden et al. 2023] and artificial intelligence [Shannon 1950; Silver et al. 2016]. In such a scheme, an agent builds a tree of possible future trajectories where every decision is represented by a branching point (Figure 4.1A). The agent then gains information by traversing the decision tree, which is used to approximate the long-term expected reward of each currently available action (Figure 4.1B). Tree search algorithms generally lead to better decisions that may have been overlooked without planning, but can be costly to run. Even with a small cost per unit of time or planning iteration, real-world tasks involve too many possible sequences to extensively evaluate each considering the breadth and depth of the trees that an agent would need to construct.

Therefore, a growing body of literature has focused on developing solutions that humans might employ for estimating the values of choices while simultaneously mitigating the computational costs associated with planning. One plausible approach is to introduce heuristics that circumvent the intractability of an exponentially growing search tree. Such heuristics include pruning initially unpromising courses of action [Huys et al. 2012], limiting the depth of planning [Snider et al. 2015; Éltető and Dayan 2023], relying on the uncertainty or accuracy of forward

search and model-free reinforcement learning methods in tandem [Daw et al. 2011b; Kool et al. 2017; Hamrick et al. 2019], or leveraging simulated experience to further expedite the transition from goal-directed to habitual behavior [Dasgupta et al. 2018]. Meanwhile, simpler choice models of human sequential decision-making often do not explore the tradeoffs between the costs of planning and decision quality [Solway and Botvinick 2015; Tajima et al. 2019]. While each of these models provides insight into how people plan efficiently, the selected heuristics are generated via researcher intuition. That is to say, they do not provide a formal analysis of why humans might use such heuristics during planning that generalizes across environments.

More recently, a few notable exceptions to the heuristics-driven approach have made progress on providing normative accounts of human planning. These accounts optimize the metalevel problem, which is to determine in which direction the search tree should be expanded. The plan-until-habit scheme computes the value of information gained by planning in a principled manner, reducing the number of search iterations by relying on a habitual system to estimate values at the frontier of the decision tree [Sezener et al. 2019]. This results in an expansion metric that is cheap to compute, but relies on cached values that summarize past experiences. Such a method may not scale well to complex tasks where an agent almost exclusively encounters unique states, thus hindering the development of informative habits. An alternative is to frame the problem as one of resource rationality, where the agent has to minimize the cognitive operations required to make a plan while maximizing the expected utility of executing that plan [Callaway et al. 2022b]. This model is applied in small state space, deterministic environments where optimal strategies can be identified and people are encouraged to fully plan before taking any actions. Moreover, the interaction between past experiences and planning is not explicitly defined. Despite both of these efforts, the mechanisms by which people are able to approximate the values associated with potential plans in real time is not fully understood and requires a thorough mathematical derivation. In other words, a theory that explains the conditions under which people will plan while considering past experiences and scaling to large state spaces is needed.



**Figure 4.1:** Comparing planning and meta-planning. **(A)** A planning problem, where nodes represent states and arrows possible actions that an agent could make. The decision is whether to make action  $a_1$  or  $a_2$ , each of which will ultimately lead to different future rewards. **(B)** Planning as tree search, where candidate actions are iteratively simulated by expanding from the current state. The values of future states ( $V$ ) are approximated by a heuristic that informs which action will lead to the highest expected reward. In environments with many states, the tree quickly becomes costly to exhaustively search over. **(C)** Meta-planning, which aims to reduce the costs associated with planning by guiding the metalevel decision of selecting which action to plan for. In our model, this is accomplished by framing a simulation as ultimately providing more information about the true value of an available action and quickly computing the expected utility gain ( $U$ ) associated with making that measurement.

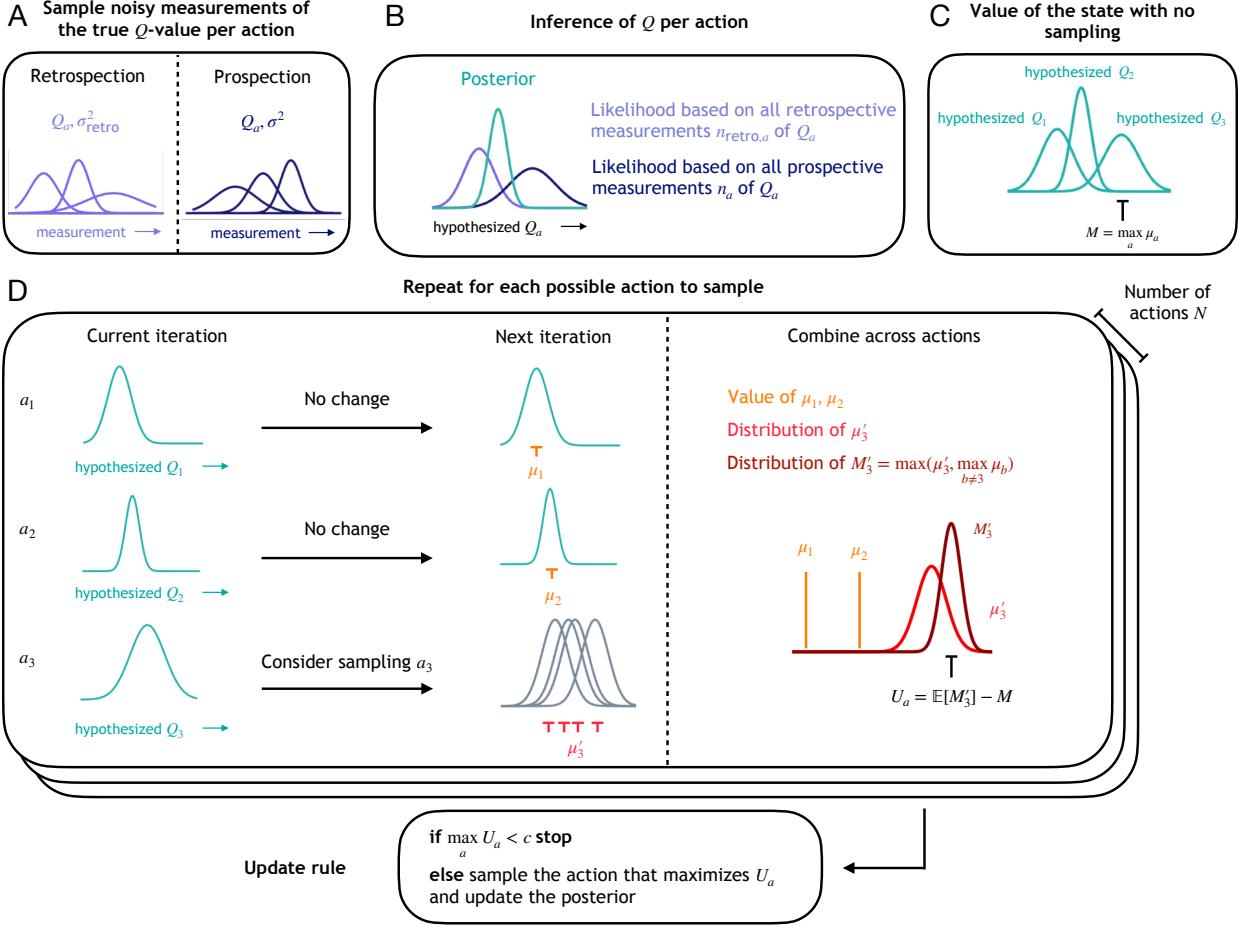
In this work, we derive an abstracted model of meta-planning, or determining which action to plan for, formalized via Bayesian inference. The key insight underlying the model is that it sacrifices the structure of a tree search algorithm in favor of a simpler, task-general statistical description that estimates the effects of planning. This is achieved by framing planning as a process for gaining noisy information about the currently available actions (Figure 4.1C). With this approach, we can precisely control parameters in the model and simulate the conditions under which people plan in a principled way that generalizes to real planning problems. Under our framework, the agent considers which action to plan for on each iteration of search by computing the expected value of planning for each action while holding the remaining actions constant. Our approach is designed to facilitate online decision-making, as the metalevel algorithm is significantly less expensive than running a full tree search and can in turn be adapted to guide plan-

ning. We show that the meta-planner makes intuitive predictions in a variety of environments, highlighting the importance of the gap between value estimates, retrospective information, and uncertainty while replicating concepts from the planning literature. We also compare our model directly against canonical search methods, namely best-first and breadth-first search, showing how it differs from these in selecting high value choices. Finally, we apply the model to a complex planning task where it captures human response time trends that previous models cannot account for. On the basis of these results, we claim that people may indeed be reasoning about which actions to plan for using principles suggested by our model as a means of reducing the computational costs of thinking ahead.

## 4.1 MODEL

The overarching goal of any planning procedure is to inform which action should be selected in the present. From a metacognitive perspective, an agent might ask “is planning worthwhile, and if so in which direction?” when it becomes intractable to evaluate every sequence of possible actions. Our Bayesian meta-planner answers this question, providing a decision rule over whether and in which direction to plan. Here we outline the underlying assumptions behind the model along with the set of equations needed to provide a general intuition for its function. A more detailed specification and derivation of the model is available in Appendix C.2.

Given a state  $s$  and a set of actions  $a_1, \dots, a_N$ , we assume that the agent has the option of executing a tree search policy  $\pi$ . Doing so is computationally expensive, but informative. We also assume that a state-action pair has a theoretical long-running expected reward under  $\pi$ ,  $Q_a$ . Since this value is not known, we take an inference view where the agent tries to build a probability distribution over each  $Q_a$ . As the agent can only consider the effects of planning for a single action at a time, the problem is deciding which action, if any, to select and better approximate. Our model uses Bayesian inference to guide this decision according to the expected



**Figure 4.2:** Formalizing Bayesian meta-planning. **(A)** The agent receives noisy measurements of the underlying  $Q$ -value for a given state-action pair. These measurements can come from retrospective experience (light purple) or a prospective planner (dark purple), each with a specific mean at  $Q_a$  and variance  $\sigma^2$ . **(B)** The posterior for each action combines retrospective and prospective information (green). This is the current probability distribution over each hypothesized  $Q_a$ . **(C)** The maximum of posterior means  $M$ , which is the highest-value choice that can be made without any additional sampling. **(D)** The inference process, where the agent computes the value of each action conditioned on sampling a specific action: this results in no change in the value of the mean for actions that aren't sampled (yellow), and a distribution over the mean for the action that is sampled (light red). To combine across actions  $N$ , the agent computes the max distribution over the possible means (which now has  $N - 1$  point estimates and a single distribution) of each action given the new sample (dark red), and  $U_a$  is the difference between the expected value of this max distribution and  $M$ . This process is repeated for all possible actions to sample. Finally, an update rule is used to decide whether it is worthwhile to keep planning. If the maximum utility across possible samples is less than a fixed cost  $c$ , then no more planning is necessary. Otherwise, the agent samples the action that maximizes utility (by, for example, executing a tree search policy) and updates the posterior for that action accordingly.

utility of making an additional measurement of each  $Q_a$ . We subsequently refer to the process of gaining new measurements within the context of the meta-planner as sampling, to reiterate the fact that there is no tree structure within the model. In practice, the samples that the meta-planner utilizes can come from a prospective planner, but we remain agnostic to the form that algorithm should take. The main idea is that planning for any action results in a better estimate of that action’s underlying value, and the meta-planner seeks to maximize utility gain relative to the goal of ultimately selecting the highest-valued action.

The agent must iterate over the available actions and compute the value of each action conditioned on sampling a single action. To do this, we assume that the true  $Q$ -value for an action follows a distribution  $p(Q_a)$  and that new measurements of  $Q_a$  are noisy and independent. These measurements can come from two sources: previous experiences, denoted by  $\mathbf{q}_{\text{retro},a}$  and captured by the retrospective likelihood  $\mathcal{L}(Q_a; \mathbf{q}_{\text{retro},a})$ , and planning, denoted by  $\mathbf{q}_a$  and captured by the prospective likelihood  $\mathcal{L}(Q_a; \mathbf{q}_a)$ . As such, we are implicitly framing the outcome of both retrospective experiences and prospective planning as additional noisy measurements of each action’s value (Figure 4.2A). Therefore, a key concept of this statistical model is that an iteration of a tree search algorithm working on a branch that starts with action  $a$  produces a new, independent measurement of  $Q_a$ . The retrospective and prospective likelihoods are a product of the individual likelihoods associated with each measurement, and the posterior is the normalized product of a prior and two likelihoods that we assume to be independent (Figure 4.2B). We compute the posterior for each action with all currently available information:

$$p(Q_a | \mathbf{q}_{\text{retro},a}, \mathbf{q}_a) \propto p(Q_a) \mathcal{L}(Q_a; \mathbf{q}_{\text{retro},a}) \mathcal{L}(Q_a; \mathbf{q}_a). \quad (4.1)$$

Since the meta-planner computes the utility gained from an additional planning operation, we define the value of the state before making an additional measurement as the maximum of posterior means ( $M$ ) across all actions (Figure 4.2C). Next, we consider the future posterior if the

agent were to sample and receive another measurement for  $a$ ,  $q'_a$ , which is unknown and has to be marginalized over. Conceptually, we are interested in the distribution over the prospective measurement one step into the future (Figure 4.2D). We compute the future distribution for  $a$  by marginalizing over the current possible values of  $Q_a$ :

$$p(q'_a | \mathbf{q}_{\text{retro},a}, \mathbf{q}_a) = \int p(q'_a | Q) p(Q | \mathbf{q}_{\text{retro},a}, \mathbf{q}_a) dQ_a. \quad (4.2)$$

To combine across  $N$  total actions, we need to compute the expected utility of making this additional measurement. Importantly, the agent can only consider sampling from a single action at a time, so the expected value of the remaining actions is known exactly. This expected value is another maximum of posterior means, this time after making the additional measurement of action  $a$ :

$$M'_a \equiv \max \left( \mathbb{E} [Q_a | \mathbf{q}_{\text{retro},a}, \mathbf{q}_a, q'_a], \max_{b \neq a} \mathbb{E} [Q_b | \mathbf{q}_{\text{retro},b}, \mathbf{q}_b] \right). \quad (4.3)$$

Note that this is a random variable because  $q'_a$  is unknown to the agent. However, we can take the expected value  $\mathbb{E}_{q'_a} [M'_a]$ . Within our framework, the reason why planning is beneficial is that the expected value of a maximum is greater than the maximum of the expected values. The expected utility of making another measurement of action  $a$ , which we call the utility of sampling, is then:

$$U_a = \mathbb{E}_{q'_a} [M'_a] - M. \quad (4.4)$$

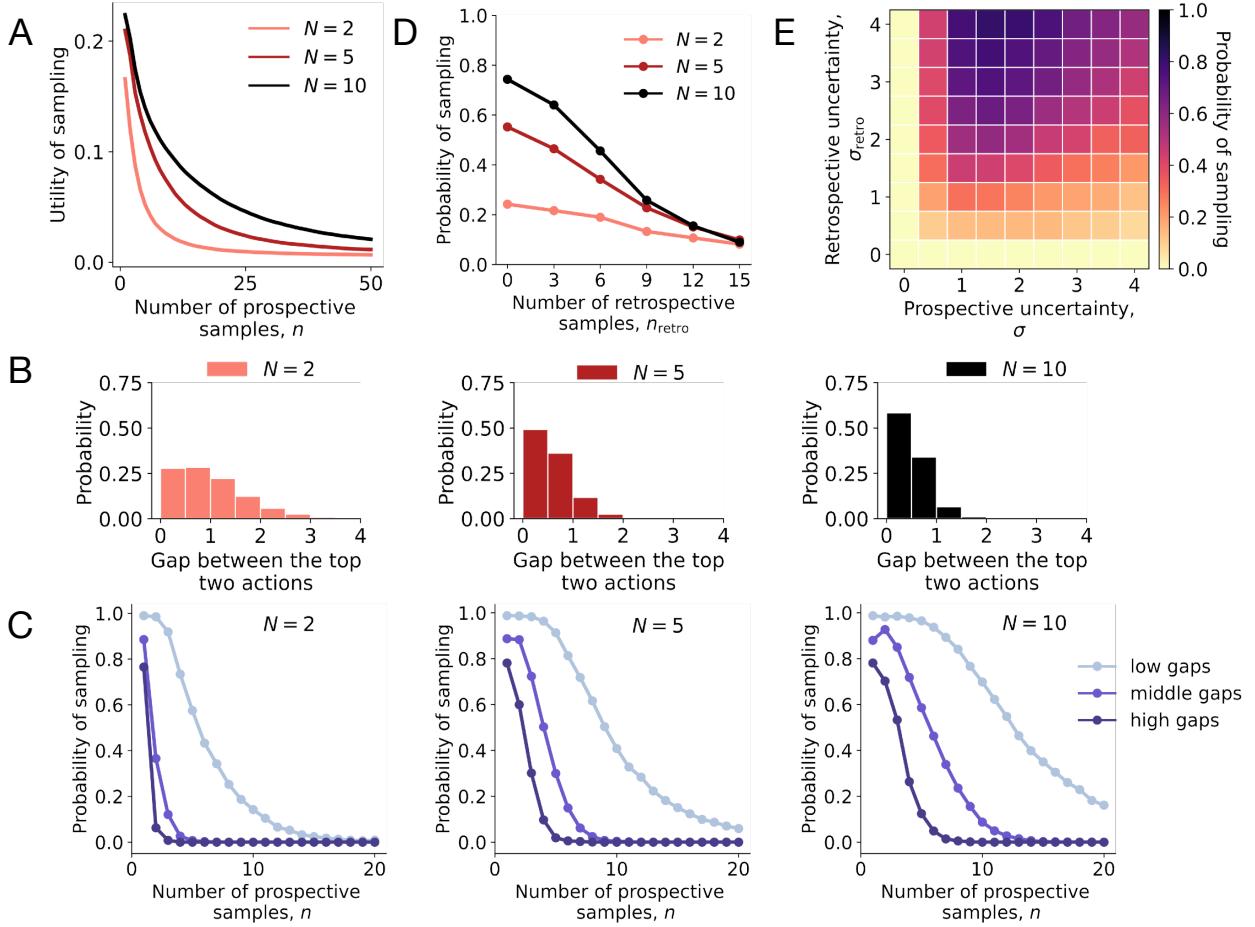
This computation has to be repeated across all possible actions to sample. Then, we propose that the agent chooses to sample if the maximal  $U_a$  exceeds a fixed cost of planning  $c$ . A new noisy measurement of the action that maximizes  $U_a$  is generated by, for example, running a tree search policy, and in turn is used to calculate the updated posterior for the sampled action. All other actions keep the same posterior. At this point, the meta-planner can use the updated values to repeat the inference process on the next iteration and decide if it is worth continuing to plan.

## 4.2 RESULTS

### 4.2.1 MODEL SIMULATIONS REVEAL PRINCIPLES FOR THINKING AHEAD

In simulations, we ran the model forward to validate that it makes intuitive predictions about when to plan. The primary goal of this section is to characterize the underlying principles that drive the meta-planner to continue sampling. We first considered the case where the agent has no prior experience and relies purely on prospective evaluations to decide how far into the future to plan. This mimics real-world planning environments where an agent has uninformed priors over their retrospective system, such as in novel tasks or tasks in which states may not repeat often. As a proof of concept, we first verified that the utility of sampling decreases exponentially as the number of prospective samples increases (Figure 4.3A). This occurs regardless of the number of alternatives, and intuitively reflects that sampling is less valuable the more samples an agent has already taken. Moving forward, we implement a cost per evaluation and report results in terms of the probability that the meta-planner will sample further. In Figure C.1A we computed sampling probability across a wide variety of costs, showing that the utility function shifts as more samples are taken.

One of the primary factors driving the shape of this utility curve is the gap in value between the top two actions [Farahmand 2011; Bellemare et al. 2016]. Suppose that the agent’s objective is to decide between two actions. If the gap between the values of these two actions is small, should the agent plan further ahead in hopes of determining which action is actually better? Or, should the agent avoid wasting resources planning, since it will be unclear which action is best regardless? And conversely, if the gap between two evaluations is large, should the agent plan more or less? In Figure 4.3B, we show that as the number of actions increases, the distribution of the difference between the top two actions, which we refer to as the action gap, becomes more right-skewed. In other words, small action gaps are more common when there are more actions



**Figure 4.3:** Meta-planner simulations. **(A)** The utility of sampling for 2 actions (light red), 5 actions (red), and 10 actions (dark red) as a function of the number of prospective samples ( $n$ ) made by the meta-planner. Panels **(B)-(D)** use the same values for  $N$ , or number of actions. **(B)** Probability distributions for the gap in value between the top two actions. The cost used is fixed to  $c = 0.1$  unless otherwise stated. **(C)** Probability of sampling as a function of the number of prospective samples made by the meta-planner, split into low (0 – 0.5, light blue), middle (0.5 – 1, blue), and high (1 – 1.5, dark blue) gap values. **(D)** Probability of sampling as a function of the number of retrospective measurements per action. **(E)** Probability of sampling for 5 actions as a function of retrospective and prospective uncertainty ( $c = 0.05$ ).

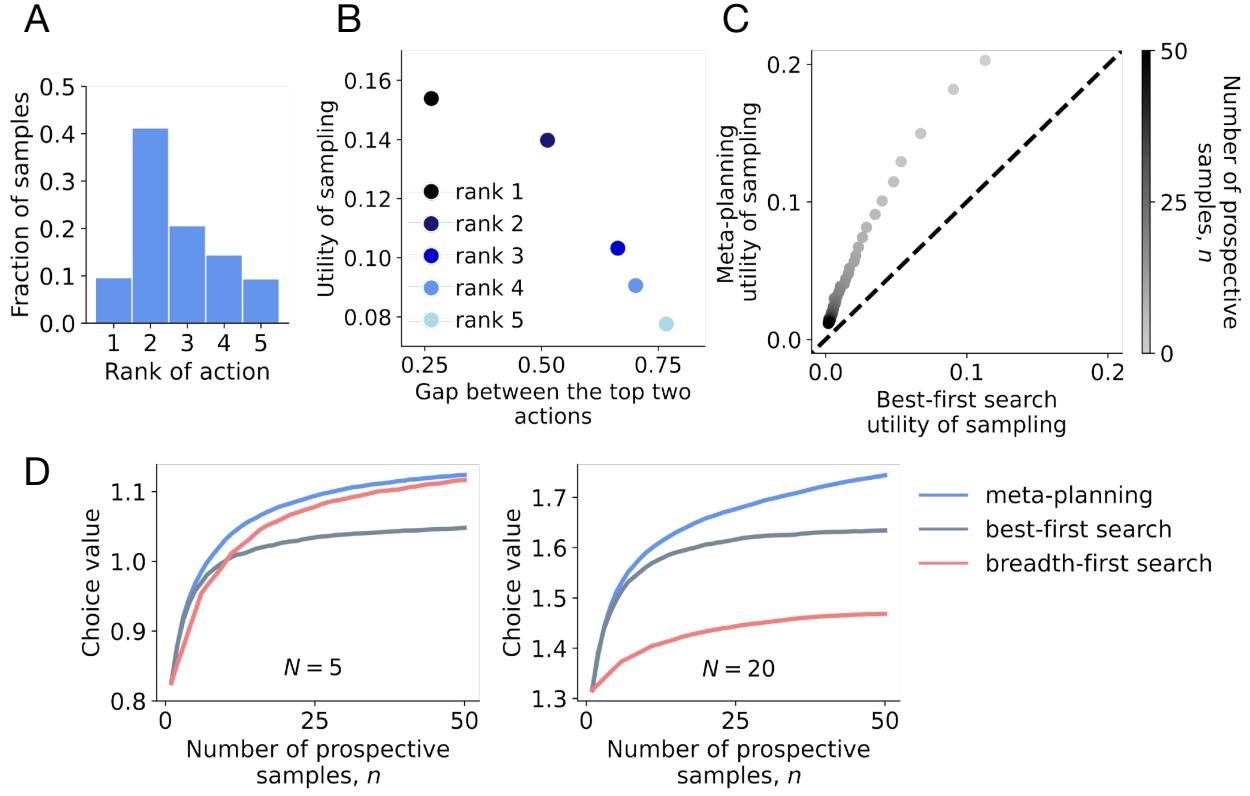
to consider, and the size of the top gap becomes more varied with fewer alternatives. Figure 4.3C then investigates the relationship between top gap size and probability of sampling: sampling is more beneficial when the action gap is smaller, and this effect occurs more often and is more pronounced with a higher number of alternatives. In Figure C.1B, we show the entire range of sampling probabilities for different combinations of action gaps and number of actions, as well as the frequency of each. In Figure C.2 we visualize the probability of sampling for different

numbers of alternatives and top gap sizes, highlighting how the shape of these curves change over time.

Next, we examined environments where the agent does have prior experience. In principle, planning should be modulated by the total amount of retrospective experience accumulated by the agent as well as the uncertainty of those estimates. These correlate directly to well-studied mechanisms in the planning literature: the transition from model-based to model-free control over time [Dickinson 1985] and uncertainty-based arbitration between prospective and retrospective systems [Daw et al. 2005]. We simulated total experience by using a variable Poisson rate ( $\lambda$ ) to determine the average number of past measurements for each action. Environments where the agent has more retrospective experience resulted in lower probabilities for sampling another measurement, once again irrespective of the number of actions that the agent considers (Figure 4.3D). The rationale behind this is that environments with low amounts of retrospective information require more planning compared to when the agent solely relies on prospectation, and the probability of sampling decreases as the agent gains more experience. In these cases, the agent can spend less resources planning and instead relies more heavily on its cost-effective retrospective experiences. We then directly varied the amount of uncertainty for both the retrospective and prospective measurements, finding that increased uncertainty with either or both sources of information leads to higher sampling probabilities (Figure 4.3E). There is, however, an asymmetry where high amounts of prospective uncertainty made sampling no longer worthwhile. The interpretation is that planning is generally beneficial in gaining high-value estimates under uncertainty, but if the uncertainty attached to prospective measurements is too high then it is no longer worthwhile to obtain these costly measurements.

#### 4.2.2 MODEL COMPARISON WITH CANONICAL SEARCH ALGORITHMS

Beyond looking at the conditions under which the meta-planner samples, we are interested in the actions that it tends to sample from and how those compare with canonical algorithms for tree

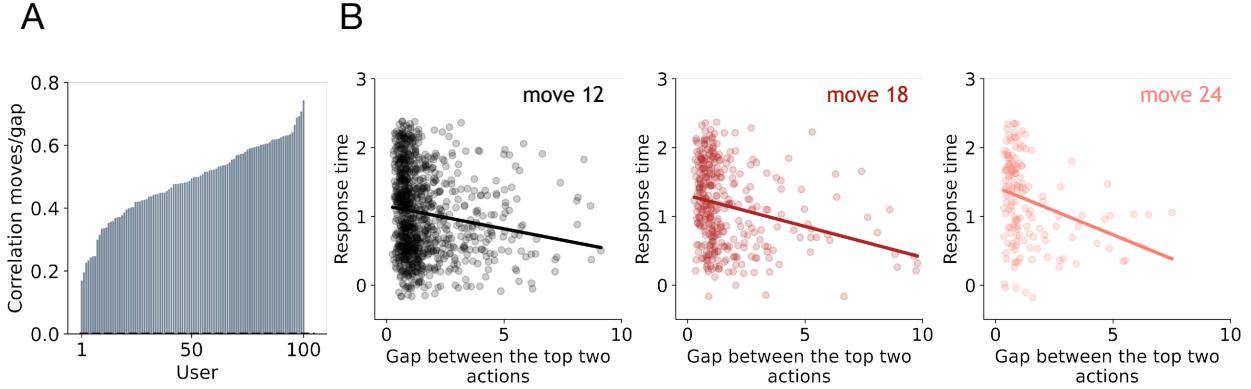


**Figure 4.4:** Comparing the meta-planner with best-first and breadth-first search. **(A)** The fraction of samples made by the meta-planner for each action rank. The number of actions used is set to  $N = 5$  unless otherwise stated. **(B)** The utility of sampling as a function of the gap between the top two actions for different action ranks. **(C)** The utility of sampling at each step of the planning process, for the meta-planner and the highest-ranked action that would be favored by a best-first search algorithm. **(D)** Choice value at 5 actions (left) and 20 actions (right) as a function of the number of prospective samples for the meta-planner (blue), best-first search (gray), and breadth-first search (orange).

search. In this section, we relate our model to best-first and breadth-first search, as these are sensible rules for expansion that have been studied in the context of human planning [Moreno-Bote et al. 2020; Mastrogiovanni and Moreno-Bote 2022]. Perhaps the simplest method for comparison is to examine the distribution of samples that the model makes conditioned on each action’s rank, where the rank of an action corresponds to its sorted order in terms of the model’s expected value for the action. The meta-planner most often sampled from the second highest-ranked action, with a decreasing number of samples for actions ranked beneath it (Figure 4.4A). This contrasts drastically with best-first search, which would always sample from the highest-ranked action, and

breadth-first search, which would sample uniformly across actions. The intuition underlying this distribution is that the meta-planner is framed in terms of utility gain, and thus will select the highest-ranked action when it is no longer worthwhile to sample. That is to say, the model is most often interested in ascertaining whether the value of the second highest-ranked action will overtake the value of the highest-ranked action and therefore be selected. The same logic can be applied to the remaining actions, with the highest-ranked action only being sampled from minimally to reduce uncertainty about its value. This is closely tied to the previous action gap result, and we plot the utility of sampling as a function of this gap (Figure 4.4B). This supports the previous explanation, as higher-ranked actions are usually sampled with low gaps where additional measurements are valuable and may reveal which action is actually better. Alternatively, lower-ranked actions are typically sampled with high gaps where new information is less valuable and the model samples more broadly since the best action is unlikely to change.

To more directly compare the meta-planner with best-first search, we tracked the development of the utility of sampling within the meta-planner’s simulations (Figure 4.4C). In general, the action that the meta-planner sampled from had higher utility than the highest-ranked action that a best-first search algorithm would select. As more samples are taken and the true value of each action is more closely approximated, the difference between the two selection rules decreased. Finally, we computed choice value, or the value of the highest-ranked action if no more samples were to be taken, for our model alongside both best-first and breadth-first search (Figure 4.4D). The distinction between choice value and value of sampling is that rather than the utility gained from receiving another measurement, we are now interested in the value associated with the choice the agent would actually make as well as how that value changes while sampling according to different models. We found that, regardless of the number of alternatives, our meta-planner consistently makes the highest value choices while best-first search does reasonably well but plateaus earlier. Meanwhile, breadth-first search performs well with few actions that allow it to gain many samples for every action, but scales poorly as the number of alternatives increases.



**Figure 4.5:** Human prospection is driven by the action gap. **(A)** The correlation between move number and gap between the top two heuristic evaluations given by the planning model’s initial evaluation for 100 users in the data set. **(B)** Response times in logarithmic space as a function of the initial gap between the top two heuristic evaluations given by the planning model. The data (circles) are shown for each position across all users conditioned on move number along with a linear regression (line).

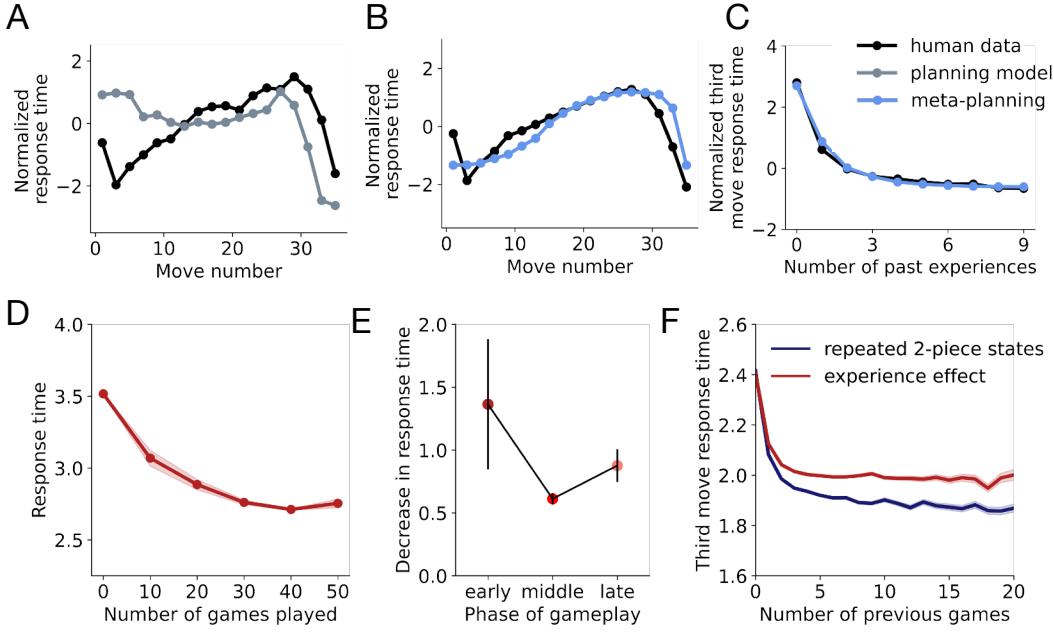
#### 4.2.3 APPLICATION TO COMPLEX PLANNING

Our simulation results highlighted a set of principles that are important for determining whether and in which direction an agent thinks ahead. Namely, these are the gap in value between the top two actions, retrospective experience, and uncertainty. To validate that the meta-planner’s predictions align with human behavior, we conducted a set of analyses in 4-in-a-row using our large-scale data set from Section 2.1.

We first wanted to replicate the action gap predictions from Figure 4.3B-C in 4-in-a-row. Since the action gap is an internal quantity that is not accessible via simply analyzing behavior, we used the cognitive model of human planning to approximate the gap. In places we utilized this planning model, we used the standard implementation from Section 2.2. To specifically investigate the gap, we computed the difference in the top two initial heuristic evaluations for every board position that each user encountered. The model assigns a node a value of positive or negative infinity if the board contains a win or loss for the current player, so we excluded positions with such nodes from our analysis. As a sanity check, we correlated this value difference with move number in the game (Figure 4.5A). This correlation should be positive, as move number within

the game has an inverse relationship to number of alternatives, and a lower number of alternatives indicates a larger gap. We did indeed observe a positive correlation across all users in the data set, indicating that the heuristic gap estimated by this model of planning is a reasonable proxy for the gap in utility values used by the meta-planner. Then, we were interested in the correlation between the amount of planning done by each user and the gap across different move numbers. For this, we used human response times as an indirect measure as is standard in the study of human planning [Russek et al. 2022]. Figure 4.5B shows a decreasing trend in response times (in logarithmic space) across all positions and users in the data as a function of gap (move 12:  $\rho = -0.197, p = 6.269 \cdot 10^{-10}$ ; move 18:  $\rho = -0.283, p = 7.651 \cdot 10^{-8}$ ; move 24:  $\rho = -0.328, p = 3.586 \cdot 10^{-5}$ ). We repeated this analysis for 3 different move numbers that tile the middle game, finding that the correlation is stronger further into gameplay. This validates one of the meta-planner’s predictions, which is that people think less when the value of the best action is significantly larger than the second-best alternative.

Next, we wanted to examine the effects of retrospection on response times in 4-in-a-row as suggested by the meta-planner. A motivating observation is that while people’s response times correlate on individual trials with the number of planning model iterations, they differ considerably on average in early gameplay (Figure 4.6A). Given the importance of retrospection in the meta-planner, a potential mechanism to explain this mismatch is that in situations where the board is fairly empty and no player can immediately win the game, there is a faster retrospective process that takes place before prospective planning begins. This also explains why response time trends in the middle and late game roughly follow the planning model’s predictions. To test that the meta-planner has the capacity to predict this trend qualitatively, we compared human response times across the entire data set with meta-planner simulations (Figure 4.6B). To mimic human experience in this task, we denoted number of past experiences as an exponential decay function across move number. Then, we simulated the number of samples taken to termination with fixed parameters with the number of alternatives dictated by the number of pieces of on



**Figure 4.6:** Human response times are driven by retrospection and uncertainty. **(A)** Average human response times (black) and number of planning model iterations (gray) taken to make a move throughout gameplay for 100 users in the data set. Response times were normalized via z-scoring here and in panels (B) and (C). **(B)** Average human response times (black) and number of samples the meta-planner makes (blue) throughout gameplay. The meta-planner is simulated with retrospective experience following a decreasing exponential function with move number. All subsequent panels are for all users in the data set unless otherwise stated. **(C)** Average human response times (black) and number of samples the meta-planner makes (blue) on the third move of gameplay as a function of the number of past experiences. **(D)** Average human response times across all moves in the data set as a function of the number of games played. This analysis was limited to the 34,810 users who played at least 50 total games. Error bars and shading denote s.e.m. here and in subsequent panels. **(E)** The average decrease in response times from the first game to the 50th game as a function of the phase of gameplay. The early game is defined as moves 1 to 5, the middle game moves 6 to 30, and the late game moves 31 to 36. The analysis was limited to the same subset of users as in (D). **(F)** Average response times on the third move of gameplay as a function of repeated 2-piece board states (blue), and the average response times of 1,000 randomly sampled users that had previously played the same number of games (red).

the board. Encouragingly, our meta-planner is then able to correctly account for the shape of people’s response time curves, something that it isn’t be able to do without retrospection at all (Figure C.3A). Another limitation of the planning model in 4-in-a-row is that it is purely prospective, and thus cannot account for decreased response times as a function of experience. In Figure 4.6C, we show that human third move response times decrease with number of past experiences and are indistinguishable from the meta-planner’s predictions. This also occurs on the opening

move, which users encounter every game (Figure C.3B).

Moving beyond qualitative descriptions of human response times using the meta-planner, we examined the concept of uncertainty in human decisions. Specifically, we hypothesized that gaining retrospective experience reduces uncertainty about the values of specific actions, and this should lead to differential response times in positions where retrospective experience is more readily available. We analyzed the subset of 34,810 users who had played at least 50 games, verifying that their overall response times reliably decreased and then plateaued in this extended experience horizon (Figure 2.1D). Then, we computed the decrease in average response times from users' first game to their 50th, split into early, middle, and late game moves (Figure 4.6E). This decrease was much greater in the early game compared to the middle game ( $t = 3.065$ ,  $p = 0.009$ ), suggesting that the first few moves where board positions are more likely to repeat drive response times down with increased retrospection. There is also a significant effect between the middle and late game which is not captured by retrospection ( $t = -2.430$ ,  $p = 0.030$ ), but rather by the fact that uncertainty is low in the last few moves of a game that will almost certainly result in a draw. Furthermore, third move response times decreased significantly when users encountered repeated 2-piece board states (Figure 4.6E). This could be a confounded result, since on average users move faster after playing multiple games regardless of which states occurred. To address this, we ran a control in which we sampled the average response times of other users that had played the same number of games, explaining some of the effect but not all. Together, these results provide further evidence for the meta-planner's predictions, namely that prior experience and uncertainty play a functional role in decreasing the amount of planning that people do. Figure C.3C-D provide additional evidence for retrospective response times in 4-in-a-row, and Figure C.4 shows how people's actions in the early game are influenced by game outcomes.

## 4.3 DISCUSSION

In this chapter, we introduced a normative model for meta-planning based on Bayesian inference that iteratively computes the value of sampling per action and decides whether and in which direction it is informative to plan. The model is abstracted in the sense that it frames tree search as gathering additional noisy measurements about the true value of each currently available action, which in turn allows us to systematically manipulate parameters to derive principles for thinking ahead. We showed that this model makes intuitive predictions about the probability of sampling over the course of planning as a function of different parameters. Then, we investigated how the value of the actions that the meta-planner samples compared with those explored by canonical search algorithms. Finally, we showed that the concepts underlying the model’s behavior can be applied to a complex planning task and account for previously unexplained trends in human response times. Ultimately, our framework provides a mathematically rigorous and cost effective method for guiding tree search that maximizes expected reward.

We must also consider how this framework might interact with a prospective planner in real tasks, and if there is any evidence for meta-planning in the brain. In this work we deliberately introduced the theory and used it to generate testable predictions that we find evidence for in human data. This simplicity is a strength of the model in that it allows us to precisely control the meta-planner’s mathematical properties, but also a limitation in its applicability. In future work, we hope to extend the framework to fit behavior directly. The first step towards this could be to investigate how our model scales to problems where the decision tree structure is preserved, to get an intuition for how the model’s predictions deal with such environments. In terms of model fitting, one viable approach is to extend an existing planning algorithm with concepts from the meta-planner. For example, in 4-in-a-row we could implement more sophisticated stopping and expansion rules to replace the simple heuristics that are currently being used. A search algorithm that utilizes the action gap, number of alternatives, retrospective information, and uncertainty to

make more informed decisions about when to plan and which actions to plan for would hopefully better fit human choices and predict response times. Since we have conceptualized this as a metacognitive algorithm, another natural extension is that an agent actually uses a meta-planner to quickly make judgements about whether to run another iteration of a prospective algorithm. In this case, the mind would actually be implementing such an algorithm alongside decision tree search. In practice, this approach might require an amended meta-planner with task-specific features and structure within the model’s evaluations. In either case, adapting the meta-planner to well-characterized planning tasks as well as novel, complex tasks like the one used throughout this work is a challenge that we leave for future work.

One assumption that we make is that meta-planning can be framed as Bayesian information sampling. As such, the method by which our model approximates the effects of search can be linked to a few different fields. Perhaps the most obvious is the resemblance to the information sampling literature, where the objective is to choose the single most rewarding given a number of alternatives. The information that is sampled can be perceptual or value-based, and there is evidence that planning and information seeking share similar neural mechanisms [Hunt et al. 2021]. More recently, related work has claimed that simple decisions are made by integrating noisy evidence that is sampled over time in a Bayesian manner [Callaway et al. 2021; Jang et al. 2021]. Our framework can be viewed as an approximation to planning via an optimal information sampling algorithm, and shares many features with these models. Conceptually, the main difference is in domain application, as prior work has explained fixation data in choice tasks with few alternatives while our model aims to derive intuitive rules to guide sequential decision-making. This is particularly relevant to the form that our model will take when interacting with a forward search algorithm in complex planning tasks.

Another connection is that our abstracted framing of planning can be interpreted as a multi-armed bandit (MAB). In such problems, people must choose between a set of alternatives that each have unknown reward in order to maximize total expected reward. This kind of task is

typically used to study sequential decision-making under uncertainty and captures the tension between exploration and exploitation [Gershman 2018]. Historically, MAB tasks have been studied across many domains including statistics [Gittins 1979; Auer et al. 2002; Chapelle and Li 2011], reinforcement learning [Kaelbling et al. 1996; Sutton and Barto 2018], and psychology and neuroscience [Daw et al. 2006; Cohen et al. 2007]. Further, Bayesian analyses of bandit problems exist, albeit typically providing a closed-form solution [Steyvers et al. 2009]. Since optimal solutions are intractable for humans to compute, it is thought that people employ heuristics for directed search [Lee et al. 2011; Payzan-LeNestour and Bossaerts 2011; Zhang and Yu 2013; Wilson et al. 2014]. However, human reasoning in a MAB and planning differ in that the goal in the former is to maximize the sum of rewards obtained over time, while in the latter all that matters is the value of the decision made after all simulations are done. Crucially, MAB tasks provide real rewards at every step, while planning is an internal process used to determine a single rewarding action. Our framework is specifically built to focus solely on estimating high-value decisions rather than the tradeoff between exploration and exploitation.

A final relationship with our model of meta-planning can be made to Monte Carlo tree search (MCTS). As a reminder, MCTS is a heuristic search algorithm that incrementally constructs a search tree in promising regions of the state space to approximate state-action values [Browne et al. 2012]. MCTS also employs a tree policy, the most popular of which is an application of a MAB called UCT. UCB1 assigns scores to actions using their expected value, number of visits, and an exploration bonus, and UCT applies this recursively to action selection in decision trees. Our meta-planner differs from MCTS in two fundamental ways. The first is in how the values of states and actions are computed, which for MCTS is based on rollouts guided by the tree policy. In our case, this is approximated via Bayesian inference over the effects of another prospective sample combined with retrospective information. Second, as with MAB problems, UCT’s focus is to ensure high net simulated value across actions, often overlooking expansions with low rewards even though they might result in a better final decision [Tolpin and Shimony 2012; Hay et al.

2014]. In sum, simulating further for specific actions is only valuable while planning because it helps select the best action, something that our meta-planner explicitly takes into account.

Over the past few years, our collective knowledge of human planning has progressed significantly. Beyond the shift from heuristics that reduce the costs of building large decision trees to normative models of how people think ahead, recent work has uncovered how planning depth changes with expertise [van Opheusden et al. 2023], the link between hippocampal replay and future decisions [Mattar and Daw 2018], task decomposition and computational representations while planning [Ho et al. 2022a; Correa et al. 2023], and machine learning methods for improving models of planning [Kuperwajs et al. 2023]. These findings, coupled with the movement towards naturalistic tasks and large-scale data sets in psychology [Schulz et al. 2019; Steyvers and Schafer 2020; Brändle et al. 2022], promise to eventually yield more precise characterizations of the cognitive processes underlying planning. In our view, a normative approach to derive principles for how people plan is a meaningful contribution towards this cause.

## 5 | LEARNING AND MOTIVATION

Learning in the real world is influenced by many disparate factors. One particularly natural relationship to consider is the one between learning, measured by performance in a given task, and motivation, measured by engagement with the same task. For example, imagine a graduate student whose scientific skills develop as they take coursework, conduct more research, and receive mentorship from their advisor. Throughout this process, having a paper or grant accepted might increase their desire to pursue an academic career, while rejection might lead to them dropping out of their program altogether. In practice, the directionality of this relationship is difficult to characterize irrespective of the task at hand. Do people drop out because of changes in performance? Or alternatively, do people who are closer to quitting already exhibit drops in performance that are driven by a loss of motivation?

To begin to understand how performance and engagement are functionally related, we can turn to two sets of literature within cognitive science: skill acquisition and intrinsic motivation. Early work on skill acquisition comes from modeling efforts that seek to define simple rules for learning as a function of practice, such as power and exponential laws [Newell et al. 1980; Heathcote et al. 2000]. Recently, more nuanced views of learning have emerged that emphasize not only practice, but also grit and perseverance as methods of improving performance [Stafford and Dewar 2014; Duckworth and Gross 2014]. Note that these underlying mechanisms are already inevitably tied to task engagement despite not explicitly taking it into account. Additionally, both applied and theoretical accounts of self-regulated learning are active areas of research where the

emphasis is often on what strategies people employ to select useful information that supports their own learning [Gureckis and Markant 2012; Lieder and Griffiths 2017]. While this is an important aspect of human cognition in environments where learners exert considerable control over aspects of their learning requirements, this remains a categorically different question than the factors that determine participation and influence performance. Online platforms have also established themselves as a viable empirical tool for investigating skill learning over the past few years [Donner and Hardy 2015; Huang et al. 2017].

Meanwhile, work on intrinsic motivation provides conflicting evidence. On the one hand it is known that moderate challenges encourage people to continue with a task, while extremely easy or difficult tasks reduce motivation [Schmidhuber 2010; Geana et al. 2016; Ten et al. 2020]. Other studies have investigated the peak-end rule, which is a psychological heuristic observing that people's retrospective assessment of an experience is strongly influenced by the intensity of the peak and final moments of that experience [Miron-Shatz 2009; Cockburn et al. 2015]. Since these events are formative for users' perceptions of a task, they may be particularly indicative of motivation levels throughout a task as well. Work on human gameplay lies somewhere in between these extremes, finding that higher rewards impact a desire to continue while peak and end experiences can have varied effects, one of which is to bias people's assessments of both game difficulty and enjoyment [Farzan et al. 2008; Gutwin et al. 2016]. Finally, a recent paper designed a novel task to study how people decide when to persist and when to quit, finding that people largely behave in accordance with the optimal strategy [Sukhov et al. 2023]. This strategy explores a relatively small number of options before settling on a sufficiently good option, abandoning an option if the number of remaining trials exceeds a performance threshold. Together, these provide a broad spectrum of potential factors that might cause people to engage more or less with a task, from intermediate difficulties over the course of the task to very specific experiences and reward structures.

Given that disentangling the relationship between learning and motivation is a major chal-

lenge, joint analyses of these cognitive processes are needed. Recently, a number of modeling efforts have been proposed to quantify learning and dropout simultaneously. One approach in this domain is to investigate performance and practice, finding correlations between the two as well as evidence for especially large improvements at the end of a session, or peak-end effects, and lower quitting rates with score drops, or persistence effects [Agarwal et al. 2017]. This study used  $\epsilon$ -machines, a type of hidden Markov model, to describe the optimal transitions between states in terms of human performance and quitting behavior. Alternatively, a related paper leveraged another established model class, specifically Bayesian geometric models with a hierarchical implementation, to model the distribution of how many items people collect in real-world decision tasks before they quit [Okada et al. 2018]. A final study provides an empirical investigation of the effects of participation and withdrawal on aggregated learning functions in the context of an online training platforming where people make voluntary choices about participation [Steyvers and Benjamin 2019]. However, these accounts are only a starting point for describing the dependencies between learning and motivation, and the field needs rich data sets and well-founded process models to gain a deeper understanding of how this interplay works in complex tasks.

In this chapter, we leverage the existing large-scale 4-in-a-row data set to study learning and motivation in the context of a combinatorial planning task. We begin by computing the endpoint of participants' learning trajectories, establishing that there is a correlation between final playing strength and overall experience as a result of gameplay. Then, we characterize the task components that drive task performance and engagement. To more precisely study the learning process, we investigate playing strength changes with experience, finding that users improve as a function of games played but ultimately plateau. In analyzing dropout behavior, we determine the conditions under which people are more likely to stop playing, namely high or low Elo ratings or large positive changes in their ratings. Then, we analyze the effect of physical time between games and opponent playing strength on performance and engagement. Finally, we construct a dynamic model that captures how people's playing strength evolves relative to the

time at which they play games and replicates our prior empirical results. Driven by behavioral findings in massive data, our work aims to derive a process-level model that explains the cognitive mechanisms underlying learning and motivation in human gameplay.

## 5.1 RESULTS

Despite the fact that 4-in-a-row was initially designed to study complex planning, it serves as a testbed for investigating human reasoning and decision-making at large. This is a key advantage of collecting large-scale behavior in naturalistic tasks: the resultant data sets can be repurposed to explore a wide array of questions. In Section 2.1, we outlined the desiderata that qualify a task as a promising candidate for studying human planning. We can restate these criteria in the context of learning and motivation:

1. The state space should be large enough so that participants continue to encounter states not previously experienced and the task remains challenging.
2. The task should be novel, so that all participants are in the steep part of their learning curves.
3. The task should have simple rules, so that improvements are not due to participants learning the rules but rather learning strategies.
4. The task should be engaging, so that participants remain motivated for many sessions.
5. The task should be amenable to computational modeling.

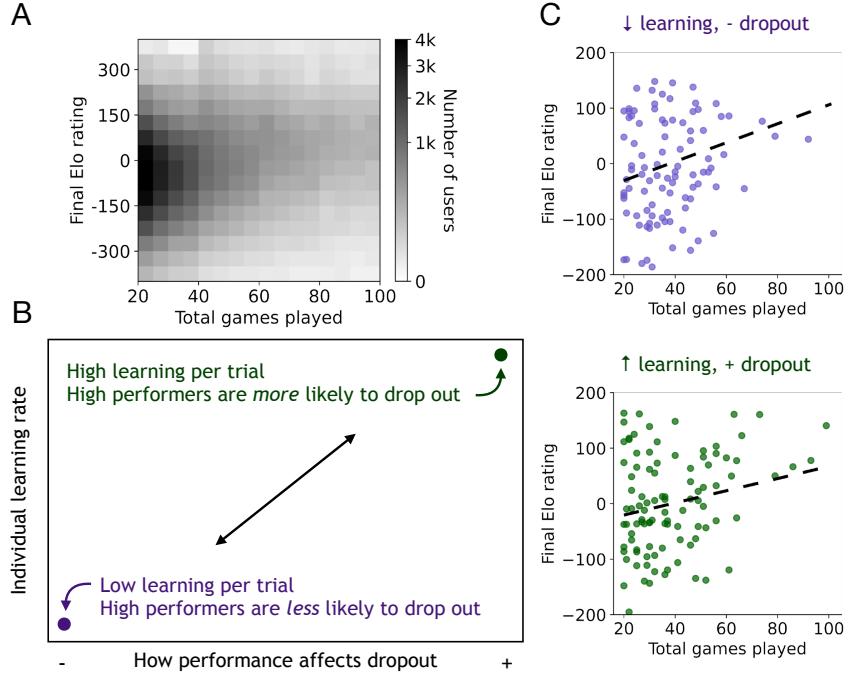
As previously stated, 4-in-a-row satisfies these competing requirements by balancing complexity and computational tractability. Additionally, the massive size of the data set allows us to run the analyses necessary to simultaneously interrogate people’s learning trajectories as well as their dropout behavior.

### 5.1.1 CHARACTERIZING THE RELATIONSHIP BETWEEN LEARNING AND DROPOUT

In order to motivate the rest of our analyses, we first investigated the relationship between task performance and total experience. We measure users' task performance using Elo ratings [Elo 1978], again for players with at least 20 total games played grouped into blocks of 20 games. This results in 115,968 unique users, and we use a common baseline to compute Elo ratings across all experimental data. Throughout this chapter, we condition each analysis on a specific range of games played, resulting in a variable number of participants. These are summarized in Table D.1, with the corresponding distribution for each quantity across the population shown in Figure D.1. In Figure 5.1A, we correlate final playing strength, or Elo rating in the last full block of gameplay, with the total number of games played by each user ( $\rho = 0.270$ ). Due to the size of the data set, our p-values are below the minimum representable float ( $2 \cdot 10^{-308}$ ) unless reported otherwise. This result illustrates that, at the endpoint of their learning trajectories, users are more likely to have higher task performance if they have accumulated more total experience with the task. However, this provides no insight into how people get to this point.

To characterize this process, we can think about the relationship between learning and dropout didactically. We start with the idea that two factors might be at play: each individual's learning rate, and how task performance leads to dropout (Figure 5.1B). If we take as an example an extreme data point in the bottom left corner, this would be a user who doesn't learn a lot per trial, meaning their playing strength stays relatively constant, and is unlikely to drop out with high scores, meaning they are persistent. In other words, initial abilities are the primary factor driving dropout behavior. Conversely, we can imagine another extreme user in the top right corner who learns a lot per trial and is more likely to drop out with higher scores. This would mean that change in performance over time is more important for dropout and there is a ceiling where they don't engage anymore.

To illustrate how framing learning and dropout in this manner can lead to the final corre-



**Figure 5.1:** The relationship between task performance and total experience. **(A)** 2-dimensional histogram of the final Elo ratings and total number of games for the 104,681 users who played at least 20 and less than 100 total games in the data set. **(B)** Didactic relationship between individual learning rates and the positive or negative effect of performance on dropout. A user in the bottom left corner of this diagram may have low learning per trial but be less likely to drop out with high performance, while a user in the top right corner might have high learning per trial but be more likely to drop out with high performance. **(C)** Simulation results for 100 pseudo-users that had their initial Elo ratings drawn from a normal distribution, learning rate drawn from a log-normal distribution, and stopping probability parameterized as a logistic function. Adjusting the model parameters to correspond with the extreme conditions in (B) and simulating playing strength increases per game until each user dropped out resulted in comparable correlations between final rating and games played as in (A). Each circle represents a pseudo-user and the dotted line a linear regression.

lation between ratings and total games that we observed in the data, we ran simulations for a fixed number of 100 pseudo-users. For each pseudo-user, we drew an initial Elo rating from a normal distribution centered at 0 with  $\sigma = 100$ . Then, we drew a learning rate from a log-normal distribution at each step of the simulation to increase or decrease each users' rating, and a logistic function of the following form dictated when users dropped out:

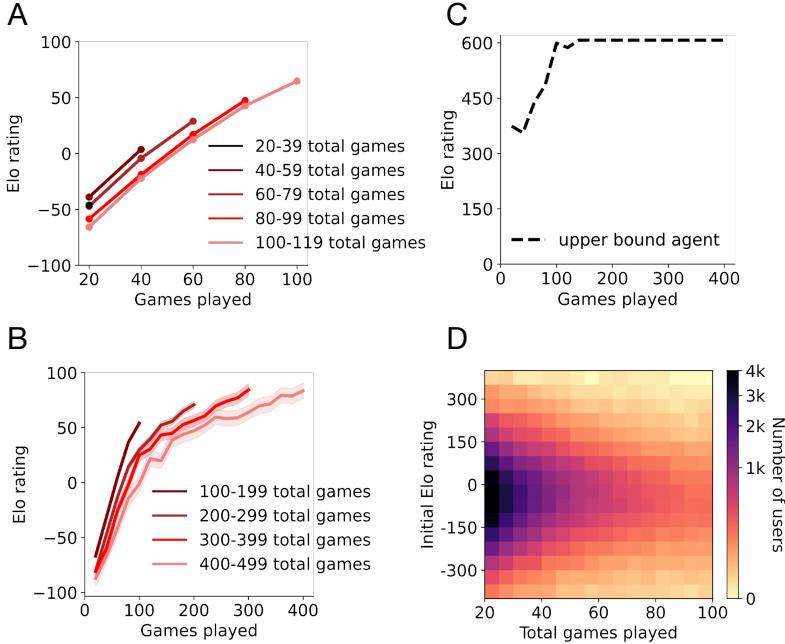
$$\frac{1}{1 + e^{-k(x-x_0)}}. \quad (5.1)$$

$k$  is the logistic growth rate and  $x_0$  is the value of the function's midpoint. Logistic regression models have been used extensively in survival analysis to characterize time to event data as hazard functions [Cox 1972], albeit in our application we are using the number of games played as a proxy for time. In the low learning, negative performance-dropout condition (Figure 5.1C, top) we parameterized the learning rate function with  $\mu = 0$  and  $\sigma = 0.1$  and set  $x_0$  to  $-200$  and  $k$  to  $100$ . For the high learning, positive performance-dropout condition (Figure 5.1C, bottom) we parameterized the learning rate function with  $\mu = 0$  and  $\sigma = 1$  and set  $x_0$  to  $200$  and  $k$  to  $100$ . In the latter case, we also added a small random dropout probability of  $0.05$  on each game. We then ran this model in each condition for  $100$  steps, recording the game number that each user dropped out at. Despite these two conditions having distinct simulation parameters, they both resulted in a similar correlation between final Elo rating and total games played to the data ( $\rho = 0.185$  and  $\rho = 0.157$ ).

Of course these are edge cases, and people may actually lie anywhere in the subspace between the two data points in Figure 5.1B. We don't consider the other two corners of the diagram because they wouldn't result in a similar correlation between final playing strength and total games played. More specifically, a user in the bottom right corner would not learn much and not persist, leading to a nonexistent correlation. Meanwhile, a user in the top left corner would have an extremely high learning rate and persist, meaning that the correlation should be much greater than what we observe with people not dropping out almost at all if they play well. In the following sections, we break down the factors contributing to learning and dropout that drive this correlation between task performance and total experience, attempting to specify where along this spectrum people behave in 4-in-a-row.

### 5.1.2 PLAYING STRENGTH AS AN INDICATOR FOR BEHAVIOR DURING GAMEPLAY

To make more precise claims regarding the factors underlying users' learning trajectories, we first validated the result from Section 2.3 that a reliable increase in playing strength over time



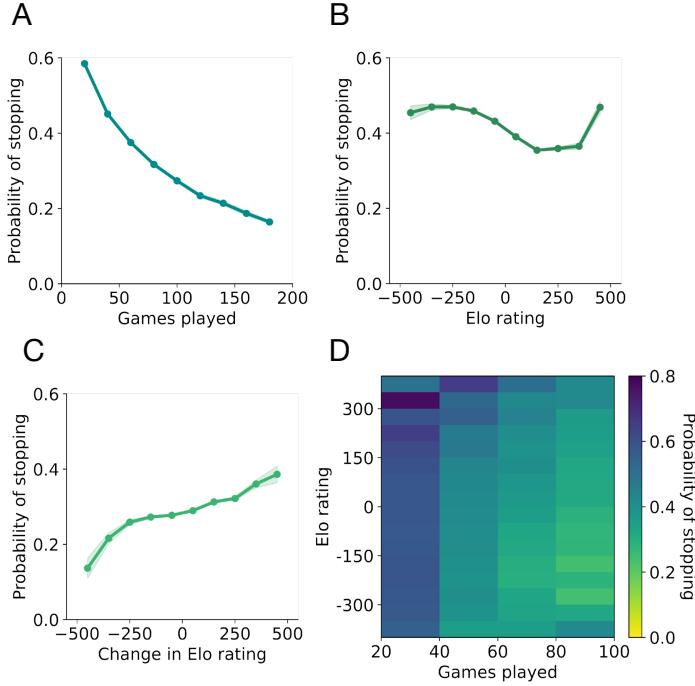
**Figure 5.2:** Playing strength increases during learning. **(A)** Average user Elo rating as a function of current experience level conditioned on total number of games played. This is computed for the set of 107,769 users who played at least 20 and less than 120 total games. **(B)** Same as (A), but for the 10,698 users who had played at least 100 and less than 500 total games. Shading denotes s.e.m. **(C)** Elo rating as a function of current experience level for an upper bound agent that wins every game against the computer opponents in the data set. **(D)** 2-dimensional histogram of the initial Elo ratings for the 104,681 users who played at least 20 and less than 100 total games in the data set.

occurs at the population level. In Figure 5.2A, we show that average Elo ratings increase as users gain more experience, and that this trend occurs irrespective of the total number of games each user ends up playing from 20 up to 119 total games (linear regression:  $\beta = 1.500 \pm 0.893 \cdot 10^{-2}$ ). We also find a reliable, albeit smaller, effect of initial Elo ratings on current playing strength (linear regression:  $\beta = 0.664 \pm 0.160 \cdot 10^{-2}$ ). Note that the correlation from Figure 5.1A can be observed by connecting the endpoints of each learning curve. Importantly, we find no evidence for changes in the slopes of users' average learning trajectory when conditioned on either total number of games played or initial playing strength. In Figure D.2, we provide additional evidence that Elo ratings are a reasonable proxy for task performance in our data set.

Given the baseline that people are improving at the task, we investigated a number of be-

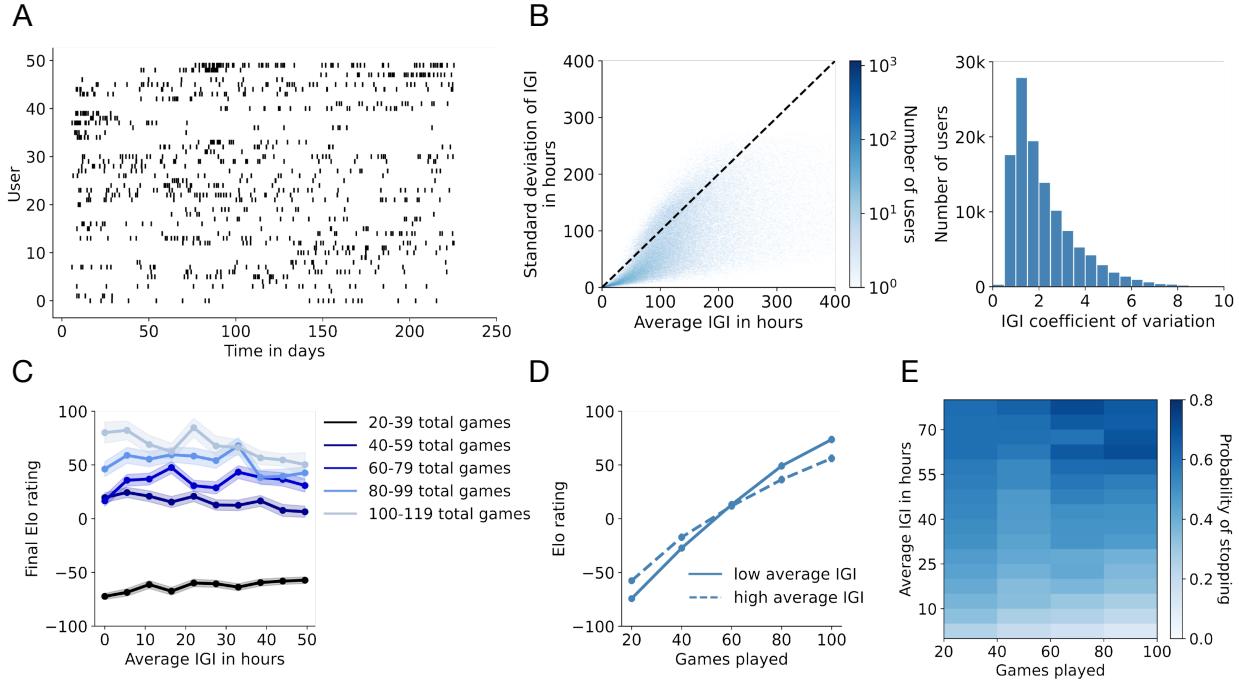
havioral signatures further. The first is people's learning rates, which seem to be close to linear despite established literature suggesting that skill learning is exponential. When we shifted the range of total games played to a range from 100 up to 499 games, we found that learning does indeed begin to plateau with significant experience (Figure 5.2B). Additionally, we verified that the flattening of the learning rate is not due to a ceiling effect. To do this, we computed the Elo ratings per block for an upper bound agent that won every game against the same computer opponents that people played against. This agent showed a similar pattern to users with an increase in playing strength that plateaus, but at much higher ratings than people reach on average (Figure 5.2C). This agent also learns significantly more quickly and converges to a fixed Elo rating. Finally, we examined the order of the learning curves by analyzing the correlation between initial rather than final playing strengths and total games played (Figure 5.2D). The effect of initial Elo ratings is small, meaning that we can exclude it as a major indicator of dropout behavior ( $\rho = -0.024$ ,  $p = 7.427 \cdot 10^{-16}$ ). This suggests that learning rates throughout gameplay are primarily captured by current playing strength, current experience level, and individual differences.

In terms of dropout, we hypothesized that quitting in 4-in-a-row is strongly influenced by current number of games played and current playing strength. In order to test this hypothesis, we examined probability of stopping, which is the probability that users' total experience is contained within the next block of 20 games given a current block, as a function of a variety of factors. In Figure 5.3A, we show that, on average, people are persistent: if they have played more games, they are less likely to drop out. In Figure 5.3B, we computed probability of stopping as a function of Elo rating, finding that people are least likely to drop out if they presently have intermediate playing strengths. In other words, they are more likely to drop out with very low or high task performance. We also investigated one additional factor, which is how the change in playing strength from one block to the next affects stopping probability (Figure 5.3C). Interestingly, this is monotonically increasing, suggesting a peak-end rule where people are more likely to drop out when they experience a significant increase in their Elo ratings. Figure 5.3D provides a 2-



**Figure 5.3:** Dropout behavior is driven by number of games played and recent playing strength. **(A)** Probability of stopping during the next block of 20 games as a function of games played so far. Results are grouped into 10 bins, and analyses were conducted for the 115,968 users who had played at least 20 games in all panels unless otherwise stated. Shading here and in (B) and (C) denotes s.e.m. **(B)** Same as (A), but for current Elo rating. **(C)** Same as (A), but for change in Elo rating from one block to the next for the 48,156 users who had played at least 40 games. **(D)** Probability of stopping during the next block of 20 games as a function of both the Elo rating and number of games played so far in the current block.

dimensional summary of this information, highlighting that as users play more games, they have a lower stopping probability (logistic regression:  $\beta = -0.020 \pm 0.058 \cdot 10^{-3}$ ) and their stopping probability increases with higher Elo ratings (logistic regression:  $\beta = 0.631 \cdot 10^{-3} \pm 0.010 \cdot 10^{-3}$ ). Each of these results maps on to distinct aspects of the intrinsic motivation literature, namely that if users devalue a task due to lack of a challenge or information gain or find the task too difficult, they will stop participating. Additionally, our analyses uncover signatures of behaviors consistent with both perseverance and peak-end effects. In sum, motivation in 4-in-a-row seems to align with previously discovered biases in human behavior by solely analyzing the development of users' gameplay. One potential concern when investigating dropout is that the subset of data we are utilizing classifies people who continue playing after the data collection period as people



**Figure 5.4:** Physical time as a factor in gameplay. **(A)** Raster plot for the time course of play for 50 randomly selected users in the data set who had played at least 20 games. Each row is a user, while each black bar is a game played. The x-axis indicates time in days, where 0 is the start of the data collection period. **(B)** Coefficient of variation (CV) for the 115,968 users in the data set who had played at least 20 games. Shading denotes s.e.m. CV is computed as the ratio of the standard deviation and the mean of the inter-game interval (IGI) in hours, visualized as a density plot (left) and histogram (right). The mean of the CV distribution is 2.245, and its variance is 2.213. **(C)** Average final user Elo rating as a function of average IGI in hours and conditioned on total number of games played. This was computed for the set of 115,968 users who played at least 20 total games, grouped into 10 bins. **(D)** Average user Elo rating as a function of games played, conditioned on low (less than average) or high (greater than average) IGI in hours. This was computed for the set of 3,088 users who played between 100 and 119 total games, grouped into 10 bins. **(E)** Probability of stopping during the next block of 20 games as a function of both the average IGI in hours and games played so far in the current block. This was computed for the set of 104,681 users who played between 20 and 99 total games.

who quit. In Figure D.3, we provide a number of analyses showing that these edge effects are insignificant across the population.

### 5.1.3 TIME AND OPPONENTS AS ADDITIONAL FACTORS

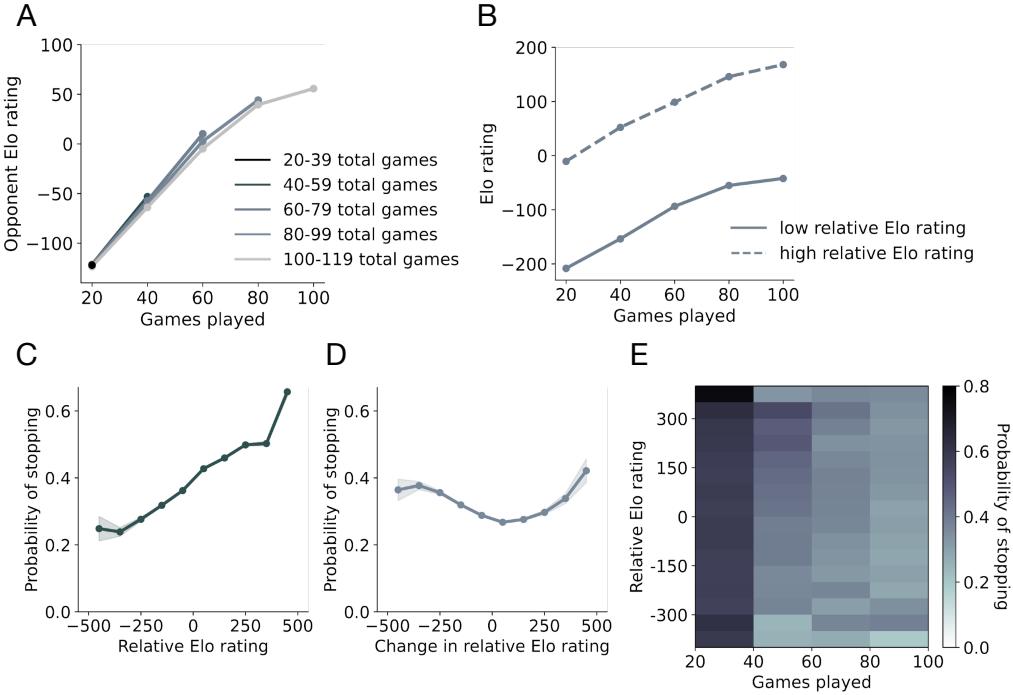
Thus far, we have analyzed dropout as a binary decision, where users continue playing until they decide to stop doing so. However, building towards a more complete understanding of motivation

requires considering the full time course of people’s playing history. To do so, we take inspiration from the computational neuroscience literature, which visualizes neural activity over time using raster plots. In a typical raster plot, each black bar represents one action potential, and each row represents the spiking activity of a neuron over the duration of recording. We adapt this formulation by replacing spikes with games played, where each row is the frequency of games played for one user over the period of data collection (Figure 5.4A). A common method for modeling neural spike trains is by assuming a Poisson distribution, which gives the probability of a neuron firing a certain number of times within a given interval [Shadlen and Newsome 1998]. This assumption requires evaluating the coefficient of variation (CV), or the ratio of the standard deviation and the mean of the data. A Poisson process produces distributions with CV equal to 1, and in Figure 5.4B we computed CV across the inter-game interval (IGI) for all users in our data. While the peak of the CV distribution is indeed near 1, the distribution has a tail greater than 1, suggesting that the time interval between games is less regular than a Poisson process for a large portion of users. When we model the time course of people’s gameplay in the next section, we will build on the notion that users’ IGI can be approximated as a Poisson-like process.

We also considered the possibility that physical time influences learning and dropout directly. In order to test this, we replicated previous analyses using IGI as an independent variable. First, we asked if final playing strength differs for users that played the same number of total games but had different average IGIs (Figure 5.4C). This analysis revealed that people with at least 20 and less than 40 games of total experience tend to have higher final Elo ratings the more time passes between games ( $\rho = 0.075, p = 1.691 \cdot 10^{-84}$ ), while people with at least 40 and less than 119 games of total experience show exactly the opposite trend ( $\rho = -0.126, p = 1.109 \cdot 10^{-141}$ ). One potential explanation is that time can play distinct functional roles, where consolidating experience over longer periods of time is critical early in gameplay while a playing strength advantage arises from quicker repetition with more experience. Figure 5.4D shows how users with both low and high average IGIs have higher Elo ratings as more games are played (linear regression:  $\beta =$

$16.306 \pm 0.377$ ). However, the learning curves differ in that users with higher average IGIs start out with higher Elo ratings while those with lower average IGIs have steeper learning curves. Finally, we investigated probability of stopping from one block to the next as a function of current games played and average IGI (Figure 5.4E). Perhaps expectedly, dropout increases monotonically with IGI (logistic regression:  $\beta = 0.020 \pm 0.020 \cdot 10^{-3}$ ). More interestingly, as users play more games their stopping probability increases with higher average IGIs and decreases with lower average IGIs. These results suggest that performance and engagement are not only mediated by playing strength and experience, but also by the time course of games played.

Another important aspect of gameplay in 4-in-a-row is that people are playing against a computer opponent. There is a large body of literature incorporating data from human players to create computer agents that learn opponent models. These models can in turn adapt their strategies to exploit weaknesses or infer hidden information [Billings et al. 1998; Bakkes et al. 2009]. Here, we take an initial step towards understanding human behavior in our task relative to the opponent by analyzing how the playing strength of the AI agent can influence both performance and engagement. Specifically, we hypothesized that relative playing strength, or the difference in computed Elo rating between a player and their opponent, is a reasonable estimate of perceived differences in ability that could impact the shape of learning and dropout functions. We first validated our staircase procedure by affirming that the opponents users are matched with are also improving over time irrespective of the total number of games that people play (linear regression:  $\beta = 22.276 \pm 0.052$ , Figure 5.5A). In Figure 5.5B, we show how the learning curves differ for users with low and high relative Elo ratings. Both groups seem to learn at a comparable rate given that the shape of the curves are similar (linear regression:  $\beta = 23.313 \pm 0.121$ ), but players who tend to be stronger than the opponents they are paired against in turn have higher raw Elo ratings while the inverse is true for weaker players. Then, we examined probability of stopping as a function of relative Elo ratings in a given block (Figure 5.5C) as well as change in relative Elo ratings from one block to another (Figure 5.5D). In the former case, dropout increases as players are much stronger



**Figure 5.5:** The effect of opponent playing strength on learning and dropout. **(A)** Average opponent Elo rating as a function of current user experience level conditioned on total number of games played. This is computed for the set of 107,769 users who played at least 20 and less than 120 total games. **(B)** Average user Elo rating as a function of games played, conditioned on low (less than 0) or high (greater than 0) Elo ratings relative to their opponent. This was computed for the set of 3,088 users who played between 100 and 119 total games, grouped into 10 bins. **(C)** Probability of stopping during the next block of 20 games as a function of current relative Elo rating. Results are grouped into 10 bins, and analyses were conducted for the 115,968 users who had played at least 20 games. Shading here and in (D) denotes s.e.m. **(D)** Same as (C), but for the change in relative Elo rating from one block to the next for the 48,156 users who had played at least 40 games. **(E)** Probability of stopping during the next block of 20 games as a function of both the relative Elo rating and number of games played so far in the current block. This was computed for the set of 104,681 users who played between 20 and 99 total games.

than their opponent and likely find the gameplay demotivating. Meanwhile, in the latter case, dropout is highest when user Elo ratings increase much more or much less than their opponents and lowest when the changes in playing strength are matched. This reinforces the idea that people prefer challenges, namely those that arise when the playing strength of an opponent adapts to changes in their playing strength over time. Figure 5.5E provides a 2-dimensional representation of relative playing strength over number of games, highlighting that as users play more games, they have a lower stopping probability (logistic regression:  $\beta = -0.018 \pm 0.058 \cdot 10^{-3}$ ) and

their stopping probability increases with higher relative Elo ratings (logistic regression: logistic regression:  $\beta = 0.705 \cdot 10^{-3} \pm 0.011 \cdot 10^{-3}$ ). Opponent ratings directly influence game outcomes, which are a core component of the graphical model we derive in the next section.

## 5.2 MODEL

To characterize the dynamic cognitive process that people undergo in terms of their performance in and engagement with 4-in-a-row on a game-to-game basis, we model each players using the formalism of a partially observable Markov decision process (POMDP). A POMDP is a generalization of an MDP where the agent does not know the state, but instead receives an observation conditional on the state and action at each time step [Kaelbling et al. 1998]. POMDP models have typically been used to describe decision-making behavior in which information is gathered from an external environment, for example with eye fixations during visual search [Chen et al. 2017]. More recently, POMDPs have been proposed as a model of internal cognitive operations such as the comparison of payoff probabilities and the computation of noisy estimates of expected value [Chen et al. 2021] and even as a general framework describing interactions between mental states [Oulasvirta et al. 2022]. Thus, we leverage the structure of POMDPs, as our problem similarly consists of interaction with a dynamic and partially observable environment.

The intuition for our computational model is based on our insights into the factors driving learning and dropout functions in this task (Figure 5.6A). We consider a state  $s$  to be a combination of a starting physical time  $t$  of a game  $n$  and a playing strength  $R$ . Since playing strength is an unobservable variable, we assume it can be measured by an Elo rating, which is observable. The outcome of a game  $o$  is determined by the user's rating as well as the opponent rating through the Elo equation. Additionally, we denote the action  $a$  to be the time that a user allows to elapse between the end of a game  $n$  and the start of the next game  $n + 1$ . After each additional game is played, users receive a new Elo rating which noisily increases or decreases based on their

individual learning rate. We also assume that ratings decay exponentially until the user plays another game. Together, these give the following update rule:

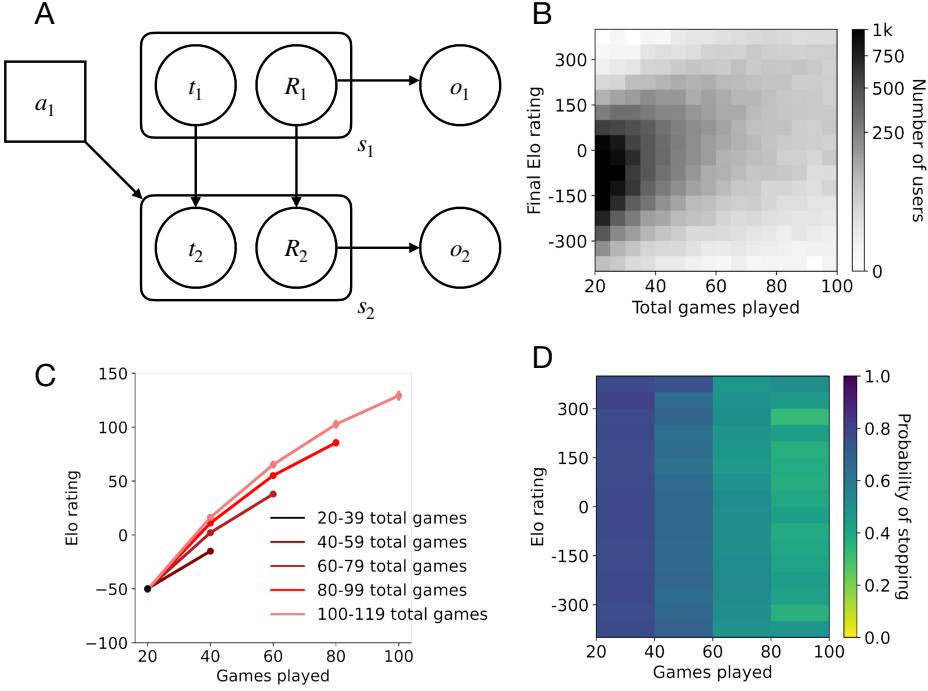
$$R_{n+1} \leftarrow R_n + \lambda_{\text{learning}}(R_\infty - R_n) + \sigma_{\text{noise}}\eta - \lambda_{\text{forgetting}}a_n(R_n - R_0) \quad (5.2)$$

where  $R_0$  is the baseline playing strength,  $R_\infty$  is the asymptotic ceiling playing strength,  $\lambda_{\text{learning}}$  is the learning rate, and  $\lambda_{\text{forgetting}}$  is the forgetting rate. Forgetting decays proportionally to elapsed time. Thus, we explicitly account for the fact that learning starts at a baseline, plateaus over time, and that large gaps between games lead to decreases in playing strength. The noise on playing strength increases  $\eta$  is standard normal noise with variance  $\sigma_{\text{noise}}$ . The update equation for the start time of each game is then:

$$t_{n+1} \leftarrow t_n + a_n. \quad (5.3)$$

We omit the policy  $\pi = p(a_n|s_n)$ , which predicts the elapsed time between games for the user. In practice this can depend on many variables, but for simplicity we assume that it is determined by  $a_n$  and thus implicitly  $R_n$ . The probability distribution we use is for this is a Weibull, which is closely related to a Poisson distribution but is used for near-continuous time between events. Finally, we need a prior on the initial Elo ratings, which we approximate as a normal distribution from the empirically computed ratings at game 20.

Our model has 9 parameters of interest, which are the learning rate and decay  $\lambda_{\text{learning}}$  and  $\lambda_{\text{forgetting}}$ , the baseline playing strength  $R_0$ , the ceiling playing strength  $R_\infty$ , the mean and variance of the initial playing strength prior, the mean and variance of the policy, and the variance of the noise for updating playing strength from game to game  $\sigma_{\text{noise}}$ . For the purposes of this chapter, we run the model forward with manually selected parameters to simulate our empirical results with details provided in Appendix D.3. In future work, we plan to fit the model to individual game results and elapsed times. Figure 5.6B-D show that the correlation between final playing strengths and total number of games played, the population level learning trajectories conditioned on total



**Figure 5.6:** A graphical model of task performance and engagement in 4-in-a-row. **(A)** Partially observable Markov decision process (POMDP) where the state  $s_n$  is the user's state at the start of the game  $n$ . Each state consists of a time  $t_n$  and a playing strength  $R_n$ . Each state results in a game outcome  $o_n$  while  $a_n$  is the time that the users lets elapse between the end of game  $n$  and the start of the next game  $n + 1$ . This panel shows  $n = 1$ , meaning the relationship between game 1 and game 2, for an arbitrary user. **(B)** Model simulations replicating the 2-dimensional histogram of final Elo ratings and total games played from Figure 5.1A. **(C)** Same as (B), but for average user Elo rating as a function of current experience level conditioned on total number of games played from Figure 5.2A. **(D)** Same as (B), but for probability of stopping during the next block of 20 games as a function of both the Elo rating and number of games played so far in the current block from Figure 5.3D.

experience, and the probability of dropout as a function of current playing strength and number of games played can all be reliably reproduced by the model. In Figure D.4, we validate that the model can also replicate the time course of people's gameplay.

### 5.3 DISCUSSION

In this chapter, we utilized the existing large-scale data set of participants playing a two-player combinatorial game in order to characterize human learning and dropout functions. We first es-

tablished that playing strength and overall experience are correlated before using playing strength as an indicator for behavior. Specifically, we found that playing strength increases with the number of games played and that experience level along with current playing strength and changes in playing strength drive stopping probabilities. Then, we expanded the notion of dropout to a continuous spectrum of games played over physical time, and investigated how inter-game intervals bias task performance and engagement. We also considered the effect of opponent playing strength on gameplay. Finally, we combined the components derived from our empirical results into a dynamic model of learning and motivation that is able to reproduce the patterns that were observed in the data.

Taken together, our results must be interpreted within the context of prior work on human skill learning and intrinsic motivation. Learning is typically exponential over time but can be altered by traits such as grit and perseverance. Meanwhile, studies on motivation have found that people prefer to engage with tasks of intermediate difficulty and are particularly influenced by peak and end experiences. Our analyses reveal signatures of each of these factors in human gameplay. Namely, people exhibit playing strength increases with experience that eventually plateau, are less likely to quit the more games they play, and are most likely to drop out with significant positive increases in their own performance or extreme changes in behavior quantified either by their own playing strength or their playing strength relative to their opponent. The key contribution we provide is to embed these aspects of cognition into a process-level model of performance and engagement that views the latter as a decision that occurs dynamically over a continuous period of time. Returning to Figure 5.1B, it seems that 4-in-a-row players lie somewhere towards the top right corner of our didactic spectrum. That is to say, users tend to have a large, albeit noisy, learning rate that mediates how likely they are to drop out. However, people's placement isn't fixed along the diagonal, as, for example, learning rates can decrease over time.

One limitation of our approach is that it is constrained to a singular instance of human gameplay, and it is not immediately clear whether our results generalize to other tasks or real-world

settings. Given that our results are generally consistent with the literature on intrinsic motivation, we suspect that some form of performance-mediated stopping occurs in most tasks for which participation is autonomous, and that people’s exact schedules of engagement vary based on the nature and difficulty of the task. We speculate that the same effects of experience on performance will exist in tasks for which skill learning is required. In particular, even more complex games like chess and Go may take longer to achieve proficiency in due to the necessity for more domain-specific knowledge and practice. That being said, our model is purposefully task agnostic, and should be relatively straightforward to adapt to other tasks if the factors underlying learning and dropout can be reliably identified.

While the work presented here was not explicitly about planning, 4-in-row is designed to elicit behavior where people must think multiple steps into the future. Our foundational learning result is that playing strength increases with the number of games played. In Section 2.3, we showed that a computational cognitive model can predict human moves and therefore attribute improvements in playing strength to increased planning depth, decreased feature dropping, and bounded increase in heuristic quality. These changes in derived metrics point to mechanistic explanations for the cognitive process underlying learning, and in future work we aim to investigate how these metrics relate to dropout. Additionally, we can take the approach from Section 4.2 that utilizes response times as an approximate measure for amount of planning. This would allow us to ask questions about the relationship between specific board features and the cognitive factors we’ve already analyzed across the entire data set rather than a subset that is amenable to model fitting. More broadly, a general model that decides whether or not to continue engaging with a task can be combined with a more task-specific model that actually performs the task itself. While our two models for these aims are currently fairly distinct, bringing together these modular components is a more concrete step forward for understanding how humans cognitively navigate such complex decisions.

One of the main contributions of this work is to advocate for concepts from seemingly dis-

parate fields, such as survival analysis and human studies on motivation, to be integrated into analyses of learning in massive cognitive science data sets. Our use of logistic regression to initially simulate dropout as well as our intuition for factors that mediate learning, such as playing strength as a proxy for task difficulty, are derived from these fields. Another related field worth mentioning is player modeling, or the study of computational models of players in games [Yanakakis et al. 2013]. Specifically, one application of player modeling is to optimize game design to be maximally enjoyable or tailored to the skill level of individual players [Togelius et al. 2006; Pedersen et al. 2010; Shaker et al. 2010]. These implementations either rely on a computational theory of fun or on models that are trained to predict experience or affect [Malone 1981]. Importantly, our modeling is inherently different in use case given that our goal is to mathematically describe cognitive processes rather than successfully engineer engaging content in games. That being said, a better understanding of what people find naturally rewarding in unstructured game-play will surely influence the development human-like player models. As the use of large-scale data sets where participants have autonomy over participation become more ubiquitous, we hope that it will become standard for accounts of human behavior to model learning and motivation simultaneously.

# 6 | CONCLUSION

## 6.1 DISSERTATION SUMMARY

In this dissertation, I outlined a framework for studying the cognitive mechanisms of complex planning. The framework in question consists of a combinatorial game called 4-in-a-row that requires thinking multiple steps into the future, a computational model that combines a feature-based value function and heuristic search to fit human behavior, and a large-scale data set of people playing the game in their daily environments. I showed that these components can be used to attribute increases in playing strength with experience to distinct cognitive mechanisms such as depth of planning and attentional oversights in Chapter 2. This foundation is built upon for the remainder of the dissertation, highlighting the breadth of scientific directions that can be pursued by pushing the boundaries of task, model, and data complexity.

To improve the computational cognitive model, I trained deep neural networks to predict people's moves in 4-in-a-row in Chapter 3. The best network approaches a satisfactory noise ceiling, reproducing signatures of human play almost exactly. The key concept behind this work is to guide model improvement with powerful machine learning techniques. To this end, the best network can be used to identify deviations between the cognitive model and the network, which are more informative than comparisons between that same model and the data. In turn, these residuals inspire three model extensions that range from biases in the early game to the recognition of opponent threats in the late game.

To understand how people might mitigate the costs associated with thinking ahead, I derived a normative model of meta-planning in Chapter 4. The meta-planner conceptualizes planning as gaining noisy measurements about the true value of given state-action pairs, combines this prospective estimate with retrospective information, and returns a decision on whether and in which direction it is worthwhile to plan. I showed how the model produces intuitive simulation results with regards to the gap in perceived value between actions, the accumulation of prior experience, and the uncertainty of simulations, and that evidence for these principles can be found in people's response times in 4-in-a-row.

To investigate the nature of learning and motivation, I analyzed the factors influencing task performance and engagement in Chapter 5. While this work is not primarily about planning, it attempts to characterize the relevant cognitive mechanisms affecting human gameplay in a planning task. I first framed the fundamental problem as one of identifying the underlying processes contributing to an observed correlation between final Elo ratings and total experience in 4-in-a-row. A series of analyses revealed that playing strength is a primary indicator of learning and dropout, and that the physical time that games are played at and opponent playing strength also influence behavior. These results led to the construction of a process-level model of task performance and engagement that reproduces the empirical findings.

In the remainder of this chapter, I conclude by embedding this approach to studying complex planning within a broader context. Specifically, I comment on the successes and limitations of this framework, noting what questions it can be extended to answer, how it relates to adjacent fields in psychology and neuroscience, and what a path forward looks like for enriching our collective understanding of planning in naturalistic environments.

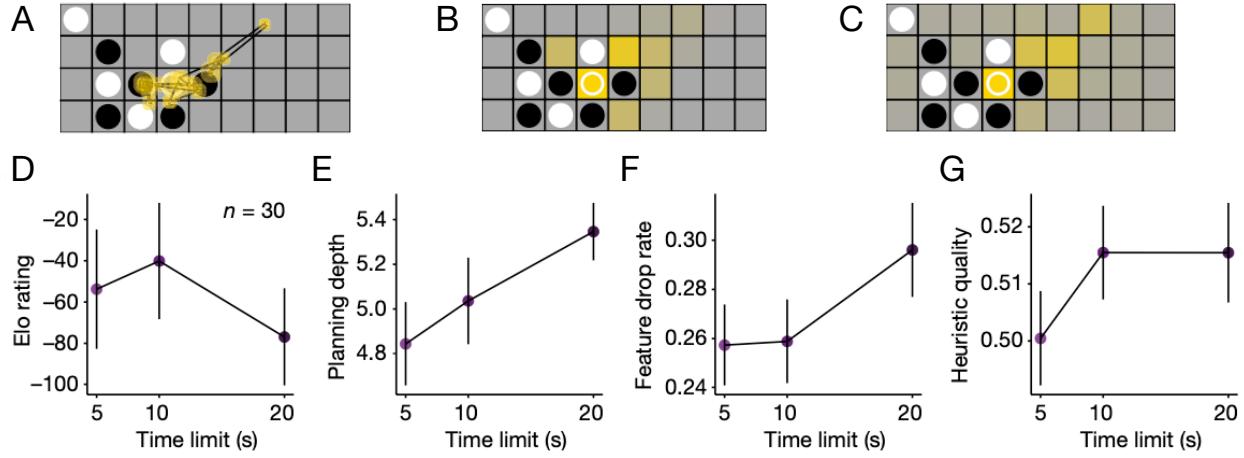
## 6.2 LIMITATIONS AND FUTURE DIRECTIONS

### 6.2.1 EXTENDED USAGE OF THE FRAMEWORK

Given that the proposed framework provides a foundation for studying complex planning, there are numerous extensions along different dimensions of the task, model, and data that I would like to discuss. Many of these are already underway, either in the laboratory or in collaboration with other colleagues, and others will almost certainly be the subject of future work. However, I believe they are directly relevant to this dissertation, especially in highlighting how an overarching research program can be constructed around this framework.

Empirical variations on 4-in-a-row were a fundamental part of the work in Chapter 2 [van Opheusden et al. 2023]. There, I focused on the methods and modeling that were essential to the remaining chapters, noting the primary expertise result that replicates from the laboratory experiment to large-scale mobile data. Two of these experimental manipulations that I have thus far neglected to mention served to validate the computational cognitive model of human planning. In a generalization experiment, 40 participants performed three tasks: playing against computer opponents, a two-alternative forced choice (2AFC) between moves in a given position, and a board evaluation task. The computational model predicted people's choices above chance, suggesting that it can generalize between different choice tasks in the 4-in-a-row domain. Moreover, we conducted a Turing test experiment [Turing 2009], in which 30 observers decided whether sequences of moves were generated by the model or by human players. Human observers achieved only 55.4% discrimination accuracy, suggesting that the model makes human-like decisions. One potential use of the neural networks in Chapter 3 is to run a similar Turing test experiment, where the hypothesis would be that people's discrimination accuracy is even lower than that of the model if it is truly capturing people's behavior.

Beyond model validation, we performed multiple experiments that could serve as starting



**Figure 6.1:** Empirical extensions of 4-in-a-row, namely eye tracking (top) and time pressure (bottom). **(A)** The trajectory of eye movements on one example trial. The black lines represent saccades and the yellow circles represent fixations. The circle area indicates the duration of fixation. **(B)** The estimated distribution of overt attention across unoccupied squares, obtained by convolving the eye trajectory with a Gaussian filter. **(C)** The distribution of squares visited by the model’s search algorithm, with parameters estimated from the participant’s choices. **(D)** The average Elo rating of the participants in the time pressure experiment, as a function of the time limit. **(E)** The average depth to which participants plan in the time pressure experiment, as estimated by the behavioral model. **(F)** The same as in (E), but for the feature drop rate. **(G)** The same as in (E), but for heuristic quality.

points for entire lines of research. To analyze eye movements, we ran an experiment in which 10 participants played against computer opponents while we tracked their eye movements with an infrared video-based eye tracker. Figure 6.1A shows one participant’s fixation trajectory in an example board position. We estimated the distribution of squares that a participant overtly attends to on an individual trial by convolving their fixation trajectory with a Gaussian filter, truncating to unoccupied squares and averaging in time. Figure 6.1B-C shows that the distribution of squares visited by the cognitive model during its search process resembles this distribution of attention (mean correlation across participants:  $\rho = 0.535 \pm 0.024, t(9) = 21, p < 0.001$ ). This provides a preliminary relationship between the model and eye tracking data, but does not provide answers to more detailed questions. For example, can eye and mouse movements more generally be used as a predictive measure of people’s internal cognitive processes, serving as an indicator for other internal states such as surprise? Further, we conducted a time pressure

experiment in which 30 participants played against computer opponents, with a time limit of 5, 10, or 20 seconds per move, randomly sampled for each game. We predicted that, if planning depth approximately measures the amount of computations that a participant performs while making a move, it should scale with time used for that move [Calderwood et al. 1988; Chabris and Hearst 2003; Van Harreveld et al. 2007]. Figure 6.1D shows that planning depth is overall lower than in the learning experiment and indeed increases with longer time limits ( $\beta = 0.042 \pm 0.018, p = 0.019$ ). Despite this increase, we find no improvement in participants' playing strength ( $\beta = -2.0 \pm 1.6, p = 0.21$ , Figure 6.1E). The model suggests a potential explanation for the lack of performance: at the most relaxed time limit, people overlook features more often ( $\beta = 0.0027 \pm 0.0010, p = 0.009$ , Figure 6.1F), and the dropped features cancel out the benefit of increased search. In this experiment, heuristic quality does not change with time pressure ( $\beta = 0.00086 \pm 0.00056, p = 0.13$ , Figure 6.1G). However, the interplay between time pressure and attention in the context of planning is itself an interesting question that can be studied with experiments that alter the specifics of the time limit conditions as well as the order in which they are shown.

Additionally, we ran a memory and reconstruction experiment with the 30 participants in the laboratory version of the learning manipulation. This showed that experts were better at reconstructing the specific set of features that our model relies on for its evaluations. Parallel work has replicated this result for memory and reconstruction of game sequences rather than individual positions [Huang et al. 2023]. These results suggest an explanation for the observed effect of expertise on planning depth: experts sharpen their representation of game-relevant features, allowing for more position evaluations per unit time and therefore deeper planning. A number of related extensions that change the task itself from free gameplay are worth considering. For example, does giving people a tryout board to augment their planning improve performance or alter planning depth? Moreover, how predictable are the board positions in which participants opt for physical over mental simulation? Taking inspiration from the chess literature, would behavior change if people were asked to search for a winning sequence of moves in 4-in-a-row

puzzles that contain minimum solutions of varying lengths or narrate their deliberation process out loud while selecting moves?

One specific application of the task and model worth highlighting is in the realm of developmental science [Ma et al. 2022a]. Participants of ages 8 to 25 years played 4-in-a-row against the same set of computer opponents used in the large-scale mobile experiment, and the behavioral model was fit to their moves. The study found asynchronous age-related changes in the model’s derived metrics, namely that heuristic evaluation of features improves rapidly from childhood to adolescence, planning depth shows more protracted developmental improvements, and attentional oversights do not vary across age groups. Collectively, these results provide a mechanistic account of the continued development of model-based decision strategies into adulthood. Moreover, this work demonstrates that the framework can be used to disentangle how multiple, distinct cognitive processes contribute to planning.

In terms of models, I have introduced three distinct classes of models throughout this dissertation: process-level models of human behavior, either for planning in Chapter 2 or performance and engagement in Chapter 5, neural networks to predict human play in Chapter 3, and a normative framework for meta-planning in Chapter 4. Naturally these models interact in various contexts, such as the neural network guiding the implementation of improvements in the heuristic search model and the meta-planner deriving principles that explain previously unaccounted for data in 4-in-a-row. However, there are interactions which have not yet been explored. For example, we can incorporate a retrospective algorithm into the behavioral model for 4-in-a-row, or perhaps more ambitiously implement a meta-planner to guide search online. Alternatively, we can hypothesize about how learning and motivation interact with planning to model gameplay at timescales beyond predicting individual moves. We also have ongoing work training LLMs to predict human moves. In contrast to the aforementioned neural networks, LLMs have a capacity to be even more powerful move predictors that can capture novel aspects of human behavior such as individual differences between participants and the effects of sequences of moves. Addi-

tionally, there is currently broad interest in comparing how humans and machines learn different cognitive tasks [Flesch et al. 2021; Kumar et al. 2022]. While networks trained to predict human play are not suitable for this approach, neural networks optimized for the task itself, similar to AlphaZero, suggest that increases in performance are mostly driven by policy quality [Zheng et al. 2022]. This type of comparison can highlight differences between the strategies that humans and artificial agents use to solve planning problems, and often engineering computer agents to exhibit more human-like behavior provides insight into our underlying cognitive function.

Finally, I have yet to mention the third component of the framework on which this dissertation is based, which is the large-scale behavioral data set. The main advantage of collecting rich behavioral data is that it can be used as a testbed for a range of psychological theories. The approach outlined in Chapter 5, which is to leverage the size of the data to answer questions about cognitive mechanisms of interest that are not necessarily related to planning, can be extended to many other domains. Perhaps the most straightforward extension is to investigate aspects of human gameplay that have already been assessed in massive chess data sets, such as blunders, risk-taking, and the selective use of cognitive resources [Anderson et al. 2017; Holdaway and Vul 2021; Russek et al. 2022]. In addition to these, questions ranging from the representations that people use while planning to identifying if distinctions in play arise in highly experienced players are all viable research directions. The results in this dissertation are based on a data collection period of roughly 8 months, but users have continued to play 4-in-a-row on the Peak platform since then. At the time of writing, this means that we have access to an additional 57 months of data. In practical terms, none of the work here would have benefited from an order of magnitude increase in the number of games. However, there may be future ideas that necessitate scaling the size of the data set respectively. Preliminary thoughts in this direction include testing methodological advancements that allow for an increased capacity to fit process-level models, limiting analyses to particular pools of participants or board configurations, and investigating how strategies develop across the population of players over years.

### 6.2.2 RELATIONSHIP TO NEUROSCIENCE

I am compelled to comment on the fact that this dissertation will be submitted in a neuroscience department despite being completely devoid of what is traditionally thought of as neural data. Indeed, the work I have presented concerns itself primarily with explaining human cognition via behavioral modeling, and contains no data collected using methods such as animal electrophysiology or human neuroimaging. This primarily reflects a change in my personal interests as a scientist, but also in my perception of how neuroscience and cognitive science are related. Perhaps the most obvious connection is that these two fields constrain each other: knowledge about biological plausibility provides bounds on the computations or algorithms that the brain can possibly implement, and behavioral modeling informs the set of signals or response characteristics to search for in neural data. However, in my experience, these are softer constraints than many people tend to initially believe, and there is value in exploring the space of algorithms that could explain a particular behavior without explicitly considering its neural implementation *a priori*. All of the process-level and normative models presented in this dissertation were constructed using this mindset, with consideration given to a rational use of cognitive resources with mechanisms such as pruning in the heuristic search model or explicitly aiming to reduce the number of costly measurements made in the meta-planner. The majority of models that cognitive scientists propose are of this form, and I do not believe that they are fundamentally incompatible with our understanding of neural hardware. In fact, as I covered in the Section 1.2, the study of multi-step planning has an interdisciplinary history with roots in both psychology and neuroscience that my work hopefully contributes to. Ultimately, both neuroscience and cognitive science aim to collectively explain how the human mind functions, and it is logical to approach such a massive question from different levels of abstraction. Simply put, human behavior is, in my opinion, at its core a form of neural data.

That being said, neural data can serve another function in relation to cognitive science not

unlike how, for example, response time data has been used throughout this dissertation to lend credibility to specific model components. This can be achieved by correlating these components directly with neural measurements, and there are various avenues of work doing so with 4-in-a-row and its respective behavioral model in order to better understanding possible neural implementations of model-based planning algorithms. Nathaniel Daw and Marcelo Mattar are leading a study in which participants perform trials in which they select the best move while in a functional magnetic resonance imaging (fMRI) scanner. Based on literature from neuroscience [Johnson and Redish 2007; Pfeiffer and Foster 2013] and behavioral economics [Levy and Glimcher 2012], the hypothesis is that the hippocampus and ventromedial prefrontal cortex (vmPFC) encode values of board states, which are either those presented currently to the participant or ones suggested by the search algorithm as possible future states which are likely to arise. Another possible question is whether vmPFC computes a weighted sum of features like the heuristic value function in the model, which can be investigated using a representational similarity analysis [Kriegeskorte et al. 2008]. Additionally, Daeyeol Lee is implementing a version of 4-in-a-row for nonhuman primates to test simultaneously record single-neuron and local-field potential (LFP) activity of neurons in the prefrontal cortex. The goal is to identify whether dorsomedial prefrontal cortex (dmPFC) contributes to the evaluation of alternative strategies and learning algorithms [Seo et al. 2014; Lee and Seo 2016] while dorsomedial prefrontal cortex (dlPFC) might be more involved in the computation and representation of integrated values of alternative choices [Lee et al. 2014]. Sanjay Manohar is also leveraging the task to investigate whether human patients with prefrontal cortex lesions show deficits in planning ability that don't seem to be present in the two-step task. Together, neural data on 4-in-a-row across both humans and nonhuman primates will serve to illuminate the role that various brain regions play in sequential decision-making.

### 6.2.3 GENERALITY AND SCALING FURTHER

Undoubtedly the most glaring limitation of this dissertation is that all of the findings are constrained to a single task and data set. This begs the question: is 4-in-a-row actually representative of complex planning in the real world? In terms of the results themselves, there is no reason to suspect they would not generalize to other paradigms and environments. The primary conclusion about the increase of planning depth with expertise is consistent with classic results in chess [Campitelli and Gobet 2004; Holding 2021], with the understanding that those results were obtained using qualitative methods. Our neural network approach to model improvement is a direct application of prior work to the domain of planning [Pedersen et al. 2021], and our meta-planner is closely related to other classes of models ranging from MCTS to information sampling [Browne et al. 2012; Callaway et al. 2021], deriving principles for thinking ahead that can be found in various studies on human planning [Dickinson 1985; Daw et al. 2005]. Similarly, our work on task performance and engagement is directly linked to the psychology of skill learning and intrinsic motivation [Heathcote et al. 2000; Schmidhuber 2010]. Thus, each chapter in this dissertation is supported by the broader literature in such a way that the choice of paradigm is not a limiting factor in and of itself.

However, I am ultimately interested in attaining an in-depth understanding of the general principles underlying human cognition. As such, it is important that what we learn about planning in my work holds across a wide range of paradigms. Before considering generalizations to other paradigms, though, it was crucial to develop a pipeline that allows for reliable, quantitative inferences about human reasoning in naturalistic environments. Due to the fact that complex planning is a relatively new field, such a pipeline was entirely missing. Thus, the approach outlined in this dissertation lays the methodological foundation for a new subfield of the study of higher cognitive function that can be readily applied to other paradigms. One example of this includes work modeling human decisions in a single-player puzzle called Rush Hour, which con-

sists of a dense configuration of rectangular cars on a 6-by-6 grid where the goal is to move cars horizontally or vertically to allow the target car to reach an exit. To directly address the concept of generalization, another line of ongoing research is running 4-in-a-row as part of a larger battery of widely used cognitive tasks that includes Corsi block-tapping [Corsi 1972], Towers of London [Shallice 1982], and mental rotation [Shepard and Metzler 1971] tasks as well as alternative planning tasks. This will elucidate to what extent the components of planning map onto more established cognitive functions via correlations, and can potentially be used to investigate if planning abilities transfer across tasks.

What about Chase and Simon’s claim that chess has the potential to become the “Drosophila of psychology,” establishing itself as a standard task for investigating theories of cognition? A central assertion made throughout this dissertation is that there are numerous benefits to using intermediate complexity tasks that preserve tractability. Until now, chess has not been amenable to the type of process-level modeling presented here. However, recent work has provided a proof of concept that it is indeed possible to test hypotheses regarding human cognition from a quantitative perspective in chess despite the innate complexities of the task [Russek et al. 2022]. Similarly to the work presented here, the approach is to leverage the accessibility of massive numbers of human games from online platforms, albeit to characterize when people make intelligent decisions about how to allocate their cognitive resources. The lack of a computational model for human play in chess is alleviated by the use of chess engines as an idealized model of planning. Pushing the boundaries of task complexity is a natural step forward for the work outlined in this dissertation, and this initial illustration provides a roadmap for validating results such as the kinds of simplified representations that people have, whether people utilize computations from previous experiences, and if people selectively ignore unpromising action sequences. Scaling this further has the potential to uncover the learning processes that govern people’s decisions, either via reinforcement learning at the individual level or cultural transmission at the collective level. The highest form of success would be to develop a detailed computational model of human plan-

ning in chess similar to the one for 4-in-a-row. A major difficulty is in identifying the features that comprise a value function, but there is promising work analyzing the knowledge that AlphaZero acquires in chess that may serve as a starting point [McGrath et al. 2022]. This overarching approach is not at all limited to chess, and can extend to other complex games such as Go [Shin et al. 2023]. As such, I am optimistic that, over 50 years later, computational cognitive scientists might finally be poised to contribute to realizing Chase and Simon’s vision.

#### 6.2.4 RELATED FIELDS

Planning in naturalistic environments doesn’t occur in a vacuum. Thus, there are many lines of research within computational cognitive science that have the potential to interact with complex planning in interesting ways. If the ultimate goal for researchers working on planning is to create a universal behavioral model of prospection, then it is imperative to start considering such interactions. I briefly review related fields here, commenting on points of intersection.

One prominent domain is that of social cognition, which investigates how people process information and make decisions in contexts that require interaction with others. Theory of mind has traditionally framed this problem as one of optimizing mutual behavior by modeling representations of a second agent’s intentions and goals [Yoshida et al. 2008]. Related work has defined resource rationality in joint action spaces [Török et al. 2019; Vélez et al. 2023], the effect of priors and uncertainty over a partner’s policy and beliefs [Barnby et al. 2022; Ma et al. 2022b], and the differences in executing programs that selectively collaborate or compete [Kleiman-Weiner et al. 2016]. Recently, it has been suggested that theory of mind is used to plan novel interventions and predict their effects [Ho et al. 2022b], but the specific ways in which people infer and incorporate the knowledge and strategies of others while thinking many steps ahead remains an open question. Does planning depth change as uncertainty over a collaborator’s policy is reduced, and what is the role of communication when making collaborative decisions? Teaching is a particular instance of collaboration in which a teacher forms a belief about the knowledge

that a student possesses and gives instructions or explanations that try to induce a target concept in the student's mind [Popp and Gureckis 2020]. Describing the cognitive mechanisms underlying people's ability to transmit abstract, generalizable concepts efficiently through a handful of examples is an active area of research [Gweon 2021; Vélez et al. 2023], and there is work applying similar concepts to improve people's planning strategies via artificial intelligence [Callaway et al. 2022a]. However, the exact components of prospective deliberation that can be transferred from one agent to another, and the methods by which this process is achieved, require further investigation. At a population level, social cognition enables cultural transmission, or the spread of ideas and strategies through large numbers of individuals [Thompson et al. 2022]. This type of social interaction certainly occurs in adversarial games such as 4-in-a-row, chess, and Go, where players are paired with different opponents and have the opportunity to learn and improve by observing their actions. As such, I believe that the interaction between planning and social cognition will steadily emerge as an exciting research area, specifically along dimensions such as collaboration and competition, teaching, and cultural transmission in large populations.

There are also a collection of current ideas in cognitive science that can be incorporated into and provide constraints on the construction of algorithms for human planning. One such example is working memory, which refers to an information-limited process used to hold representations in the mind temporarily for use in thought and action [Cowan 2017; Oberauer et al. 2018]. The literature on working memory is much too broad for me to cover, but there has already been work advocating for the serious treatment of working memory in reinforcement learning agents that aim to replicate humans' ability to learn, generalize, and make flexible decisions [Yoo and Collins 2022]. During tree search, what is the optimal strategy under which to expand nodes if people have limited memory capacity rather than a simple global cost per computation? Further, if the process of planning includes forgetting the values of certain states over time, how should an algorithm be designed to revisit states and update beliefs accordingly? Goals are another factor that play a central role in driving human cognition, but the majority of models of learning and

decision-making take goals as given [Molinaro and Collins 2023]. Investigating goal selection as an independent value-based decision process is essential to develop more complete accounts of behavior in both biological and artificial agents. Finally, the interaction between various aspects of cognition requires people to intentionally exert effort to overcome automatic biases such as habits. The field of cognitive control aims to uncover how cost-benefit analyses are impacted by innate processes and motivational barriers [Frömer et al. 2021; Collins and Shenhav 2022]. While these are just a few examples, any cognitive process that impacts decision-making should also be studied in the context of planning, since by definition planning extends decision-making to a sequential problem that is simulated internally. In sum, directly studying the interplay between disparate processes is particularly important when aiming to build a comprehensive understanding of human cognition. Planning is no exception to that, and thus far computational theories of planning tend to ignore factors such as rewards, goal selection, and cognitive control that undoubtedly affect behavior.

### 6.3 PARTING WORDS

Throughout this dissertation, I have hopefully convincingly advocated for an approach to science that first conceives of a complex task and then uses rigorous computational methods to extract meaning from data generated by behavior in that single task instance. This might give the impression that I am fundamentally against a more traditional approach to research, or at the very least that I consider my way to be superior. This could not be further from the truth. In fact, I have found myself quite often fascinated by talks I have attended or papers I have read in which the elegance of a task designed to cleverly test a specific hypothesis instantly struck me. Perhaps this is a case of the grass always being greener on the other side, where my lack of training in experimental design becomes all the more apparent upon self reflection. Nonetheless, I am equally drawn to the inverse approach to science than what I have presented here, which is primarily

motivated by a question and then considers how to best isolate a variable of interest.

This is all to say that there is no correct or incorrect way to do science. More complex tasks are not better than simpler tasks, nor can they by themselves inherently elicit more interesting behavior or give us more insight about the human mind. The same can be said for specific computational methods, and for every other choice we make as scientists. In the end, the approach I've taken in my research reflects some combination of what appeals to me in a logical sense, but also the circumstances that I've been placed in that are mostly out of my control. This is something that I strongly believe in, and throughout my time in graduate school I've always felt uncomfortable by people dismissing certain research directions because they differed from their own. To create a truly inclusive academic environment, it is imperative that we support and show interest in a multitude of diverse variations on the scientific method.

This level of open-mindedness, in one way or another, relates to how I ended up studying complex planning in the first place. I would love to say that it was the result of deep thought in which I carefully considered a number of the possible benefits and drawbacks to pursuing this line of work compared to alternatives. Somewhat ironically, it had nothing to do with planning at all, and I really just thought working on games might be fun when presented with the idea as someone who decidedly lacked a singular topic I wanted to dedicate years of research to. This nicely connects back to the quote at the beginning of this dissertation as well as one of Jorge Luis Borges's most famous stories, *The Garden of Forking Paths*, which is known for its description of an infinite labyrinth that branches in time rather than space. Tomihiko Morimi's *The Tatami Galaxy* operates in this exact environment, with the protagonist repeatedly making a slightly different choice at the outset of his college years, none of which lead to his desired, idealized campus life. Ultimately, what matters is the acceptance of past decisions and moving forward, with the knowledge that exploring another branch of the labyrinth would not necessarily lead to a better outcome, whatever that may mean. Similarly to the novel's protagonist, I did not intend to study complex planning, but I can't imagine what my life would look like if I had

chosen a different path, and trying to do so is not a particularly fruitful exercise. Maybe the right interpretation here is that planning is overrated.

To conclude, I think it is an extremely exciting time to be a computational cognitive scientist. Our methodological toolkit continues to expand as a field, eliciting newfound interactions between seemingly disparate processes underlying the function of the human mind. I hope that with the work outlined in this dissertation, I have played a small role in contributing to our collective understanding of human decision-making and cognition more broadly. And in the future, maybe I'll finally get to design an experiment.

# A | APPENDIX FOR CHAPTER 2

## A.1 HUMAN-VERSUS-HUMAN EXPERIMENT

For our human-versus-human experiment, we recruited 40 participants in pairs. For each pair, we provided consent forms and instructed participants on the task together, after which we separated them into different rooms from which they played games against each other through an online interface. After 50 minutes had expired and they finished their last game, the participants completed a post-task questionnaire, during which we provided them with compensation (12 USD in cash). Only after completing the survey and receiving compensation did the participants leave their respective rooms. Thus, the participants interacted socially before and after the experiment, but not during games.

The participants played games against each other, switching colors every game. After each game, we presented both participants with a pop-up showing both players' names, the current score, and a button to continue to the next game. The interface proceeded only after both players had clicked the "continue" button. Every time the participant or their opponent moved, the interface made a faint clicking noise. During games, instead of making a move, participants could offer a draw to their opponent, which caused a pop-up prompt to appear on the other participant's screen to accept or reject the offer. If the opponent accepted the draw, the game ended immediately, otherwise the pop-up disappeared and the player who made the offer could make a move instead. We did not restrict how many draw offers participants could make (including multiple

offers on the same move), but participants made relatively few draw offers. In this experiment, we never imposed any time limits.

## A.2 LARGE-SCALE MOBILE DATA

In collaboration with the mobile app company Peak (<https://www.peak.net>), we designed a large-scale study of people playing 4-in-a-row. When signing up for the app, users consented to a privacy policy, which included a provision that aggregated and anonymized data might be shared with third parties such as universities. The Institutional Review Board of New York University determined that no further consent was required and approved the research protocol as exempt.

Overall, we collected 11, 529, 163 games where users always play first and the game board itself is vertically oriented and gamified. Users play at-will against a computer opponent. We filtered the data to remove games where the user times out of any move, since timing out creates games where the computer opponent plays first or makes two consecutive moves. The time limit to make a move is 10 seconds. This procedure resulted in 82, 761, 594 moves from 10, 874, 547 games and 1, 234, 844 unique users. In Table A.1, we provide the number of users that played 4-in-a-row in each country on the Peak platform. In order to generate the computer opponents, we made slight modifications to the cognitive model described in Chapter 2 and in greater detail later on in this supplement: (1) we used a pruning rule that keeps only the  $K$  highest-value children in each node of the search tree, (2) we added a scaling constant that multiplies the weights of features belonging to the opponent and for features of different orientation, and (3) we artificially added a delay to each computer move to simulate thinking times, which monotonically increased with the number of search iterations that the computer performed on each move. This ensured that the computer played faster in easy positions compared to hard ones. We created 7 classes of computer opponents of varying strength by specifying distinct parameter ranges derived from the human-versus-human experiment that corresponded to estimated Elo ratings [Elo 1978], and

Country code	Number of users
US	556,838
GB	254,638
FR	129,728
BR	120,999
DE	116,603
JP	96,485
IT	75,274
CA	72,094
AU	63,809
MX	52,982
NL	41,712
ES	39,961
CN	38,646
IN	26,624
TW	23,225
CH	21,132
DK	20,869
BE	20,775
SE	20,757
AR	19,297

**Table A.1:** Number of users per country that play 4-in-a-row on the Peak platform. Only the top 20 countries with the most users are shown.

matched users with an opponent on each game based on their track record of game results.

In terms of the implementation of the experiment on the Peak platform, each turn is scored based on the time remaining for each move. Specifically, there is a base score per move and a multiplier that is added depending on how long the user takes:

$$\text{perTurnScore} = \text{baseScore} + \lfloor (\text{multiplier} \cdot \sqrt{\text{timeRemaining}}) \rfloor. \quad (\text{A.1})$$

At the end of each game, a bonus is added based on the result. This is the sum of all scored turns multiplied by a value that is based on losing, drawing, or the number of turns it took to win:

$$\text{endBonus} = \lfloor \left( \sum \text{perTurnScore} \cdot \text{multiplier} \right) \rfloor. \quad (\text{A.2})$$

Rank	Base Score	Multiplier
1	50	10
2	100	10
3	150	10
4	250	10
5	350	10
6	450	10
7	550	10

**Table A.2:** Per turn scoring for 4-in-a-row on the Peak platform. Each rank corresponds to a class of users that are matched with AI agents. These users receive a combination of a base score and multiplier to compute a score on each turn.

Thus, the final score that the user receives is:

$$\text{finalScore} = \sum \text{perTurnScore} + \text{endBonus}. \quad (\text{A.3})$$

In Tables A.2 and A.3, we provide the full per turn and bonus scoring. Each user is assigned a rank based on their score, which corresponds to the 7 AI agent classes. For each turn, an AI is randomly chosen from this range to decide which computer opponent makes the move. This happens during each move, so multiple AIs are used in a single game to match other task AIs on the Peak platform. Overall, the values are chosen so that the theoretical maximum score at each rank increases between ranks and follows a fair downward progression as the games plays out.

Incidentally, 4-in-a-row is one of the most popular games on the Peak platform. In a qualitative survey of 520 randomly selected users, participants were asked to report what they thought about the task. We provide a selection of the responses below that illustrate aspects we considered in designing the task or that make 4-in-a-row particularly suited to studying complex planning.

*What do you like about the task?*

- Another way of developing my problem solving skills, different from the other tasks. It is challenging.

Result	Number of moves remaining	Multiplier
Loss	Any	0
Draw	Any	1.5
Win	4	28
Win	5	22
Win	6	18
Win	7	15
Win	8	12
Win	9	10
Win	10	8
Win	11	7
Win	12	6
Win	13	5
Win	14	4
Win	15	3
Win	16+	2

**Table A.3:** End bonus scoring for 4-in-a-row on the Peak platform. After every game played, users receive a bonus score based on a combination of result (win, loss, or draw), number of moves remaining in the game, and a multiplier.

- Having an opponent makes the game different from other games and makes it interesting.
- It looks easy but it turns out to be quite challenging.
- It's a more strategic game of the classic "tic tac toe" hence combining classic childhood games whilst exercising the mind in a more complex matter.
- It's challenging but so satisfying when you beat the AI.
- It's fun to be thinking ahead.
- Keeps you thinking about your next move in order to win the game whilst also thinking how to not let the computer win.
- Makes me try to look ahead for moves. It seems though that the computer lets me win sometimes by placing marker in totally useless position.

- More advanced noughts and crosses. First to start (me) always has the advantage.
- The computer usually will win if you leave it the opportunity, which does make it feel more challenging to you.

*What do you dislike about the task?*

- Getting a diagonal line is incredibly difficult.
- I always have to make the first move.
- I can't believe that the computer doesn't beat me every time.
- I haven't yet worked out how to improve at this game. It feels a bit more dependant on luck than skill.
- It takes a while for the game to choose a move sometimes.
- It's hard to stay consistent and win a lot.
- Not dislike exactly and I understand that the Peak player has to be fair but sometimes its moves seem illogical.
- Playing to draw (deliberately not playing winning moves) to move up ranks.
- Presumably there is a formula for beating the robot, so once you have worked it out, it will no longer be challenging.
- You have to trust that the computer is using a process similar to a human rather than making the most obvious move.

## A.3 DETAILED MODEL SPECIFICATION

### A.3.1 VALUE FUNCTION

The value function consists of two terms, the first of which measures whose pieces are closer to the board center:

$$V_{\text{center}} = \sum_{\vec{x} \in \text{Pieces}(s, \text{black})} \frac{1}{\|\vec{x} - \vec{x}_{\text{center}}\|} - \sum_{\vec{x} \in \text{Pieces}(s, \text{white})} \frac{1}{\|\vec{x} - \vec{x}_{\text{center}}\|} \quad (\text{A.4})$$

where  $\text{Pieces}(s, p)$  enumerates the locations of all pieces that player  $p$  owns,  $\vec{x}_{\text{center}}$  denotes the coordinate of the board center, and  $\|\cdot\|$  is the Euclidean distance.

The second term counts how often particular patterns occur on the board in any orientation. A feature is a binary function  $f_{t,x,y,o}(s)$  that returns 1 if a pattern of type  $t$  occurs at location  $(x, y)$  with orientation  $o$ , and 0 otherwise. We use the following patterns: connected 2-in-a-row, unconnected 2-in-a-row, 3-in-a-row, and 4-in-a-row. We define  $F$  to be the set of all such features (one for each type, orientation, and board location), and associate a weight  $w$  to each feature in this set. The feature weight depends only on its type, not the orientation or location. Finally, we write the value function as:

$$V_F(s) = w_{\text{center}} V_{\text{center}}(s) + c_{\text{black}} \sum_{i \in F} w_i f_i(s, \text{black}) - c_{\text{white}} \sum_{i \in F} w_i f_i(s, \text{white}) + \mathcal{N}(0, 1) \quad (\text{A.5})$$

where  $c_{\text{black}} = C$  and  $c_{\text{white}} = 1$  whenever black is to move in state  $s$ , and  $c_{\text{black}} = 1$  and  $c_{\text{white}} = C$  when it is white's move. The final term  $\mathcal{N}(0, 1)$  represents additive Gaussian noise with mean zero and unit variance.

### A.3.2 TREE SEARCH

The search algorithm constructs a decision tree, consisting of nodes that contain a state  $s$ , the color of the active player in that state, and a value associated to the state. Upon initialization, the value of a new node is set by calling the feature-based evaluation function. However, this value changes as the algorithm investigates the consequences of future play from that state. The algorithm starts with a single node and gradually grows the decision tree. Each iteration, the algorithm selects a leaf node, expands it by adding one child node each for a number of candidate moves, and backpropagates the value of these new nodes recursively into the leaf node as well as its parents.

---

**Algorithm 1:** MakeMove(state  $s$ )

---

```

if Lapse( $\lambda$ ) then
    return RandomMove( $s$ );
else
    DropFeatures( $\delta$ );
    root  $\leftarrow$  node( $s$ );
    while !Stop( $\gamma$ ) and !Determined(root) do
         $n \leftarrow$  SelectNode();
        ExpandNode( $n$ );
        Backpropagate( $n$ );
    end
end
return argmax $_{c \in \text{children}(\text{root})} c.\text{val};$ 

```

---

Here, Lapse and Stop represent stochastic functions that return true with probability  $\lambda$  and  $\gamma$ , respectively, and Determined checks if the value of the root node (winning, losing, or drawn)

has been determined with certainty.  $\text{RandomMove}(s)$  returns a random legal move in state  $s$ . The  $\text{SelectNode}$  function determines the order by which nodes are added to the tree. We use best-first search, which selects a node by following the principal variation, in which both players always make the best moves according to the currently estimated values, starting from the root to a leaf node. Because the value of nodes in the tree change after each iteration, so does the principal variation, and therefore the search algorithm dynamically switches between different branches of the tree.

---

**Algorithm 2:**  $\text{SelectNode}()$ 


---

```

 $n \leftarrow \text{root};$ 
while  $\text{children}(n) \neq \emptyset$  do
    if  $n.\text{color} = \text{black}$  then
         $| n = \arg \max_{c \in \text{children}(n)} c.\text{val}$ 
    else
         $| n = \arg \min_{c \in \text{children}(n)} c.\text{val}$ 
return  $n;$ 

```

---

After the search algorithm has selected a leaf node to explore, it expands it by adding one child node for each legal move in the associated state. As it initializes the children, it automatically evaluates their states using  $V(s)$  as defined above. The algorithm does not yet check whether either of these states is terminal (that is, either player has achieved 4-in-a-row or the board is full), but it effectively does so if  $w_{\text{4-in-a-row}}$  is high enough. Next, the algorithm prunes unpromising children whose value difference with the best candidate move exceeds a threshold  $\theta$ . Only afterwards does it assign  $V = 10,000$  to each child state in which black has won,  $V = -10,000$  if white has won, and  $V = 0$  for draws. It is therefore possible that, if  $w_{\text{4-in-a-row}}$  is too low, or if the algorithm has dropped a 4-in-a-row feature in a relevant location, it will prune away an immediately winning move, which can result in bad (but human-like) blunders.

---

**Algorithm 3:** ExpandNode(node  $n$ )

---

```
s ← n.state;  
foreach legal move  $m$  in  $s$  do  
    | n.AddChild(node( $s + m$ ));  
  
if  $n.color = black$  then  
    |  $V_{max} = \max_{c \in children(n)} c.val$   
  
else  
    |  $V_{max} = \min_{c \in children(n)} c.val$   
  
for  $c \in children(n)$  do  
    | if  $|c.val - V_{max}| > \theta$  then  
        | | RemoveChild( $c$ )
```

---

Next, the search algorithm incorporates the value of the newly created nodes into the decision tree with minimax backpropagation. After backpropagation, the value of each state reflects the search algorithm's best estimate of the result of a game starting in that state with perfect play from both sides.

---

**Algorithm 4:** Backpropagate(node  $n$ )

---

```
if  $n.color = black$  then  
    |  $n.val \leftarrow \max_{c \in children(n)} c.val$   
  
else  
    |  $n.val \leftarrow \min_{c \in children(n)} c.val$   
  
if  $n \neq root$  then  
    | | Backpropagate ( $n.parent$ )
```

---

The search algorithm continues to run until the Stop routine returns true, after which it makes the best move according to its estimated values. Since the Stop routine is random and

independently drawn each iteration, the total number of iterations follows a geometric distribution with parameter  $\gamma$ . When implementing our model as a computer algorithm to play against human opponents, we convert the number of iterations  $N$  into a “thinking time” for the agent by  $t = a\sqrt{N\gamma} + b$ , where  $a = 4\text{s}$  and  $b = 0.5\text{s}$ .

The main model has 10 parameters: the pruning threshold  $\theta$ , the stopping probability  $\gamma$ , the lapse rate  $\lambda$ , the feature drop rate  $\delta$ , the active scaling constant  $C$ , and the feature weights  $w_{\text{center}}$ ,  $w_{\text{connected 2-in-a-row}}$ ,  $w_{\text{unconnected 2-in-a-row}}$ ,  $w_{\text{3-in-a-row}}$ ,  $w_{\text{4-in-a-row}}$ . We do not add a parameter for the variance of the value noise, since changing the noise distribution from  $\mathcal{N}(0, 1)$  to  $\mathcal{N}(0, \sigma^2)$  has the same effect as changing  $\theta \rightarrow \frac{\theta}{\sigma}$  and  $w \rightarrow \frac{w}{\sigma}$  for each feature. Therefore, adding  $\sigma$  would over-parametrize the model and cause  $\sigma$ ,  $\theta$ , and  $\{w_i\}$  to be unidentifiable from data.

## A.4 STOPPING RULE

To make predictions for response times, we amend the model with a stopping rule, which terminates the best-first search algorithm when the model’s preferred move in the root node remains unchanged for 50 consecutive iterations. This stopping rule is in addition to the random stopping rule implied by the stopping probability  $\gamma$ . To obtain a response time prediction from the model with given parameters on a given trial, we simulate 100 moves from the model, and measure the average number of search iterations executed before the algorithm terminates.

The stopping rule allows the model to predict differences in response times across trials for the same participant. In positions where one move is clearly preferred over other options (for example, if the opponent has a direct threat that needs to be parried), the algorithm will quickly rule out any alternatives and spend its remaining time calculating the future consequences of that move. Therefore, the preferred move in the root node remains unchanged for many iterations, and the stopping rule causes the search to terminate, resulting in a low number of iterations. In other positions with many plausible alternatives, the model will waver, the stopping rule will not

trigger, and the model will search longer. Thus, the termination rule is one way of incorporating the known effect of decision difficulty on response times. Other plausible stopping rules might behave similarly. For example, one can base the termination criterion on relative node values or node visits, as in MCTS.

## A.5 MODEL COMPARISON

### A.5.1 LESIONS

Our first set of alternative models are lesion models, obtained by removing components from the main model. Each lesion can be implemented by fixing a parameter to a constant. The No center, No connected 2-in-a-row, No unconnected 2-in-a-row, No 3-in-a-row, and No 4-in-a-row models are obtained by setting the respective feature weight to zero. The No feature drop model is obtained by fixing  $\delta$  to zero, and the No active scaling model results from fixing  $C$  to 1. To obtain the No pruning model, we fix  $\theta$  to 20,000, which is larger than any value difference that occurs in search and causes the model to never prune. Note that the model cannot compensate by increasing feature weights since their order of magnitude is yoked by fixing the value noise to have unit variance. Finally, the No tree model is achieved by fixing  $\gamma$  to 1. This causes the algorithm to stop after 1 iteration, in which case it will have expanded only the root node, and its choice will be the highest-value child. Pruning lower-value children does not affect this choice, so  $\theta$  is not a parameter in this model.

### A.5.2 MODIFICATIONS

In our first modified model, Fixed iterations, we change the stopping routine `Stop` from a stochastic function to a deterministic function that returns `true` whenever the number of iterations has exceeded a constant  $N$ . In the Fixed depth model, we amend the search process to explore every

branch up to a fixed depth  $D$ . In the Fixed branching model, we amend the pruning rule to keep the  $K$  highest-value children in each node (lowest-value when white is to move). If the expanded node has less than  $K$  children, the algorithm prunes nothing. Next, we consider removing the feature drop mechanism and instead applying a function in which each child is pruned with a probability  $\varepsilon$  in `ExpandNode` before the value-based pruning, resulting in the Square dropping model. For the Optimal weights model, we restrict the feature weights  $\{w_i\}$  to a constant vector, which we chose by maximizing the Pearson correlation between  $\tanh(V(s)/20)$  and the game-theoretic value  $\tilde{V}(s)$  across all states  $s$  that occurred in the human-versus-human experiment.

Finally, we consider Monte Carlo tree search (MCTS). In this algorithm, instead of evaluating a state with  $V(s)$ , we perform a rollout: a simulated game starting from state  $s$  between two agents that follow a myopic policy. That is, in state  $s'$ , the agent chooses the move  $m$  that maximizes  $V(s'+m)$ , or the one that minimizes it when white is to move. We then assign a value of 1 to state  $s$  if the rollout results in a win for black, 0 for white wins, and  $\frac{1}{2}$  if the game is a draw. Note that, since the evaluation function contains noise, the myopic policy and the outcome of the rollout are also stochastic. We only perform a single rollout when evaluating a state.

After performing a rollout, MCTS backpropagates by averaging rather than minimax, ensuring that the value of each intermediate node of the tree is equal to the average outcome of the rollouts conducted in all descendants of that node. We amend the best-first selection rule as follows:

$$n = \operatorname{argmax}_{c \in \text{children}(n)} c.\text{val} \quad (\text{A.6})$$

to the UCB formula

$$n = \operatorname{argmax}_{c \in \text{children}(n)} c.\text{val} + C_{\text{exp}} \sqrt{\frac{\log(n.N_{\text{rollouts}})}{c.N_{\text{rollouts}}}} \quad (\text{A.7})$$

where  $n.N_{\text{rollouts}}$  counts the number of rollouts that have been conducted in node  $n$  or any of its descendants, and  $C_{\text{exp}}$  is a parameter that controls the balance between exploitation (investigating high-value children) and exploration (investigating children that haven't been investigated

much). Finally, after the tree search terminates, the algorithm makes a move by maximizing  $N_{\text{rollouts}}$  across all children of the root node.

#### A.5.2.1 EXTENSIONS

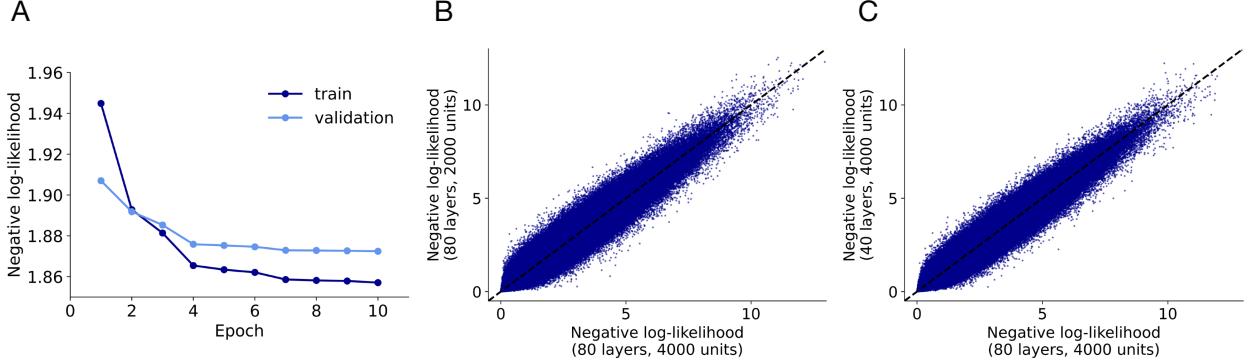
We create the Orientation-dependent weights by multiplying the weight of vertically or diagonally oriented features by scaling constants  $c_{\text{vert}}$  and  $c_{\text{diag}}$ , respectively. For the Orientation-dependent dropping model, we allow the feature drop rate for horizontally, vertically or diagonally oriented features to vary, whereas in the Type-dependent dropping, we let the drop rate depend on the feature type. In the Triangle model, we include a feature which counts the number of times that any of a set of 3-piece patterns occurs on the board. Finally, the Opponent scaling model extends the main model by adding a scaling constant  $c_{\text{opp}}$  that multiplies weights of features belonging to the opponent. Note that opponent scaling and active scaling are dissociated since the former multiplies weights of the opponent’s features regardless of whose move it is, whereas the latter is adaptive.

## B | APPENDIX FOR CHAPTER 3

### B.1 NEURAL NETWORK TRAINING AND TESTING

The neural networks were all implemented using PyTorch [Paszke et al. 2019]. We used stochastic gradient descent for training and reduced the learning rate if the loss associated with the validation set was stagnant for 3 epochs. The initial learning rate was set to 0.001, and we decayed the learning rate by a multiplicative factor of 0.2 at each decrease. We trained each network for a total of 10 epochs using a cross entropy loss function. Cross entropy loss is an appropriate choice for our task because it combines a logarithmic Softmax and negative log-likelihood, and is often used for classification problems where the goal is to assign weight to each of a number of classes. All layers had their biases initialized to 0 and weights drawn from a normal distribution with mean 0 and standard deviation 0.01, and we use a batch size of 128. In Figure B.1A, we show example training and validation curves for the largest network. Curves for the remaining networks look similar, with a sharp improvement in the first few epochs that flattens out in later epochs, along with a minor effect caused by decreasing the learning rate.

In Figure B.1B-C, we validated the network’s training procedure by showing that the likelihoods are correlated between the largest network and the networks that are one step smaller in terms of either number of units per layer or number of hidden layers. In Table B.1, we enumerate all combinations of networks that we trained, including the average negative log-likelihood per move and overall accuracy on the test set.



**Figure B.1:** Neural network training procedure. **(A)** Training and validation curves over the 10 training epochs for the best network, which has 80 hidden layers and 4,000 units per layer. **(B)** Scatterplot of the negative log-likelihood for every move on the test data set between the best network and the network with 80 hidden layers but only 2,000 units per layer ( $\rho = 0.99, p < 2 \cdot 10^{-308}$ ). **(C)** Same as (B), but comparing the best network with the network that has 4,000 units per layer but only 40 hidden layers ( $\rho = 0.99, p < 2 \cdot 10^{-308}$ ).

## B.2 MODEL EXTENSION SPECIFICATION

To iterate on the baseline model, we implemented mechanisms inspired by comparison with the best neural network. Specifically, we investigated the board positions that resulted in the largest difference in terms of predictability between the neural network and the baseline model. This resulted in three model variants, which we describe in this section. Note that each model addition is kept for later extensions. For example, the defensive weighting model has all mechanisms of the baseline model, the opening bias mechanism, and the additional defensive weighting mechanism. The negative log-likelihoods for each of these model variants on the test data set as well as their overall accuracy are shown in Table B.2.

The opening bias model was inspired by early game moves that are played towards the left side of the board and the corners of the board, a phenomenon that can be corroborated by looking at the histogram of first moves made by users in the data set (Figure B.5D). Therefore, we added 4 feature weights to  $V(s)$ , which are only active on the opening move and correspond to each of the corners of the board. This allows the model to more flexibly predict human moves that stray from

<b>Number of hidden layers</b>	<b>Number of units per layer</b>	<b>NLL</b>	<b>Accuracy</b>
5	200	1.98	38.75%
10	200	1.95	39.68%
20	200	1.92	40.20%
40	200	1.91	40.63%
80	200	1.90	40.88%
5	500	1.95	39.55%
10	500	1.93	40.18%
20	500	1.91	40.73%
40	500	1.89	41.04%
80	500	1.89	41.22%
5	1000	1.93	39.93%
10	1000	1.91	40.68%
20	1000	1.89	41.03%
40	1000	1.88	41.30%
80	1000	1.88	41.41%
5	2000	1.92	40.41%
10	2000	1.89	40.98%
20	2000	1.88	41.33%
40	2000	1.88	41.49%
80	2000	1.87	41.58%
5	4000	1.90	40.69%
10	4000	1.88	41.32%
20	4000	1.87	41.55%
40	4000	1.87	41.67%
80	4000	1.87	41.71%

**Table B.1:** All trained neural networks designated by a combination of the number of hidden layers and number of units per layer. Each network has a corresponding average negative log-likelihood per move as well as overall prediction accuracy on the test data.

the center of the board. While this addition improved the model fit, the magnitude of the effect is minor, likely because it only affects 1 out of the possible 18 moves that a player makes in any given game. A more sophisticated mechanism could extend these biases to all moves by decaying their influence throughout gameplay. Even further, humans likely use a retrospective system in early game decisions where planning is less informative [Kuperwajs et al. 2019]. If this is the case, these biases might be shaped by habit or the success of certain opening sequences in previous games. Investigating the tradeoff between prospective and retrospective decision-making is out

Model	NLL	Accuracy
Baseline	2.17	34.88%
Opening bias	2.16	34.86%
Defensive weighting	2.13	35.23%
Phantom features	2.14	34.51%

**Table B.2:** All tested cognitive models designated by added mechanism. Each model represents the best parameter combination on the training data over 20 runs, and has a corresponding average negative log-likelihood per move as well as overall prediction accuracy on the test data.

of the scope of the current chapter, but is an entire field in and of itself and integrating such a mechanism into this model would most likely improve its performance.

The defensive weighting model was inspired by situations where the baseline model failed to defend against immediate threats. In specific positions where the human player should defend against an immediate loss, the baseline model predicts that the user will instead create high-value features for themselves elsewhere on the board. This happened when the created features were valued much more highly than the removed 3-in-a-row feature for the opponent, such that the defending move was pruned from the decision tree. Both the neural network and the data did not show this pattern of oversights. To fix this, we added another feature weight to  $V(s)$ , which explicitly targets immediate threats made by the opponent. With this change, leaving a winning move for the opponent on the board is devalued such that the move that defends against this threat is not pruned from the tree. Additionally, we noticed that the baseline model could not overlook 4-in-a-row features during its search because it used the correct win condition to build the tree. To enable overlooking the 4-in-a-row feature, we made the detection of terminal states dependent on the 4-in-a-row feature instead and fixed the value for this feature to the arbitrary, very high value of 10,000. This is certainly not the only possible implementation to push the model to consider defending against immediate threats, but it successfully eliminated these errors and improved the model fit. Beyond that, it is certainly plausible that people pay special attention to opponent threats as a cognitive mechanism.

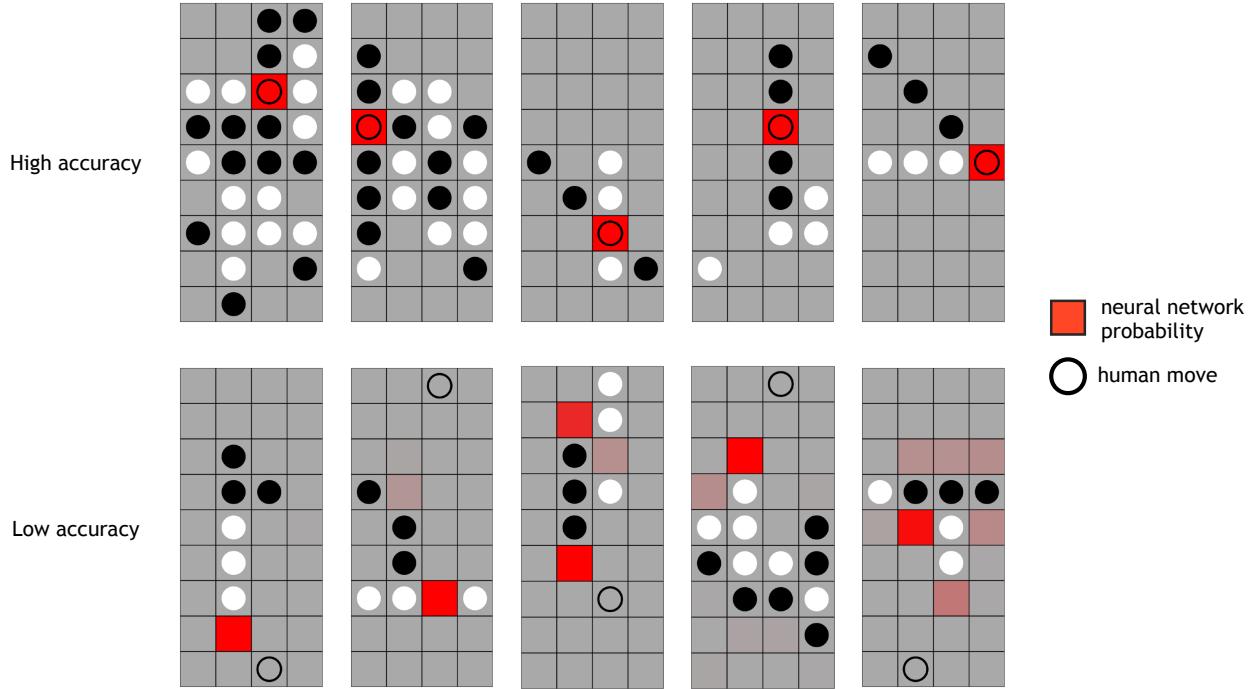
Finally, the phantom features model was inspired by board positions in which the the network

and humans seemed to create or defend against 3-in-a-row features where there is no space for the final piece needed to win the game. The features in the baseline model all require that there are empty squares on the board to complete a 4-in-a-row. We enumerated the 4 3-in-a-rows that occur in the corners of the board following this pattern, and added a feature weight that scales their contribution to  $V(s)$ . Interestingly, this did not improve the model fit from the defensive weighting model although it did improve the predictions for the board positions that we based this extension on. This means that adding these features is causing the model to perform worse in other positions. A more general mechanism might take this tradeoff into account by, for example, checking the proximity of the piece that is being considered by the player to the rest of their own pieces. Another possibility is that these boards do not represent phantom features at all, but rather a different mechanism that can still account for these board positions. Regardless, this extension can be iterated on further to create a better cognitive model.

### B.3 MODEL FITTING

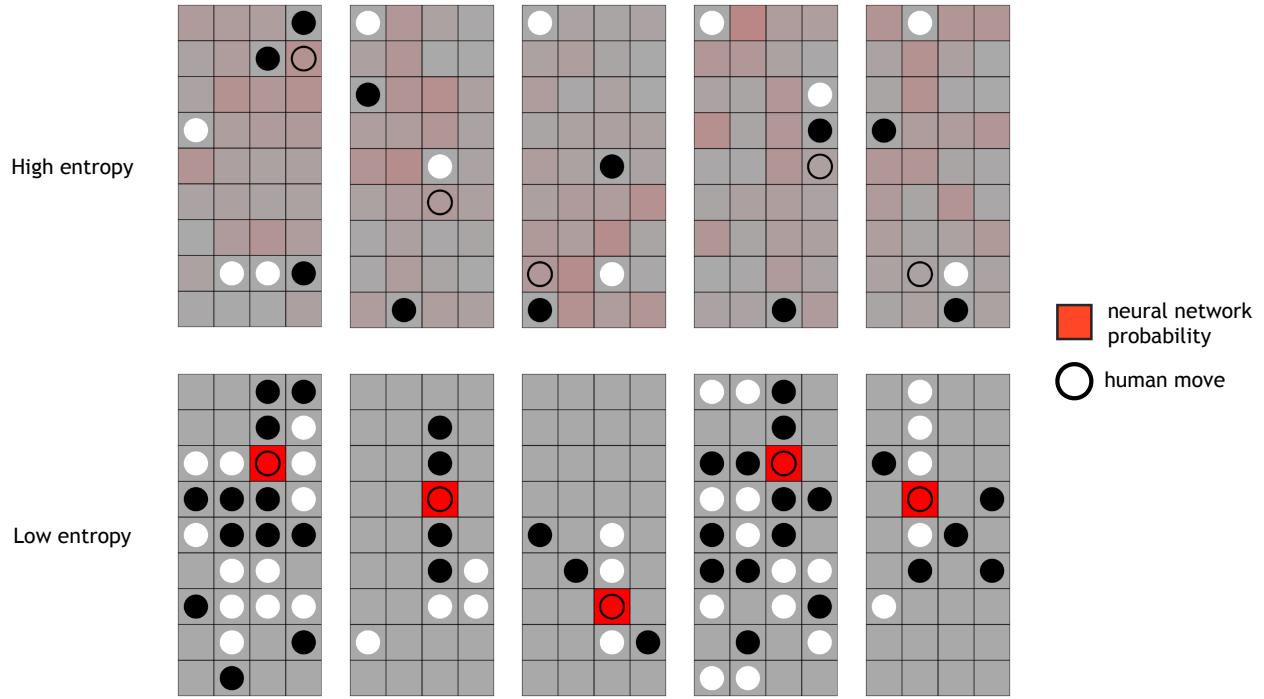
The baseline model has 9 parameters: the 5 feature weights, the pruning threshold  $\theta$ , stopping probability  $\gamma$ , the feature drop rate  $\delta$ , and the lapse rate  $\lambda$ . For the various model improvements, we added a few additional parameters: the 4 corner weights for the opening move, the defensive scaling weight, and the phantom features weight. For the defensive weighting and phantom features model variants, we removed one of the feature weights from the baseline model, namely the one for 4-in-a-row that is replaced by a fixed high value. Therefore, our model improvements have a total of either 13 or 14 parameters.

In the previous chapter, the cognitive model was fit to individual users using 5-fold cross-validation to reduce overfitting. Since we wanted to make the model fits comparable with the neural network, we inferred parameters while treating the entire training set as one user, effectively eliminating any concerns regarding overfitting. We continued using both IBS [van Opheus-



**Figure B.2:** Example high and low accuracy board positions for the neural network’s predictions. The user is playing black while the computer opponent is playing white. Additionally, the red shading indicates the probability distribution of the network’s move prediction and the open circle indicates the user’s selected move.

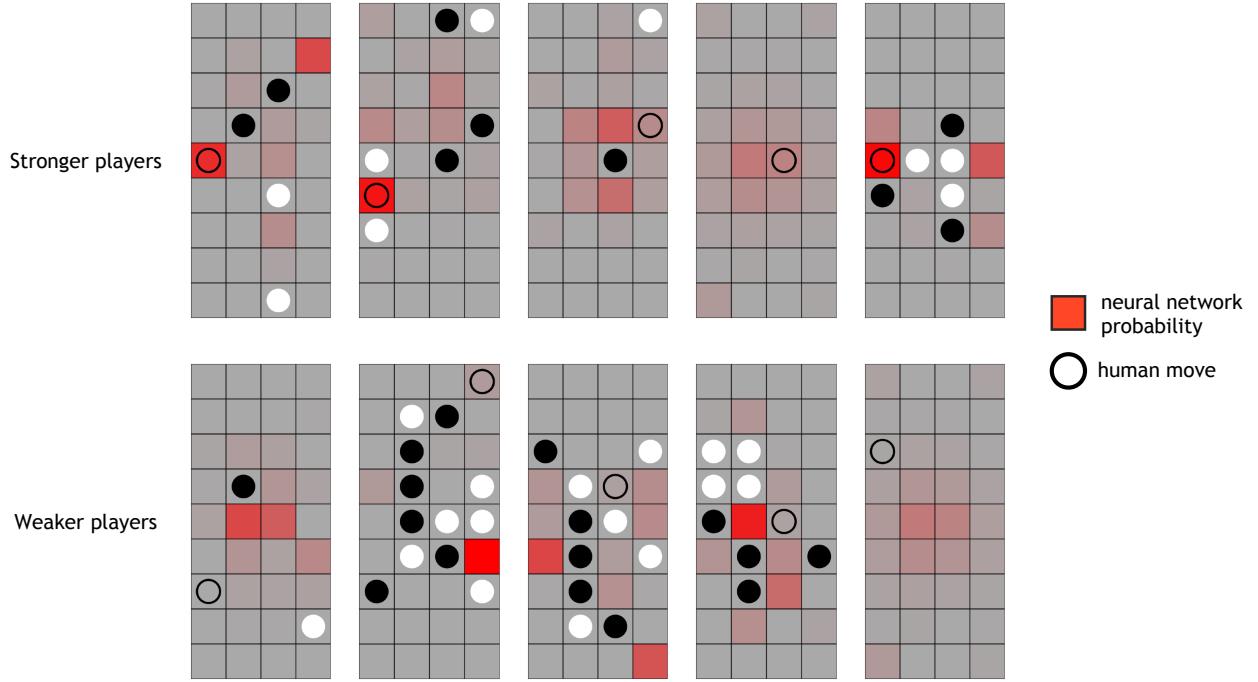
den et al. 2020] as well as Bayesian adaptive direct search [Acerbi and Ma 2017] as before. To make this computationally feasible, we evaluated the log-likelihood on 100,000 trials that are randomly sampled for each evaluation. We tested both lower and higher numbers of evaluations, deciding on the value that balanced reliability of the likelihood estimates across training runs as well as fitting time. For each model variant, we ran the fitting procedure 20 different times, choosing the combination of parameters that resulted in the best log-likelihood. On the test data set, we then ran 100 repetitions to estimate the log-likelihoods for each move and 200 simulations in each board position to get a probability distribution over potential moves. On our hardware, fitting takes anywhere from one or two days to a week depending on the size of the network or the number of evaluations used for the planning model. Cumulatively, we trained a total of 25 neural networks and fit a total of 80 cognitive models.



**Figure B.3:** Example high and low entropy board positions for the neural network’s output distribution. The format for the board positions is the same as for Figure B.2.

## B.4 EXAMPLE BOARD POSITIONS

In order to ensure that the best neural network is capturing human gameplay in 4-in-a-row with its predictions, we examined board positions sorted according to different criteria. In this section we provide a number of illustrative examples for each analysis. For the accuracy analysis, we sorted board positions by the negative log-likelihood of the network’s prediction compared to the data (Figure B.2). High accuracy boards were those in which there was an immediate win or loss present, or a combination of both, and the user made the same move as the network. Low accuracy boards were those in which the human made a clear error in gameplay, usually playing far away from the pieces on the board. These positions also typically include an immediate win or loss, or just a generally strong move to make that the network favors. This further serves to show that the network is approximating human behavior, minus the mistakes that we are not



**Figure B.4:** Example board positions played by stronger and weaker users for the neural network’s predictions. The format for the board positions is the same as for Figure B.2.

interested in capturing with any model. For the entropy analysis, we sorted board positions by the entropy  $H$  of the network’s output distribution  $p_n$  (Figure B.3):

$$H = - \sum p_n \cdot \log(p_n). \quad (\text{B.1})$$

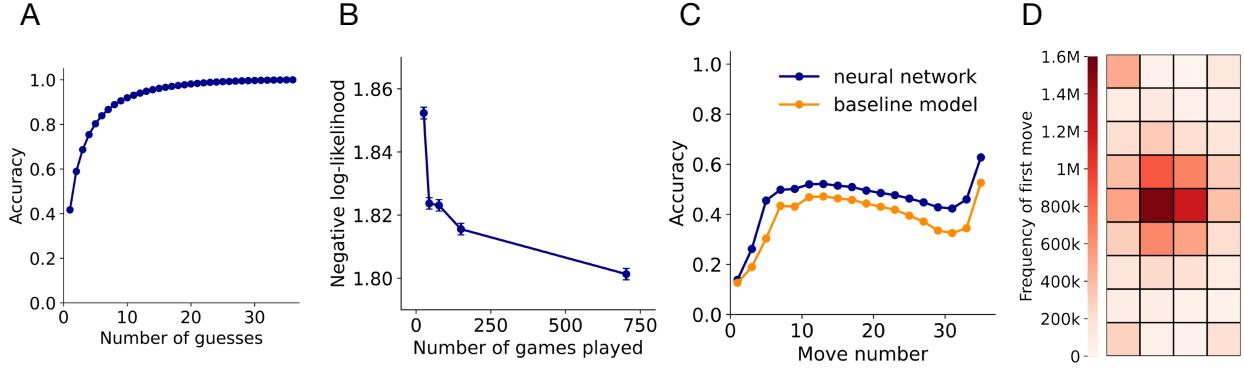
High entropy boards were those in which the network was unsure of where to play, typically consisting of only a few pieces on the board where presumably human behavior is highly variable. Even in these positions, the network assigns higher probability to squares adjacent to existing pieces on the board where people tend to play. Meanwhile, low entropy boards where the network is sure of its prediction were similar to high accuracy boards, with the network and the data agreeing on exploiting 3-in-a-rows for the player or defending against opponent 3-in-a-rows. Once again, this shows that the network is behaving in a way that aligns with our intuitions about gameplay in 4-in-a-row, confidently predicting human moves when user behavior is more

stereotyped and there are more pieces on the board.

For the playing strength analysis, we estimated a user’s playing strength from games against computer opponents using Elo ratings [Elo 1978] via Bayeselo [Hunter 2004] (Figure B.4). Ratings calculated for relatively few games can be statistically unreliable, so we included only players who had played at least 20 total games played in our analysis, resulting in 115,968 unique users. We used a common baseline to compute Elo ratings across all experimental data, which outputs an Elo rating for each user and AI agent in the data set that can be directly compared to one another. Moves made by players with higher Elo ratings tended to be easier for the network to correctly predict, as stronger players play more consistently and make fewer errors. For example, strong players tend to create features for themselves or block opponent features when it is rational to do so, and they play in the most common squares of the board as predicted by the network in the opening. Moves made by players with lower Elo ratings tended to be more difficult for the network to correctly predict, as weaker players make more mistakes and have more lapses in gameplay. This includes behaviors like playing far away from existing pieces, overlooking opportunities to create or defend against strong features, and playing away from the center on the first move.

## B.5 NEURAL NETWORK VALIDATION

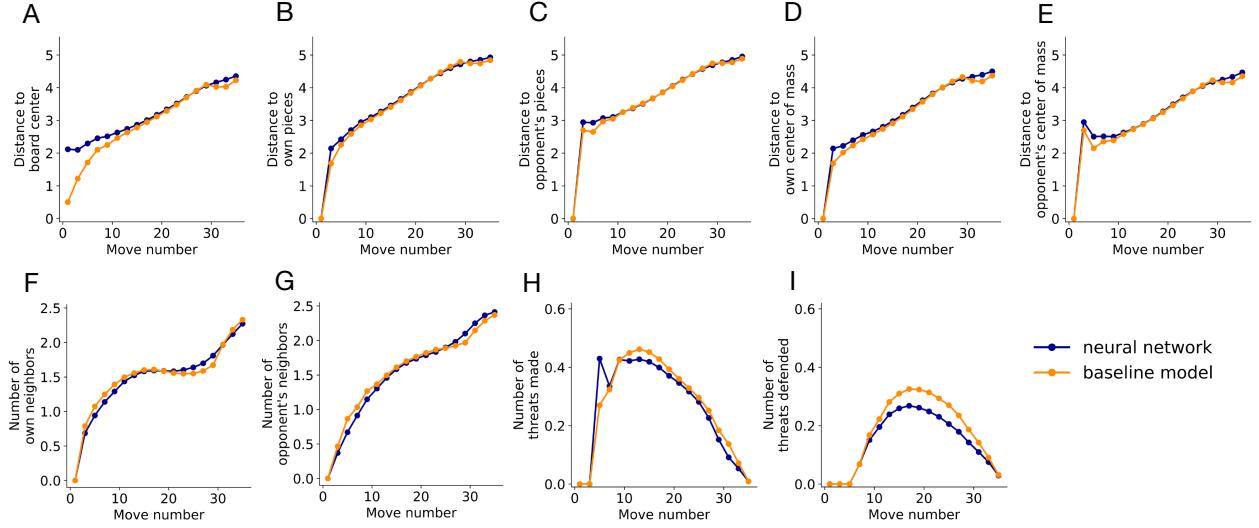
We conducted a few additional analyses to corroborate the neural network’s performance and compare with the cognitive model. First, we validated that giving the network more opportunities to correctly predict the human move quickly converged to 100% accuracy (Figure B.5A). This was indeed the case when averaged by move number, as the network starts out at its overall accuracy of 41.71% with a single guess, and converges to near perfect accuracy with only a few additional guesses. This is important as a sanity check that even if the network is wrong about the human move, the correct move is still among the top candidates. Next, we investigated the effect



**Figure B.5:** Additional validation that the neural network achieves a satisfactory upper bound on goodness of fit and exhibits human-like behavior. **(A)** Accuracy as a function of the number of guesses given to the neural network to correctly predict the human move, averaged across the test set. **(B)** Negative log-likelihood on the test data set as a function of the user's experience level, or total number of games played (binned into quantiles). **(C)** Accuracy as a function of move number for the neural network (blue) and baseline model (orange), averaged across the test set. **(D)** Histogram of the user's first move across all games in the data set, which the network approximates. An example board position where the network's opening move distribution is shown can be found in Figure B.4.

of experience on the negative log-likelihood of the network's predictions (Figure B.5B). Number of games played is roughly correlated with Elo ratings, and repeating our playing strength analysis with experience resulted in a similar decreasing trend. Further, looking at the board positions associated with different experience levels provided additional evidence for the relationship between playing strength and experience. As expected, high experience players made moves and were predicted by the network similarly to stronger players and low experience players make errors and were predicted by the network similarly to weaker players.

In order to further compare the neural network with the cognitive model, we repeated a number of analyses from Chapter 3 with the model. One of these is average accuracy as a function of move number, where the model shows a similar trend across gameplay as the network but with a consistently lower accuracy (Figure B.5C). Finally, we computed all 9 summary statistics for the model as well. Overall, the model performs similarly to the network (and in turn the data) on many of these, highlighting the fact that the model was already capturing many aspects of human play. This was expected given the large number of iterations on the model in previous



**Figure B.6:** Comparison of summary statistics between the neural network and the baseline model. Each statistic is averaged by move number for the neural network (blue lines) and the baseline model (orange lines).

work [van Opheusden et al. 2023]. The largest deviations between the network and the model occurred in a few distinct places: (1) the distance to the center of the board in the early game where the network strayed from the center more than the model does, (2) the number of threats made where the model both overestimated and underestimated the rate at which to create 3-in-a-rows at different points in gameplay, and (3) the number of threats defended against where the model played too defensively in the middle game. These differences relate to the mechanisms we extracted from our comparison of the model and the network that we ended up implementing in our model extensions. This analysis was done in the main text by looking at board positions with high KL divergence  $L$  between the network’s output distribution  $p_n$  and the model’s output distribution  $p_b$  on every move, defined as:

$$L = p_b \cdot \log \frac{p_b}{p_n} = p_b \cdot (\log p_b - \log p_n). \quad (\text{B.2})$$

# C | APPENDIX FOR CHAPTER 4

## C.1 DETAILED MODEL DERIVATION

Here we provide the full model derivation for the meta-planner. We assume that the agent has the option of executing a tree search policy  $\pi$ , and that a state-action pair  $s, a$  has a theoretical long-running expected reward under  $\pi$ ,  $Q_a$ . This value is not known and therefore has to be approximated. In our model, we take an inference view, where  $Q_a$  is unknown to the agent, and the agent tries to build a probability distribution over each  $Q_a$ .

### C.1.1 GENERATIVE MODEL

We begin by describing the generative model for our framework, meaning the set of underlying assumptions that generate the measurements the agent has available to them prior to making a decision.

#### C.1.1.1 DISTRIBUTION OF $Q$ VALUES

We assume that the true  $Q$ -value for an action follows a distribution  $p(Q_a)$ . We assume a normal distribution as follows:

$$Q_a \sim \mathcal{N}(\mu_0, \sigma_0^2). \quad (\text{C.1})$$

The Q-values per action are independently drawn from this distribution.

### C.1.1.2 RETROSPECTION

We model an experience with action  $a$  in state  $s$  as a noisy measurement of the true  $Q$  value:

$$q_{\text{retro},a} \sim \mathcal{N}(Q_a, \sigma_{\text{retro}}^2). \quad (\text{C.2})$$

The agent can have numerous experiences with each state-action pair, and we denote  $n_{\text{retro},a}$  as the number of past experiences with action  $a$  in state  $s$ . Together, these measurements form a vector  $\mathbf{q}_{\text{retro},a}$  that has  $n_{\text{retro},a}$  entries.

### C.1.1.3 PROSPECTION

Next, we assume that a tree expansion can be represented as another noisy measurement  $q_a$  of the true value  $Q_a$  of action  $a$  in state  $s$ :

$$q_a \sim \mathcal{N}(Q_a, \sigma^2). \quad (\text{C.3})$$

In a tree search algorithm, the noisy measurement may be obtained through a heuristic value function. It is an open question to what extent heuristic values obtained in practice follow the normal distribution above and, in particular, if they are unbiased. The core part of our framework is a statistical model maintained by the agent of the effects of prospective tree search, without actually doing tree search. Additionally, it is assumed within this statistical model that an iteration of the tree search algorithm working on a branch that starts with action  $a$  produces a new, independent measurement of  $Q_a$ .

### C.1.2 INFERENCE

The overarching goal of this framework is to allow the agent to decide whether and in which direction to plan. We take a normative approach, where the agent makes a decision by calculating the expected gain obtained from making an additional measurement of each action and selecting the action that maximizes this gain. The algorithm uses Bayesian inference and works as follows:

1. The agent has the constraint of only being able to consider the effects of planning for a single action at a time. Thus, we consider the problem to be deciding which action to sample another measurement of the underlying value from. The agent iterates over actions and computes the value of each action conditioned on sampling a specific action: this results in no change in the value of the mean for actions that aren't sampled, and a distribution over the mean of the action that is sampled.
2. To combine across actions  $N$ , the agent computes the max distribution over the possible means (which now has  $N - 1$  point estimates and a single distribution) of each action given the new sample, and  $U_a$  is the expected value of this max distribution. This is the utility of sampling, and the process is repeated for all possible actions to sample.
3. Finally, an update rule is used to decide whether it is worthwhile to keep planning. If the maximum utility across possible samples is less than a fixed cost  $c$ , then no more planning is necessary. Otherwise, the agent samples the action that maximizes utility and updates the posterior for that action accordingly.

### C.1.2.1 RETROSPECTIVE LIKELIHOOD

The retrospective likelihood captures the information that the agent has about the Q-values based on previous experiences. The likelihood we are interested in is over  $Q_a$  based on the data  $\mathbf{q}_{\text{retro},a}$ :

$$\begin{aligned}
 \mathcal{L}(Q_a; \mathbf{q}_{\text{retro},a}) &= p(\mathbf{q}_{\text{retro},a} | Q_a, \sigma_{\text{retro}}^2) \\
 &= \prod_{i=1}^{n_{\text{retro},a}} p(q_{\text{retro},a_i} | Q_a, \sigma_{\text{retro}}^2) \\
 &= \prod_{i=1}^{n_{\text{retro},a}} \mathcal{N}(q_{\text{retro},a_i}; Q_a, \sigma_{\text{retro}}^2) \\
 &\propto \mathcal{N}\left(Q_a; \bar{q}_{\text{retro},a}, \frac{\sigma_{\text{retro}}^2}{n_{\text{retro},a}}\right),
 \end{aligned} \tag{C.4}$$

where  $\bar{q}_{\text{retro},a} \equiv \sum_{i=1}^{n_{\text{retro},a}} q_{\text{retro},a_i}$ . This likelihood function captures all the information that the agent can gain from previous experiences.

### C.1.2.2 PROSPECTIVE LIKELIHOOD

After  $n_a$  measurements are taken for each action  $a$ , the agent has a vector of measurements  $\mathbf{q}_a$  with  $n_a$  entries. The prospective likelihood over  $Q_a$  is then a product of the likelihoods associated with the individual measurements, similar to the retrospective likelihood:

$$\mathcal{L}(Q_a; \mathbf{q}_a) \propto \mathcal{N}\left(Q_a; \bar{q}_a, \frac{\sigma^2}{n_a}\right), \tag{C.5}$$

where  $\bar{q}_a \equiv \sum_{j=1}^{n_a} q_{a_j}$ . For now, we assume  $\bar{q}_a$  to be known. Later, we will marginalize over all possible values of  $\bar{q}_a$ .

### C.1.2.3 POSTERIOR

The posterior is the normalized product of a prior and two likelihoods that we assume to be independent. We compute the posterior for each action with all currently available information:

$$\begin{aligned}
p(Q_a | \mathbf{q}_{\text{retro},a}, \mathbf{q}_a) &= p(Q_a)p(\mathbf{q}_{\text{retro},a}|Q)p(\mathbf{q}_a|Q_a) \\
&= \mathcal{N}(Q_a; \mu_a, \sigma_a^2) \\
\mu_a &= \frac{J_0 \mu_0 + J_{\text{retro}} n_{\text{retro},a} \bar{q}_{\text{retro},a} + J n_a \bar{q}_a}{J_a} \\
\sigma_a^2 &= \frac{1}{J_a}
\end{aligned} \tag{C.6}$$

where we define the following precision quantities:

$$\begin{aligned}
J_{\text{retro}} &\equiv \frac{1}{\sigma_{\text{retro}}^2} \\
J &\equiv \frac{1}{\sigma^2} \\
J_0 &\equiv \frac{1}{\sigma_0^2} \\
J_a &\equiv J_0 + n_{\text{retro},a} J_{\text{retro}} + n_a J.
\end{aligned} \tag{C.7}$$

### C.1.2.4 EXPECTED VALUE

Next, we rewrite the future posterior if the agent were to make an additional measurement  $q'_a$ . Under this assumption, the mean and variance of the posterior would be

$$\begin{aligned}
\mu'_a &= \frac{J_0 \mu_0 + J_{\text{retro}} n_{\text{retro},a} \bar{q}_{\text{retro}} + J(n_a \bar{q}_a + q'_a)}{J'_a} \\
\sigma'^2_a &= \frac{1}{J'_a}.
\end{aligned} \tag{C.8}$$

We can rewrite the future posterior:

$$\mu'_a = k_1 q'_a + k_0 \quad (\text{C.9})$$

where

$$\begin{aligned} k_0 &\equiv \frac{J_a \mu'_a}{J'_a} \\ k_1 &\equiv \frac{J}{J'_a}. \end{aligned} \quad (\text{C.10})$$

We will also later use the fact that

$$J'_a = J + J_a. \quad (\text{C.11})$$

#### C.1.2.5 DISTRIBUTION OF THE FUTURE MEAN MEASUREMENT

Now, the new measurement  $q'_a$ , which the agent receives if they sample another measurement for that action, is unknown and has to be marginalized over. Conceptually, this is the distribution over future prospective measurements one step into the future for  $a$  given current information.

We compute this by marginalizing over the current possible values of  $Q_a$ :

$$\begin{aligned} p(q'_a | \bar{q}_a, \mathbf{q}_{\text{retro},a}) &= \int p(q'_a | Q) p(Q | \bar{q}_a, \mathbf{q}_{\text{retro},a}) dQ_a \\ &= \int \mathcal{N}(q'_a; Q_a, \sigma^2) \mathcal{N}(Q_a; \mu_a, \sigma_a^2) dQ_a \\ &= \mathcal{N}(q'_a; \mu_a, \sigma^2 + \sigma_a^2). \end{aligned} \quad (\text{C.12})$$

Finally, we know that the expected value of  $\mu'_a$  given  $\mathbf{q}_{\text{retro},a}$ ,  $\mathbf{q}_a$  must be a normal distribution with the following mean and variance:

$$\begin{aligned}
 \mathbb{E}[\mu'_a] &= k_1 \mu_a + k_0 \\
 &= \frac{J\mu_a + J_a \mu_a}{J'_a} \\
 &= \frac{\mu_a(J + J_a)}{J'_a} \\
 &= \frac{\mu_a J'_a}{J'_a} \\
 &= \mu_a \\
 \text{Var}[\mu'_a] &= k_1^2 (\sigma^2 + \sigma_a^2) \\
 &= \frac{J^2}{J_a^2} \left( \frac{1}{J} + \frac{1}{J_a} \right) \\
 &= \frac{J}{J'_a J_a} \\
 &= \frac{J}{J_a (J + J_a)} \\
 &= \frac{1}{J_a (1 + \frac{J_a}{J})}.
 \end{aligned} \tag{C.13}$$

Note that this implies that, given an additional sample, the mean of the resultant distribution stays at  $\mu_a$  while the variance becomes narrower.

#### C.1.2.6 EXPECTED UTILITY OF MAKING ANOTHER MEASUREMENT

To combine across actions, we need to compute the expected utility of making another measurement of action  $a$ . First, we compute the maximum of posterior means before making an additional measurement:

$$M \equiv \max_a \mu_a. \tag{C.14}$$

The maximum of posterior means after making an additional measurement of action  $a$  is

$$M'_a \equiv \max \left( \mu'_a, \max_{b \neq a} \mu_b \right). \quad (\text{C.15})$$

This is a random variable because we don't know  $\mu'_a$  exactly. However, we can take the expected value  $\mathbb{E}_{\mu'_a}[M'_a]$ . We know this distribution from Equation C.13, and can evaluate the expected value of this quantity via two methods: (1) analytically, by computing the max distribution (as outlined in the next section) over all actions and taking the mean of the resultant distribution, and (2) using Monte-Carlo simulation, where  $\mathbb{E}_{\mu'_a}$  is normally distributed. These methods produce identical solutions, with the analytical method having a run time advantage of about an order of magnitude over sampling. Within our framework, the mathematical reason why planning is beneficial is that the expected value of a maximum is greater than the maximum of the expected values. Finally, the expected utility of making another measurement of action  $a$ , which we call the utility of sampling, is

$$U_a = \mathbb{E}_{\mu'_a}[M'_a] - M. \quad (\text{C.16})$$

This utility computation has to be repeated across all actions. Then, we propose that the agent chooses to sample if the maximum utility exceeds a fixed cost  $c$ :

$$\text{sample if } \max_a U_a > c. \quad (\text{C.17})$$

If this holds true, then the agent makes a measurement of the action that maximizes  $U_a$ . Denoting this measurement by  $q'_a$ , we calculate the parameters of the updated posterior for  $a$ ,  $\mu'_a$  and  $\sigma'^2_a$ :

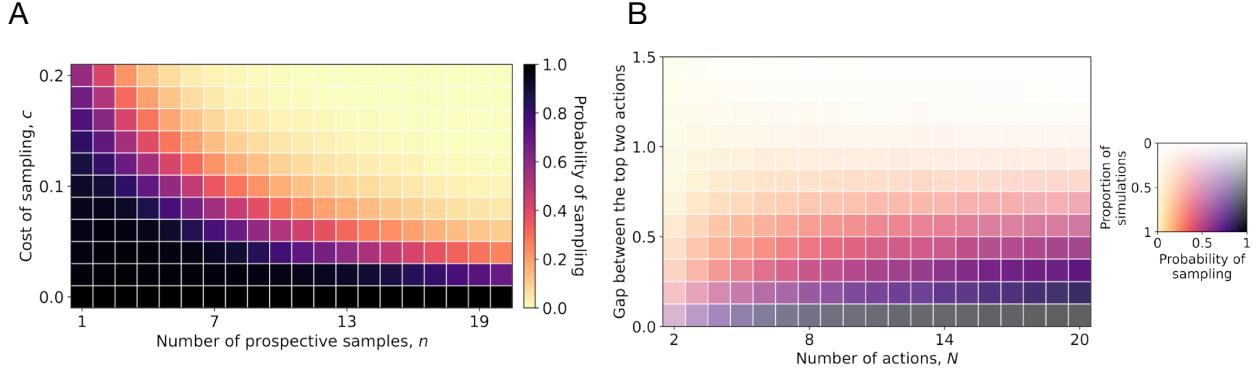
$$\begin{aligned}\mu'_a &= \frac{J_a \mu_a + J q'_a}{J_a + J} \\ \sigma'^2_a &= \frac{1}{J_a + J}.\end{aligned}\tag{C.18}$$

All other actions keep the same posterior. Once again, note that here we are using the terms associated with a new prospective measurement for action  $a$  rather than any terms for action  $a$  that are computed virtually in the inference procedure. At this point the agent has selected an action and sampled it, and the meta-planner can use the updated values to repeat the inference process to decide if it is worth continuing to plan.

#### C.1.2.7 DISTRIBUTION OF THE MAXIMUM OF RANDOM VARIABLES

The distribution of the maximum of random variables can in general be computer semi-analytically, and we briefly recap this derivation in a general form here. We are interested in the distribution of the maximum variable  $M = \max_i X_i$ , where the independent, real-valued random variables  $X_i$  have densities  $p_i(x)$  and cumulative distribution functions  $F_i(x)$ . We know that  $M \leq m$  if and only if  $M \leq m_i$  for all  $i$ . Thus, we can calculate the cumulative distribution function of  $M$  as:

$$\begin{aligned}F_M(m) &= P(M \leq m) \\ &= P(X_1 \leq m, \dots, X_n \leq m) \\ &= \prod_{i=1}^n P(X_i \leq m) \\ &= \prod_{i=1}^n F_i(m).\end{aligned}\tag{C.19}$$



**Figure C.1:** 2-dimensional representations of sampling probability. **(A)** Probability of sampling as a function of the cost per measurement ( $c$ ) and the number of prospective samples made by the meta-planner. **(B)** Interaction between the action gap and number of actions ( $N$ ), for a fixed cost ( $c = 0.1$ ). The color code is two-dimensional: the hue represents probability of sampling and the saturation proportion of total simulations for the combination of top gap size and  $N$ .

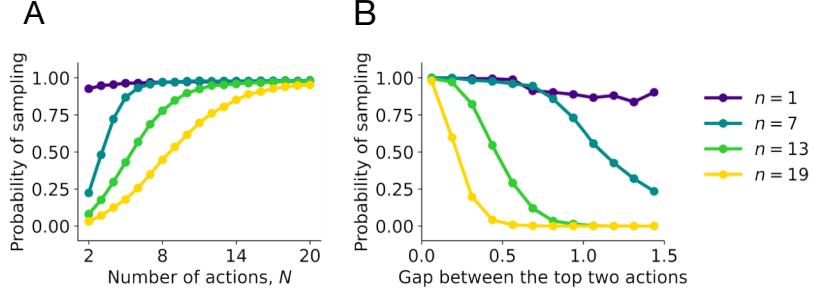
The density of  $M$  is obtained by differentiation:

$$\begin{aligned}
 p_M(m) &= \frac{dF_M}{dm} \\
 &= \sum_{j=1}^n p_j(m) \left( \prod_{i \neq j} F_i(m) \right) \\
 &= \left( \prod_{i=1}^n F_i(m) \right) \sum_{j=1}^n \frac{p_j(m)}{F_j(m)}. \tag{C.20}
 \end{aligned}$$

We use this expression to create the max distribution and to compare with an equivalent sampling solution.

## C.2 MODEL SIMULATIONS

In all model simulations presented in Chapter 4, we ran the model locally by selecting the range of parameters we were interested in. The modifiable parameters were as follows: the number of actions  $N$ , the fixed cost per sample  $c$ , the variances attached to the prospective and retrospective measurements  $\sigma^2$  and  $\sigma_{\text{retro}}^2$  respectively, and the average number of retrospective experiences



**Figure C.2:** Progression of the action gap. **(A)** Probability of sampling as a function of the number of actions available to the agent ( $N$ ). Each line represents a different number of prospective samples having been taken so far in the sampling process, and simulations were run for a fixed cost ( $c = 0.1$ ). **(B)** Same as (A) but for the gap between the top two actions the meta-planner is considering.

per action  $\lambda$ . For all simulations, we drew the underlying true  $Q$ -values for each action from a normal distribution with  $\mu = 0$  and  $\sigma = 1$ , and conducted the analytical computation of the max distribution with 1,000 samples. We also ran enough simulations per analysis so that there was no noise in the results averaged across simulations, typically either 1,000 or 10,000. For the simulations with retrospection, we drew the number of retrospective experiences,  $n_{\text{retro},a}$ , independently per action from a Poisson distribution:

$$n_{\text{retro},a} = \frac{\lambda^{n_{\text{retro},a}} e^{-\lambda}}{n_{\text{retro},a}!} \quad (\text{C.21})$$

where  $\lambda$  is the expected value over the number of retrospective experiences desired.

To visualize the meta-planner's sampling probability more compactly, we computed its predictions as a function of both cost of sampling ( $c$ ), number of actions ( $N$ ), and action gap. In Figure C.1A, we show that deeper planning is beneficial with low costs and when less samples have already been taken. We elected to not include the simulations highlighting the role of cost in Chapter 4.2, since its effect on sampling is straightforward. In Figure C.1B, we replicated our analysis regarding the action gap from the main text to emphasize that sampling is beneficial when more actions are available and thus the action gap is smaller. Note that this analysis also includes the proportion of simulations that the model encountered for each combination of top

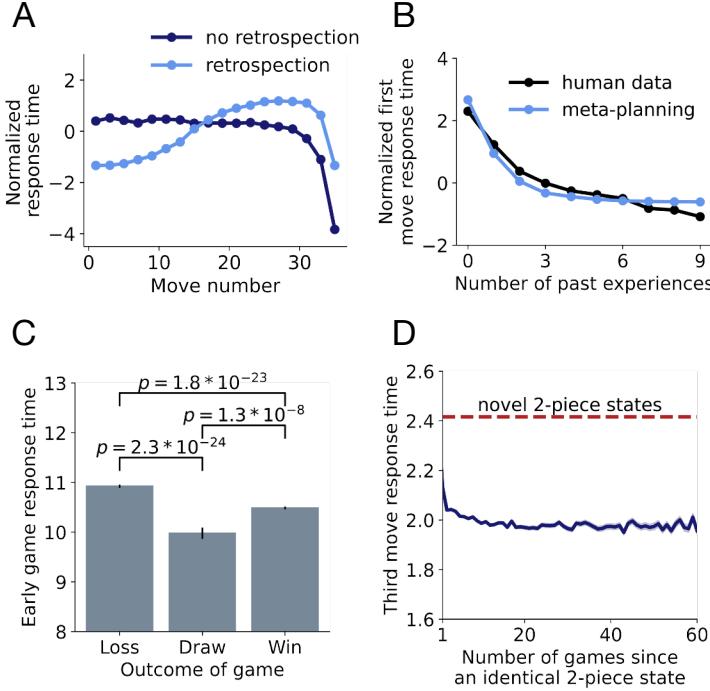
gap and  $N$ . In Figure C.2 we investigated the progression of the action gap, namely that probability of sampling increased with number of alternatives and decreased with gap. This additionally shows how sampling is almost always valuable regardless of these two quantities on the first model iteration, and then shifts to the expected trend as more samples are taken.

### C.3 ADDITIONAL BEHAVIORAL ANALYSES

We conducted a set of additional analyses to show that response times in early stages of a 4-in-a-row game follow patterns predicted by retrospective learning. We first highlight the importance of retrospection in the meta-planner’s qualitative description of human response times as a function of move number. Specifically, we show that removing the exponential decay function for past experiences altogether leads to a monotonically decreasing prediction for response times (Figure C.3A). This is expected in the context of the meta-planner, as less alternatives to consider will typically lead to smaller action gaps and number of samples. For the exponential decay function, we selected a formulation that would roughly mimic human experiences in the task:

$$\lambda = a(1 - r)^i \quad (\text{C.22})$$

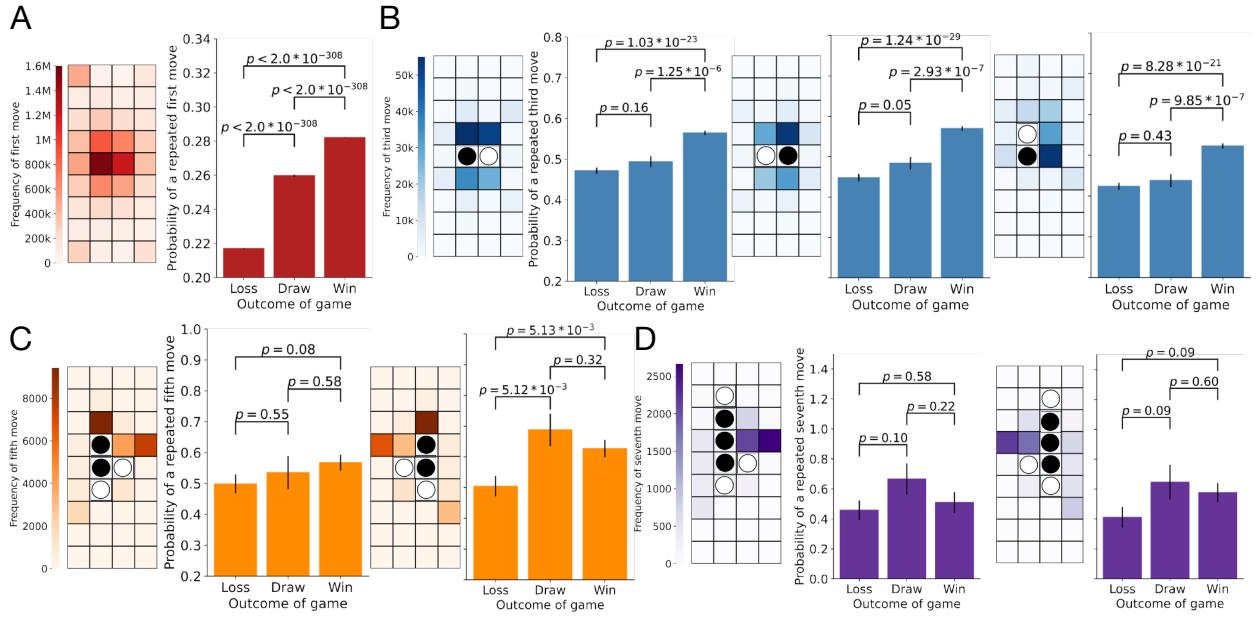
where we fixed the rate of decay  $r$  to 0.25 and the initial value  $a$  to 10 while varying the time interval  $i$  with move number. This results in a quickly decreasing  $\lambda$ , which we use to dictate the number of past experiences that the meta-planner has, that starts at 10 on the first move of the game and decreases to less than 1 within a few moves. We also repeated our analysis which captured the fact that human third move response times are indistinguishable from the meta-planner’s predictions with more experience, validating that it holds for other move numbers (Figure C.3B). In Figure C.3C, we show that user response times across the first 7 moves were, on average, longer after losses rather than wins. Furthermore, we verified that our result



**Figure C.3:** Evidence for retrospective response times in 4-in-a-row. **(A)** Average number of samples the meta-planner makes throughout gameplay with retrospection (blue) and without retrospection (dark blue). As in the main text, the meta-planner is simulated with retrospective experience following a decreasing exponential with move number. All subsequent panels are for all users in the data set. **(B)** Average human response times (black) and number of samples the meta-planner makes (blue) on the first move of gameplay as a function of the number of past experiences. **(C)** Average response times across the first 7 moves of a game directly following a loss, draw, or win. Error bars denote s.e.m. **(D)** Average third move response times as a function of the number of games in the past that the same 2-piece board state occurred (blue) compared to novel 2-piece board states (red). Shading denotes s.e.m.

from the main text showing that third move response times decreased significantly when users encountered repeated 2-piece board states was not solely due to recent memory of encountered states. To do this, we averaged third move response times based on the number of games in the past that the same 2-piece board state occurred, and found that response times were consistent regardless of how long ago a given state had been seen (Figure C.3D). These response times were also drastically lower than for novel 2-piece board states.

The size of our data set allowed us to uncover clear evidence for retrospective decision-making in early-game positions despite using a task with such a large state space where states don't often repeat. We found that users were significantly more likely to repeat their opening moves



**Figure C.4:** Evidence for retrospective decision-making in 4-in-a-row. Each panel contains the probability that all users in the data set that encountered the given board state in subsequent games repeated a move directly after a loss, draw, or win as well as the distribution of the selected moves. Error bars denote s.e.m. The user pieces are in black while the AI pieces are in white. **(A)** The first move (10,875,547 users). **(B)** The third move following the most frequent 2-piece board states (from left to right: 213, 042, 174, 606, and 171,314 users). **(C)** The fifth move following the most frequent 4-piece board states (from left to right: 27, 522 and 25, 452 users). **(D)** The seventh move following the most frequent 6-piece board states (from left to right: 7, 756 and 7, 319 users).

following wins rather than losses, and that these moves were primarily distributed in the center or corners of the board (Figure C.4A). This effect continued on the third move, where users most often elected to play in the center positions closest to the two pieces already on the board (Figure C.4B). On the fifth and seventh moves, however, the proportion of move repetitions based on game outcome were not always significant, varying by specific board position (Figure C.4C-D). These population-wide trends suggest that people make decisions partially based on whether or not an opening strategy was successful in previous games in their first two or three moves, and then begin to utilize alternative strategies in subsequent moves when board positions are more likely to be unique. This set of results motivated the derivation of the meta-planner, as the need for a framework that integrates prospective and retrospective information became apparent.

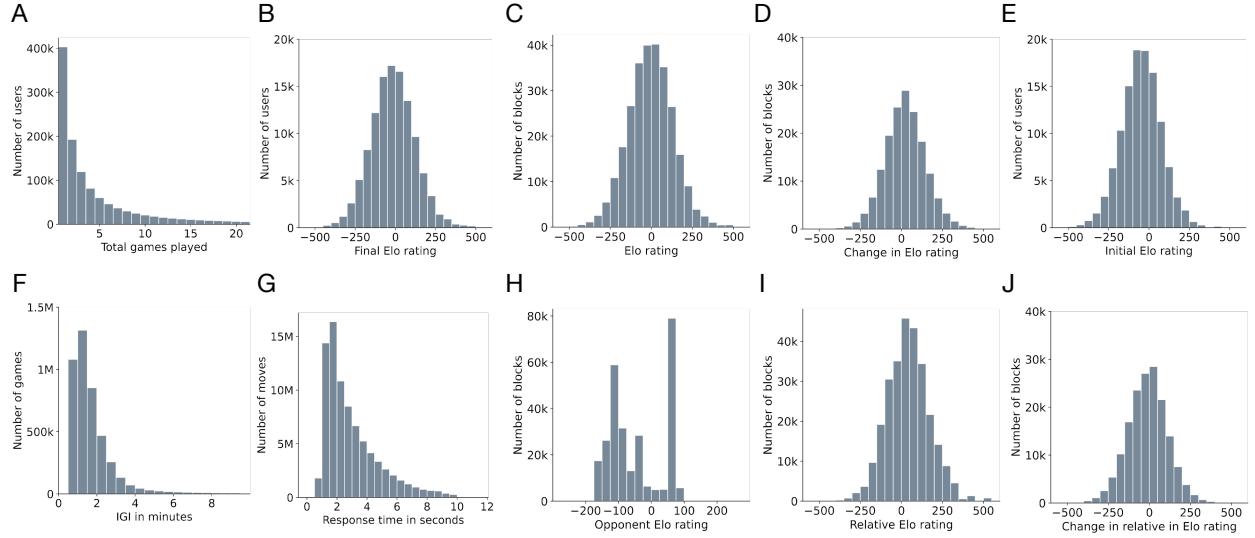
# D | APPENDIX FOR CHAPTER 5

## D.1 METHODS

The analyses in Section 5.1 were conducted for different subsets of the large-scale mobile data set. This is because each analysis was naturally conditioned by a range of number of games played per user, which we summarize in Table D.1. For example, any analysis that required Elo rating estimation must exclude users who played less than 20 games, and any analysis that investigated changes in Elo ratings required a minimum of 40 games played in order to observe at least a single change in ratings from 20 to 40 games of experience. Other analyses are conditioned on specific experience ranges, and in some cases are shown as references for individual users. In Figure D.1, we provide the distribution for each quantity used in our results across the entire data set: total number of games played, final Elo ratings, change in Elo ratings, initial Elo ratings, inter-game intervals, response times, opponent Elo ratings, relative Elo ratings, and change in relative Elo ratings.

For any playing strength analysis, we used a common baseline to compute Elo ratings across all experimental data as before, split into blocks of 20 games. This outputs an Elo rating for each user in every block as well as an estimate of the overall opponents' ability faced by the same user in that block. To compute relative Elo ratings, we simply calculated the difference between these two quantities:

$$R_{\text{relative}} = R_{\text{user}} - R_{\text{Opponent}} \quad (\text{D.1})$$



**Figure D.1:** The distribution for each quantity used for the analyses in Section 5.1. The number of users and range of games played per user for each panel is available in Table D.1. **(A)** Histogram of the total number of games played by users in the data set. **(B)** Same as (A), but for Elo ratings in the final block of gameplay. **(C)** Same as (A), but for overall Elo ratings across all blocks. **(D)** Same as (A), but for change in Elo ratings from block to block. **(E)** Same as (A), but for Elo ratings in the first block of gameplay. **(F)** Same as (A), but for the inter-game interval (IGI) in minutes between every game. **(G)** Same as (A), but for the response time in seconds of every move. **(H)** Same as (A), but for opponent Elo ratings across all blocks. **(I)** Same as (A), but for the difference between user and opponent Elo ratings across all blocks. **(J)** Same as (A), but for the change in difference between user and opponent Elo ratings from block to block.

where  $R$  stands for Elo rating. For the upper bound agent analysis, we computed the exact same joint Elo estimation, but added a pseudo-user who won every game against opponents. To roughly mimic our staircasing procedure, the class of computer agent that the pseudo-user was matched with incremented by 1 up until 7 every 20 games.

Since Elo rating estimation is particularly central to this chapter, we review the fundamental methods that Bayeselo employs here [Hunter 2004]. The Elo formula provides the expected result  $E$  of a game as a function of the rating difference  $D$  between players:

$$E = \frac{1}{1 + 10^{\frac{D}{400}}}. \quad (\text{D.2})$$

The Elo rating system assumes that the underlying strength of a player can be described by a

<b>Figure</b>	<b>Panel</b>	<b>Number of users</b>	<b>Number of games played per user</b>
5.1	B	104,681	20-99
5.2	A	107,769	20-119
5.2	B	10,698	100-499
5.2	C	1	400
5.2	D	104,681	20-99
5.3	A	115,968	20+
5.3	B	115,968	20+
5.3	C	48,156	40+
5.3	D	104,681	20-99
5.4	A	50	20+
5.4	B	115,968	20+
5.4	C	115,968	20+
5.4	D	3,088	100-119
5.4	E	104,681	20-99
5.5	A	107,769	20-119
5.5	B	3,088	100-119
5.5	C	115,968	20+
5.5	D	48,156	40+
5.5	E	104,681	20-99
D.1	A	1,234,844	1+
D.1	B	115,968	20+
D.1	C	115,968	20+
D.1	D	48,156	40+
D.1	E	115,968	20+
D.1	F	1,234,844	1+
D.1	G	1,234,844	1+
D.1	H	115,968	20+
D.1	I	115,968	20+
D.1	J	48,156	40+
D.2	A	4	1000+
D.2	B	115,968	20+
D.2	C	115,968	20+
D.3	A	1,234,844	1+
D.3	B	1,234,844	1+
D.3	C	1,234,844	1+

**Table D.1:** The number of users and range of games played per user for each analysis in Section 5.1 and Appendices D.1 and D.2. We label these values by figure number and panel letter.

single value, and that game results are drawn according to the formula above. The problem is then to estimate the Elo rating of a set of players from the observation of results of their games. The Elo formula above can be reversed to obtain an estimation of the rating difference between two players as a function of the average score. This is traditionally done in two steps:

1. A fixed-point equation is solved iteratively such that the rating of every player is in accordance to the reverse Elo formula. This assumes that an expected result is equal to the average score against an opponent whose Elo rating is equal to the average opponent, and is done under a constraint of a given average Elo over all players.
2. Uncertainty is estimated as the variance of score.

The main flaw of this approach is in the estimation of uncertainty. In other words, the expected result against two players is not equal to the expected result against one single player whose rating is the average of the two players. More concretely, 10 wins and 10 losses against a 1500 Elo opponent should result in less uncertainty than 10 wins against a 500 Elo opponent and 10 losses against a 2500 Elo opponent.

Bayeselo takes a Bayesian approach to ameliorate these problems, and consists of choosing a prior likelihood distribution over Elo ratings and computing a posterior distribution as a function of the observed results. Formally, we can write this as:

$$P(\text{Elos}|\text{Results}) \propto P(\text{Results}|\text{Elos}) \quad (\text{D.3})$$

where the prior  $P(\text{Elos})$  is assumed to be uniform. In order to actually perform this calculation, we need the probability of a win, draw, and loss as a function of the Elo difference for two players,

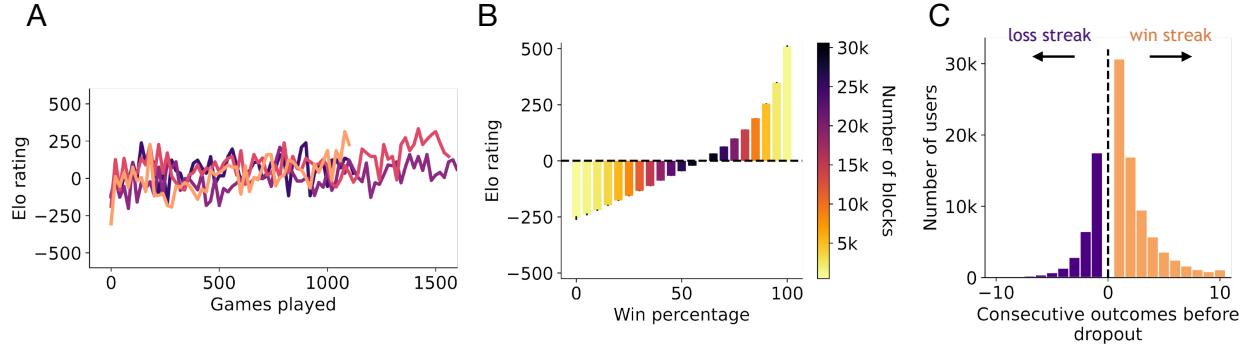
black and white:

$$\begin{aligned}
 P(\text{White wins}) &= f(\text{Elo}_{\text{black}} - \text{Elo}_{\text{white}} - \text{Elo}_{\text{advantage}} + \text{Elo}_{\text{draw}}) \\
 P(\text{Black wins}) &= f(\text{Elo}_{\text{white}} - \text{Elo}_{\text{black}} - \text{Elo}_{\text{advantage}} + \text{Elo}_{\text{draw}}) \\
 P(\text{Draw}) &= 1 - P(\text{White wins}) - P(\text{Black wins})
 \end{aligned} \tag{D.4}$$

where  $f$  is the Elo formula.  $\text{Elo}_{\text{advantage}}$  is the advantage of playing first while  $\text{Elo}_{\text{draw}}$  indicates how likely draws are. These two quantities have been estimated empirically with 95% confidence intervals as  $32.8 \pm 4.0$  and  $97.3 \pm 2.0$  respectively. Bayeselo then finds the maximum-likelihood ratings using a minorization-maximization algorithm. The difference in the ratings between two players serves as a predictor of the outcome of a match. For example, two players with equal ratings who play against each other are expected to score an equal number of wins. A player whose rating is 100 points greater than their opponent's is expected to score 64%, and if the difference is 200 points then the expected score for the stronger player is 76%.

## D.2 CONTROL AND VALIDATION ANALYSES

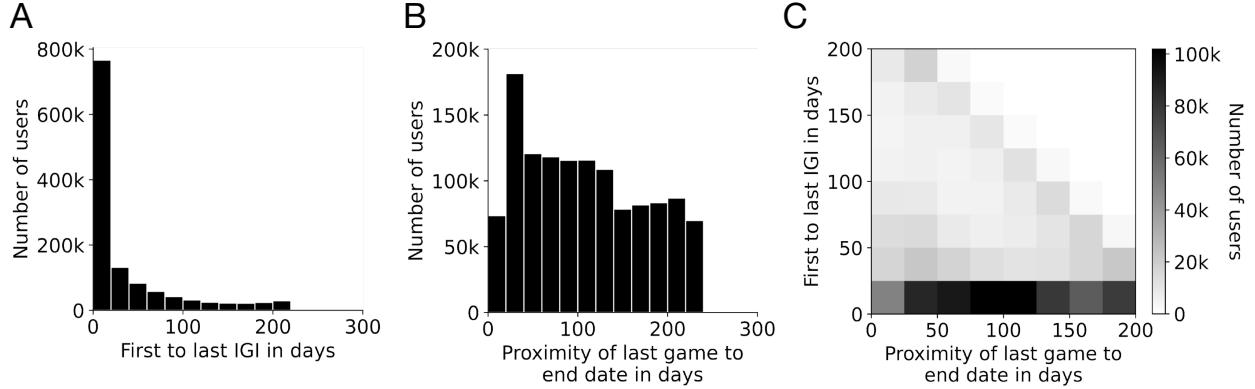
To validate the use of Elo ratings as a measure of playing strength in 4-in-a-row, we conducted a number of analyses. In Figure D.2A, we show Elo ratings over time for 4 users very experienced users who had played at least 1,000 games each. Since we are interested in characterizing population level learning, we primarily visualize trajectories averaged across the population. Individual learning trajectories are noisy but still increase gradually as more games are played, an assumption that carries over into our model. We also found that estimated Elo ratings within a given block and win percentage within that same block were highly correlated ( $\rho = 0.708$ , Figure D.2B). This is expected, as users should be stronger players the more games they win. We further conditioned this on number of blocks that had each combination of Elo rating and win percent-



**Figure D.2:** Analyses to validate the use of Elo ratings as a measure of task performance. The number of users and range of games played per user for each panel is available in Table D.1. **(A)** Individual learning trajectories for 4 experienced users who played at least 1,000 games. **(B)** Histogram of the relationship between Elo ratings and win percentage for every block in the data set. **(C)** Histogram of the number of consecutive wins (orange) or losses (purple) for each user before dropout.

age, highlighting that extremely low or high values occurred much less often than values closer to 0 Elo rating or a win rate of 50%. This further validates that users were generally matched against opponents of equal skill level where they won about half of their games. Finally, we revisited the idea of a peak-end rule with results rather than ratings. Namely, we computed the number of consecutive wins and consecutive losses leading up to each user’s final game (Figure D.2C). This analysis suggested that users more often quit after a comparable number of wins as opposed to losses. Additionally, a number of people in the data set dropped out after larger wins streaks up to 10, which didn’t occur for losses. This corroborates our finding that people were more likely to quit after large increases in their playing strength.

Another important control was to ensure that our results weren’t biased by the fact that subset of data we analyzed has an arbitrary cutoff. In principle, many users could have still been actively playing games after April 2019, meaning that their gameplay masquerades as dropout when it is not. To check that this wasn’t the case for the majority of our data set, we computed two quantities: the inter-game interval (IGI) between the first and last game of play (Figure D.3A) and the proximity of the last recorded game to the end date of data collection (Figure D.3B). Ideally, we would find that the time from people’s last game until the end of data collection is, for most users,



**Figure D.3:** Analyses to control for edge effects in dropout. The number of users and range of games played per user for each panel is available in Table D.1. **(A)** Histogram of the inter-game interval (IGI) in days between the first and last game by users in the data set. **(B)** Histogram of the time in days between each user’s last game and the final day of data collection. **(C)** 2-dimensional histogram of the intersection between the quantities in (A) and (B).

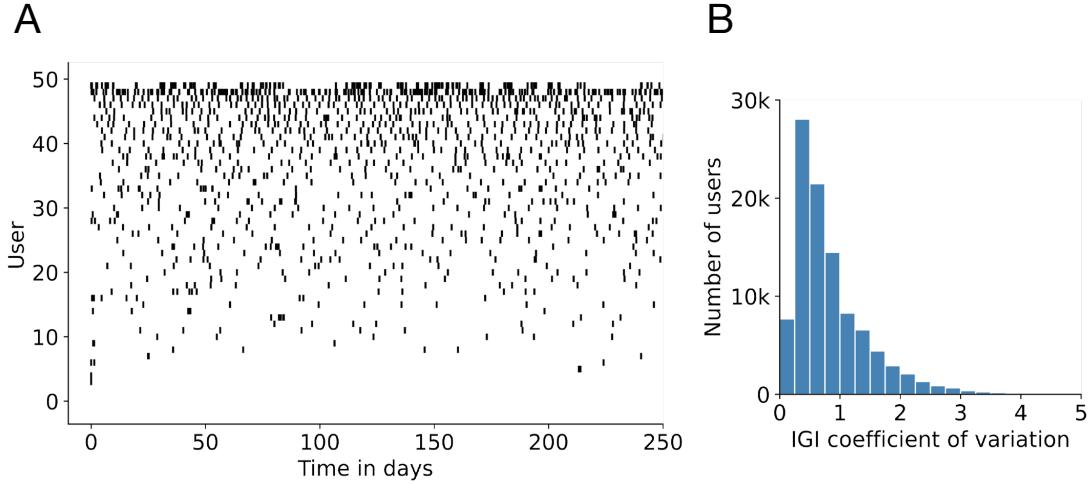
much longer than the entire duration of their playing history. Indeed, this was overwhelmingly true throughout the data set (Figure D.3C). Given the large number of users in each analysis, this control suggested that most of the behavior we consider to be dropout is not corrupted by significant edge effects.

### D.3 MODEL SIMULATIONS

To implement our model of task engagement and performance, we use the equations in Section 5.2. This requires a prior on Elo ratings, which we formalize as a normal distribution with  $\mu = -50$  and  $\sigma = 125$  approximating the initial Elo rating distribution in Figure D.1E. To predict the elapsed time between games for each user, we use a Weibull distribution:

$$a = \frac{k}{\alpha} \left( \frac{x}{\alpha} \right)^{k-1} e^{-\frac{x}{\alpha}} \quad (\text{D.5})$$

where  $k$  is the shape parameter and  $\alpha$  is the scale parameter of the distribution. We assume the shape and scale parameters vary per player and are drawn from two separate normal distribu-



**Figure D.4:** The model reproduces the time course of people’s gameplay. **(A)** Model simulations replicating the raster plot for the time course of play for 50 randomly selected pseudo-users from Figure 5.4A. **(B)** Same as (A), but for the coefficient of variation (CV), or the ratio of the standard deviation and the mean of the inter-game interval (IGI), visualized as a histogram from Figure 5.4B.

tions, both with  $\mu = 0$  and the former with  $\sigma = 2$  and the latter with  $\sigma = 20$ . Similarly, we draw individual learning rates from a normal with  $\mu = 0$  and  $\sigma = 0.01$ . Since none of these values can be negative, we truncate the distribution by taking the absolute value of each draw. The remaining fixed parameters that we use to run the model forward are  $\lambda_{\text{forgetting}} = 0.001$ ,  $R_0 = -600$ ,  $R_\infty = 600$ , and  $\sigma_{\text{noise}} = 10$ . The baseline and ceiling ratings are based on the upper bound agent as well as the observed minimum and maximum empirical Elo ratings from Figure D.1C. The  $\lambda$  and noise parameters are selected to provide a reasonable fit to data after extensive testing. We ran the model simulations locally per analysis for 100,000 pseudo-users. To compute game outcomes, we could use another prior to approximate the relative Elo rating distribution in Figure D.1I. The output of a draw from this distribution would determine the difference in ratings between a user and their opponent. Then, the Elo equation would be used to calculate the outcome of a game between the two players. In the future, we will use maximum likelihood estimation to obtain parameters by fitting the model to individual games and elapsed times.

In Figure D.4 we validate that the model simulations can capture the physical time between games. To do so, we show a raster plot generated by pseudo-users as well as the distribution

of the coefficient of variation (CV) across the inter-game interval (IGI) for all pseudo-users. This provides evidence that our model not only accounts for the main learning and motivation results, but also the full time course of people's playing history.

## REFERENCES

- Aarseth, E. (2014). I fought the law: Transgressive play and the implied player. In *From literature to cultural literacy*, pages 180–188. Springer.
- Acerbi, L. and Ma, W. J. (2017). Practical bayesian optimization for model fitting with bayesian adaptive direct search. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1834–1844.
- Agarwal, T., Burghardt, K., and Lerman, K. (2017). On quitting: Performance and practice in online game play. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, pages 452–455.
- Agrawal, M., Peterson, J. C., and Griffiths, T. L. (2020). Scaling up psychology via scientific regret minimization. *Proceedings of the National Academy of Sciences*, 117(16):8825–8835.
- Akam, T., Costa, R., and Dayan, P. (2015). Simple plans or sophisticated habits? state, transition and learning interactions in the two-step task. *PLoS computational biology*, 11(12):e1004648.
- Akam, T., Rodrigues-Vaz, I., Marcelo, I., Zhang, X., Pereira, M., Oliveira, R. F., Dayan, P., and Costa, R. M. (2021). The anterior cingulate cortex predicts future states to mediate model-based action selection. *Neuron*, 109(1):149–163.
- Akata, E., Schulz, L., Coda-Forno, J., Oh, S. J., Bethge, M., and Schulz, E. (2023). Playing repeated games with large language models. *arXiv preprint arXiv:2305.16867*.
- Allen, K. R., Brändle, F., Botvinick, M., Fan, J., Gershman, S. J., Griffiths, T. L., Hartshorne, J., Hauser, T. U., Ho, M. K., de Leeuw, J., et al. (2023). Using games to understand the mind.
- Allen, K. R., Smith, K. A., and Tenenbaum, J. B. (2020). Rapid trial-and-error learning with simulation supports flexible tool use and physical reasoning. *Proceedings of the National Academy of Sciences*, 117(47):29302–29310.
- Anderson, A., Kleinberg, J., and Mullainathan, S. (2017). Assessing human error against a benchmark of perfection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 11(4):1–25.
- Anderson, J. R. (2013). *The adaptive character of thought*. Psychology Press.

- Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256.
- Bakkes, S. C., Spronck, P. H., and Van Den Herik, H. J. (2009). Opponent modelling for case-based adaptive game ai. *Entertainment Computing*, 1(1):27–37.
- Barnby, J. M., Raihani, N., and Dayan, P. (2022). Knowing me, knowing you: Interpersonal similarity improves predictive accuracy and reduces attributions of harmful intent. *Cognition*, 225:105098.
- Battleday, R. M., Peterson, J. C., and Griffiths, T. L. (2020). Capturing human categorization of natural images by combining deep networks and cognitive models. *Nature communications*, 11(1):1–14.
- Bellemare, M. G., Ostrovski, G., Guez, A., Thomas, P., and Munos, R. (2016). Increasing the action gap: New operators for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.
- Bellman, R. (1966). Dynamic programming. *Science*, 153(3731):34–37.
- Bengio, Y. et al. (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.
- Bilalić, M., Langner, R., Erb, M., and Grodd, W. (2010). Mechanisms and neural basis of object and pattern recognition: a study with chess experts. *Journal of Experimental Psychology: General*, 139(4):728.
- Billings, D., Papp, D., Schaeffer, J., and Szafron, D. (1998). Opponent modeling in poker. *Aaaai/iaai*, 493(499):105.
- Bizo, L. A., Chu, J. Y., Sanabria, F., and Killeen, P. R. (2006). The failure of weber’s law in time perception and production. *Behavioural processes*, 71(2-3):201–210.
- Bonet, B. and Geffner, H. (2001). Planning as heuristic search. *Artificial Intelligence*, 129(1-2):5–33.
- Botvinick, M. and Toussaint, M. (2012). Planning as inference. *Trends in cognitive sciences*, 16(10):485–488.
- Brändle, F., Binz, M., and Schulz, E. (2021). Exploration beyond bandits. *The drive for knowledge: The science of human information seeking*, pages 147–168.
- Brändle, F., Stocks, L. J., Tenenbaum, J., Gershman, S. J., and Schulz, E. (2022). Intrinsically motivated exploration as empowerment.
- Brown, N. and Sandholm, T. (2019). Superhuman ai for multiplayer poker. *Science*, 365(6456):885–890.

- Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012). A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43.
- Busemeyer, J. R. and Diederich, A. (2010). *Cognitive modeling*. Sage.
- Calderwood, R., Klein, G. A., and Crandall, B. W. (1988). Time pressure, skill, and move quality in chess. *The American journal of psychology*, pages 481–493.
- Callaway, F., Jain, Y. R., van Opheusden, B., Das, P., Iwama, G., Gul, S., Krueger, P. M., Becker, F., Griffiths, T. L., and Lieder, F. (2022a). Leveraging artificial intelligence to improve people’s planning strategies. *Proceedings of the National Academy of Sciences*, 119(12):e2117432119.
- Callaway, F., Rangel, A., and Griffiths, T. L. (2021). Fixation patterns in simple choice reflect optimal information sampling. *PLoS computational biology*, 17(3):e1008863.
- Callaway, F., van Opheusden, B., Gul, S., Das, P., Krueger, P. M., Griffiths, T. L., and Lieder, F. (2022b). Rational use of cognitive resources in human planning. *Nature Human Behaviour*, 6(8):1112–1125.
- Campbell, M., Hoane Jr, A. J., and Hsu, F.-h. (2002). Deep blue. *Artificial intelligence*, 134(1-2):57–83.
- Campitelli, G. and Gobet, F. (2004). Adaptive expert decision making: Skilled chess players search more and deeper. *ICGA Journal*, 27(4):209–216.
- Carroll, M., Shah, R., Ho, M. K., Griffiths, T., Seshia, S., Abbeel, P., and Dragan, A. (2019). On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32.
- Chabris, C. F. and Hearst, E. S. (2003). Visualization, pattern recognition, and forward search: Effects of playing speed and sight of the position on grandmaster chess errors. *Cognitive Science*, 27(4):637–648.
- Chapelle, O. and Li, L. (2011). An empirical evaluation of thompson sampling. *Advances in neural information processing systems*, 24.
- Charness, N. (1989). Expertise in chess and bridge. *Complex information processing: The impact of Herbert A. Simon*, pages 183–208.
- Charness, N., Tuffiash, M., Krampe, R., Reingold, E., and Vasyukova, E. (2005). The role of deliberate practice in chess expertise. *Applied Cognitive Psychology*, 19(2):151–165.
- Chase, W. G. and Simon, H. A. (1973). Perception in chess. *Cognitive psychology*, 4(1):55–81.
- Chen, H., Chang, H. J., and Howes, A. (2021). Apparently irrational choice as optimal sequential decision making. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 792–800.

- Chen, X., Starke, S. D., Baber, C., and Howes, A. (2017). A cognitive model of how people make decisions through interaction with visual displays. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 1205–1216.
- Chinchalkar, S. (1996). An upper bound for the number of reachable positions. *ICGA Journal*, 19(3):181–183.
- Clark, C. and Storkey, A. (2015). Training deep convolutional neural networks to play go. In *International conference on machine learning*, pages 1766–1774. PMLR.
- Cockburn, A., Quinn, P., and Gutwin, C. (2015). Examining the peak-end effects of subjective experience. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 357–366.
- Cohen, J. D., McClure, S. M., and Yu, A. J. (2007). Should i stay or should i go? how the human brain manages the trade-off between exploitation and exploration. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1481):933–942.
- Collins, A. G. and Shenhav, A. (2022). Advances in modeling learning and decision-making in neuroscience. *Neuropsychopharmacology*, 47(1):104–118.
- Correa, C. G., Ho, M. K., Callaway, F., Daw, N. D., and Griffiths, T. L. (2023). Humans decompose tasks by trading off utility and computational cost. *PLOS Computational Biology*, 19(6):e1011087.
- Corsi, P. M. (1972). Human memory and the medial temporal region of the brain.
- Costa-Gomes, M., Crawford, V. P., and Broseta, B. (2001). Cognition and behavior in normal-form games: An experimental study. *Econometrica*, 69(5):1193–1235.
- Coughlan, G., Coutrot, A., Khondoker, M., Minihane, A.-M., Spiers, H., and Hornberger, M. (2019). Toward personalized cognitive diagnostics of at-genetic-risk alzheimer’s disease. *Proceedings of the National Academy of Sciences*, 116(19):9285–9292.
- Coutrot, A., Silva, R., Manley, E., de Cothi, W., Sami, S., Bohbot, V. D., Wiener, J. M., Hölscher, C., Dalton, R. C., Hornberger, M., et al. (2018). Global determinants of navigation ability. *Current Biology*, 28(17):2861–2866.
- Cowan, N. (2017). The many faces of working memory and short-term storage. *Psychonomic bulletin & review*, 24:1158–1170.
- Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202.
- Dasgupta, I., Schulz, E., Goodman, N. D., and Gershman, S. J. (2018). Remembrance of inferences past: Amortization in human hypothesis generation. *Cognition*, 178:67–81.

- Daw, N. D. et al. (2011a). Trial-by-trial data analysis using computational models. *Decision making, affect, and learning: Attention and performance XXIII*, 23(1).
- Daw, N. D., Gershman, S. J., Seymour, B., Dayan, P., and Dolan, R. J. (2011b). Model-based influences on humans' choices and striatal prediction errors. *Neuron*, 69(6):1204–1215.
- Daw, N. D., Niv, Y., and Dayan, P. (2005). Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature neuroscience*, 8(12):1704–1711.
- Daw, N. D., O'doherty, J. P., Dayan, P., Seymour, B., and Dolan, R. J. (2006). Cortical substrates for exploratory decisions in humans. *Nature*, 441(7095):876–879.
- De Groot, A. D. (2014). *Thought and choice in chess*, volume 4. Walter de Gruyter GmbH & Co KG.
- Dechter, R. and Pearl, J. (1985). Generalized best-first search strategies and the optimality of a. *Journal of the ACM (JACM)*, 32(3):505–536.
- Dezfouli, A., Griffiths, K., Ramos, F., Dayan, P., and Balleine, B. W. (2019). Models that learn how humans learn: the case of decision-making and its disorders. *PLoS computational biology*, 15(6):e1006903.
- Dickinson, A. (1985). Actions and habits: the development of behavioural autonomy. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 308(1135):67–78.
- Dolan, R. J. and Dayan, P. (2013). Goals and habits in the brain. *Neuron*, 80(2):312–325.
- Donner, Y. and Hardy, J. L. (2015). Piecewise power laws in individual learning curves. *Psychonomic Bulletin & Review*, 22:1308–1319.
- Dubey, R., Agrawal, P., Pathak, D., Griffiths, T. L., and Efros, A. A. (2018). Investigating human priors for playing video games. *arXiv preprint arXiv:1802.10217*.
- Duckworth, A. and Gross, J. J. (2014). Self-control and grit: Related but separable determinants of success. *Current directions in psychological science*, 23(5):319–325.
- Eckstein, M. K., Summerfield, C., Daw, N. D., and Miller, K. J. (2023). Predictive and interpretable: Combining artificial neural networks and classic cognitive models to understand human learning and decision making. *bioRxiv*, pages 2023–05.
- Elo, A. E. (1978). *The rating of chessplayers, past and present*. Arco Pub.
- Éltető, N. and Dayan, P. (2023). Habits of mind: Reusing action sequences for efficient planning. *arXiv preprint arXiv:2306.05298*.
- Ericsson, K. A. and Smith, J. (1991). *Toward a general theory of expertise: Prospects and limits*. Cambridge University Press.

- (FAIR)†, M. F. A. R. D. T., Bakhtin, A., Brown, N., Dinan, E., Farina, G., Flaherty, C., Fried, D., Goff, A., Gray, J., Hu, H., et al. (2022). Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074.
- Farahmand, A.-m. (2011). Action-gap phenomenon in reinforcement learning. *Advances in Neural Information Processing Systems*, 24.
- Farzan, R., DiMicco, J. M., Millen, D. R., Dugan, C., Geyer, W., and Brownholtz, E. A. (2008). Results from deploying a participation incentive mechanism within the enterprise. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 563–572.
- Feher da Silva, C. and Hare, T. A. (2020). Humans primarily use model-based inference in the two-stage task. *Nature Human Behaviour*, 4(10):1053–1066.
- Finnsson, H. and Björnsson, Y. (2008). Simulation-based approach to general game playing. In *Aaai*, volume 8, pages 259–264.
- Flesch, T., Juechems, K., Dumbalska, T., Saxe, A., and Summerfield, C. (2021). Rich and lazy learning of task representations in brains and neural networks. *BioRxiv*, pages 2021–04.
- Frömer, R., Lin, H., Dean Wolf, C., Inzlicht, M., and Shenhav, A. (2021). Expectations of reward and efficacy guide cognitive control allocation. *Nature communications*, 12(1):1030.
- Geana, A., Wilson, R., Daw, N. D., and Cohen, J. (2016). Boredom, information-seeking and exploration. In *CogSci*.
- Gershman, S. J. (2018). Deconstructing the human algorithms for exploration. *Cognition*, 173:34–42.
- Gershman, S. J. (2019). Uncertainty and exploration. *Decision*, 6(3):277.
- Getty, D. J. (1975). Discrimination of short temporal intervals: A comparison of two models. *Perception & psychophysics*, 18(1):1–8.
- Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 41(2):148–164.
- Gläscher, J., Daw, N., Dayan, P., and O’Doherty, J. P. (2010). States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. *Neuron*, 66(4):585–595.
- Gobet, F. (1997). A pattern-recognition theory of search in expert problem solving. *Thinking & Reasoning*, 3(4):291–313.
- Gobet, F. and Jansen, P. (1994). Towards a chess program based on a model of human memory. *Advances in computer chess*, 7:35–60.

- Gobet, F., Retschitzki, J., and de Voogt, A. (2004). *Moves in mind: The psychology of board games*. Psychology Press.
- Gobet, F. and Simon, H. A. (1998). Expert chess memory: Revisiting the chunking hypothesis. *Memory*, 6(3):225–255.
- Griffiths, T. L., Chater, N., Kemp, C., Perfors, A., and Tenenbaum, J. B. (2010). Probabilistic models of cognition: Exploring representations and inductive biases. *Trends in cognitive sciences*, 14(8):357–364.
- Griffiths, T. L., Lieder, F., and Goodman, N. D. (2015). Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. *Topics in cognitive science*, 7(2):217–229.
- Groman, S. M., Rich, K. M., Smith, N. J., Lee, D., and Taylor, J. R. (2018). Chronic exposure to methamphetamine disrupts reinforcement-based decision making in rats. *Neuropsychopharmacology*, 43(4):770–780.
- Grondin, S. (1992). Production of time intervals from segmented and nonsegmented inputs. *Perception & Psychophysics*, 52(3):345–350.
- Gureckis, T. M. and Markant, D. B. (2012). Self-directed learning: A cognitive and computational perspective. *Perspectives on Psychological Science*, 7(5):464–481.
- Gutwin, C., Rooke, C., Cockburn, A., Mandryk, R. L., and Lafreniere, B. (2016). Peak-end effects on player experience in casual games. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 5608–5619.
- Gweon, H. (2021). Inferential social learning: Cognitive foundations of human social learning and teaching. *Trends in Cognitive Sciences*, 25(10):896–910.
- Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Pfaff, T., Weber, T., Buesing, L., and Battaglia, P. W. (2019). Combining q-learning and search with amortized value estimates. *arXiv preprint arXiv:1912.02807*.
- Hay, N., Russell, S., Tolpin, D., and Shimony, S. E. (2014). Selecting computations: Theory and applications. *arXiv preprint arXiv:1408.2048*.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Heathcote, A., Brown, S., and Mewhort, D. J. (2000). The power law repealed: The case for an exponential law of practice. *Psychonomic bulletin & review*, 7(2):185–207.
- Ho, M. K., Abel, D., Correa, C. G., Littman, M. L., Cohen, J. D., and Griffiths, T. L. (2022a). People construct simplified mental representations to plan. *Nature*, 606(7912):129–136.

- Ho, M. K., Saxe, R., and Cushman, F. (2022b). Planning with theory of mind. *Trends in Cognitive Sciences*, 26(11):959–971.
- Holdaway, C. and Vul, E. (2021). Risk-taking in adversarial games: What can 1 billion online chess games tell us? In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 43.
- Holding, D. H. (1989a). Counting backward during chess move choice. *Bulletin of the Psychonomic Society*, 27(5):421–424.
- Holding, D. H. (1989b). Evaluation factors in human tree search. *The American Journal of Psychology*, 102(1):103–108.
- Holding, D. H. (1992). Theories of chess skill. *Psychological Research*, 54(1):10–16.
- Holding, D. H. (2021). *The psychology of chess skill*. Routledge.
- Huang, J., Velarde, I., Ma, W. J., and Baldassano, C. (2023). Schema-based predictive eye movements support sequential memory encoding. *Elife*, 12:e82599.
- Huang, J., Yan, E., Cheung, G., Nagappan, N., and Zimmermann, T. (2017). Master maker: Understanding gaming skill through practice and habit from gameplay behavior. *Topics in cognitive science*, 9(2):437–466.
- Hunt, L., Daw, N., Kaanders, P., MacIver, M., Mugan, U., Procyk, E., Redish, A., Russo, E., Scholl, J., Stachenfeld, K., et al. (2021). Formalizing planning and information search in naturalistic decision-making. *Nature neuroscience*, 24(8):1051–1064.
- Hunter, D. R. (2004). Mm algorithms for generalized bradley-terry models. *The annals of statistics*, 32(1):384–406.
- Huys, Q. J., Lally, N., Faulkner, P., Eshel, N., Seifritz, E., Gershman, S. J., Dayan, P., and Roiser, J. P. (2015). Interplay of approximate planning strategies. *Proceedings of the National Academy of Sciences*, 112(10):3098–3103.
- Huys, Q. J. M., Eshel, N., O’Nions, E., Sheridan, L., Dayan, P., and Roiser, J. P. (2012). Bonsai trees in your head: How the pavlovian system sculpts goal-directed choices by pruning decision trees. *PLOS Computational Biology*.
- Jang, A. I., Sharma, R., and Drugowitsch, J. (2021). Optimal policy for attention-modulated decisions explains human fixation behavior. *Elife*, 10:e63436.
- Ji-An, L., Benna, M. K., and Mattar, M. G. (2023). Automatic discovery of cognitive strategies with tiny recurrent neural networks. *bioRxiv*, pages 2023–04.
- Johnson, A. and Redish, A. D. (2007). Neural ensembles in ca3 transiently encode paths forward of the animal at a decision point. *Journal of Neuroscience*, 27(45):12176–12189.

- Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134.
- Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.
- Kirchner, W. K. (1958). Age differences in short-term retention of rapidly changing information. *Journal of experimental psychology*, 55(4):352.
- Kleiman-Weiner, M., Ho, M. K., Austerweil, J. L., Littman, M. L., and Tenenbaum, J. B. (2016). Coordinate to cooperate or compete: abstract goals and joint intentions in social interaction. In *CogSci*.
- Kolling, N., Scholl, J., Chekroud, A., Trier, H. A., and Rushworth, M. F. (2018). Prospection, perseverance, and insight in sequential behavior. *Neuron*, 99(5):1069–1082.
- Kool, W., Cushman, F. A., and Gershman, S. J. (2016). When does model-based control pay off? *PLoS computational biology*, 12(8):e1005090.
- Kool, W., Gershman, S. J., and Cushman, F. (2017). Cost-benefit arbitration between multiple reinforcement-learning systems. *Psychological Science*, 28:1321–1333.
- Kriegeskorte, N., Mur, M., and Bandettini, P. A. (2008). Representational similarity analysis—connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, page 4.
- Krusche, M. J., Schulz, E., Guez, A., and Speekenbrink, M. (2018). Adaptive planning in human search. *bioRxiv*, page 268938.
- Kumar, A. A., Steyvers, M., and Balota, D. A. (2021). Semantic memory search and retrieval in a novel cooperative word game: A comparison of associative and distributional semantic models. *Cognitive Science*, 45(10):e13053.
- Kumar, S., Correa, C. G., Dasgupta, I., Marjeh, R., Hu, M. Y., Hawkins, R., Cohen, J. D., Narasimhan, K., Griffiths, T., et al. (2022). Using natural language and program abstractions to instill human inductive biases in machines. *Advances in Neural Information Processing Systems*, 35:167–180.
- Kuperwajs, I. and Ma, W. J. (2021). Planning to plan: a bayesian model for optimizing the depth of decision tree search. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 43.
- Kuperwajs, I. and Ma, W. J. (2022). A joint analysis of dropout and learning functions in human decision-making with massive online data. In *Proceedings of the 44th Annual Conference of the Cognitive Science Society*.
- Kuperwajs, I., Schuett, H. H., and Ma, W. J. (2022). Improving a model of human planning via large-scale data and deep neural networks. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

- Kuperwajs, I., Schütt, H. H., and Ma, W. J. (2023). Using deep neural networks as a guide for modeling human planning. *Scientific Reports*, 13(1):20269.
- Kuperwajs, I., van Opheusden, B., and Ma, W. J. (2019). Prospective planning and retrospective learning in a large-scale combinatorial game. In *2019 Conference on Cognitive Computational Neuroscience*, pages 13–16.
- Lai, C., Tanaka, S., Harris, T. D., and Lee, A. K. (2023). Volitional activation of remote place representations with a hippocampal brain–machine interface. *Science*, 382(6670):566–573.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.
- Lee, C.-S., Müller, M., and Teytaud, O. (2010). Special issue on monte carlo techniques and computer go. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(4):225–228.
- Lee, D. and Seo, H. (2016). Neural basis of strategic decision making. *Trends in neurosciences*, 39(1):40–48.
- Lee, M. D., Zhang, S., Munro, M., and Steyvers, M. (2011). Psychological models of human and optimal performance in bandit problems. *Cognitive Systems Research*, 12(2):164–174.
- Lee, S. W., Shimojo, S., and O’Doherty, J. P. (2014). Neural computations underlying arbitration between model-based and model-free learning. *Neuron*, 81(3):687–699.
- Leonard, J. A., Kleiman-Weiner, M., Lee, Y., Tenenbaum, J., and Schulz, L. (2017). Preschoolers and infants calibrate persistence from adult models. In *CogSci*.
- Levy, D. J. and Glimcher, P. W. (2012). The root of all value: a neural common currency for choice. *Current opinion in neurobiology*, 22(6):1027–1038.
- Lieder, F. and Griffiths, T. L. (2017). Strategy selection as rational metareasoning. *Psychological review*, 124(6):762.
- Linhares, A., Freitas, A. E. T., Mendes, A., and Silva, J. S. (2012). Entanglement of perception and reasoning in the combinatorial game of chess: Differential errors of strategic reconstruction. *Cognitive Systems Research*, 13(1):72–86.
- Ma, I., Phaneuf, C., van Opheusden, B., Ma, W. J., and Hartley, C. (2022a). The component processes of complex planning follow distinct developmental trajectories.
- Ma, I., Westhoff, B., and van Duijvenvoorde, A. (2022b). Uncertainty about others’ trustworthiness increases during adolescence and guides social information sampling. *Scientific Reports*, 12(1):7634.
- Malone, T. (1981). What makes computer games fun? In *Proceedings of the Joint Conference on Easier and More Productive Use of Computer Systems.(Part-II): Human Interface and the User Interface-Volume 1981*, page 143.

- Marr, D. (2010). *Vision: A computational investigation into the human representation and processing of visual information*. MIT press.
- Mastrogiuseppe, C. and Moreno-Bote, R. (2022). Deep imagination is a close to optimal policy for planning in large decision trees under limited resources. *Scientific reports*, 12(1):10411.
- Mattar, M. G. and Daw, N. D. (2018). Prioritized memory access explains planning and hippocampal replay. *Nature neuroscience*, 21(11):1609–1617.
- Mattar, M. G. and Lengyel, M. (2022). Planning in the brain. *Neuron*, 110(6):914–934.
- McGrath, T., Kapishnikov, A., Tomašev, N., Pearce, A., Wattenberg, M., Hassabis, D., Kim, B., Paquet, U., and Kramnik, V. (2022). Acquisition of chess knowledge in alphazero. *Proceedings of the National Academy of Sciences*, 119(47):e2206625119.
- Miller, K. J., Botvinick, M. M., and Brody, C. D. (2017). Dorsal hippocampus contributes to model-based planning. *Nature neuroscience*, 20(9):1269–1276.
- Miller, K. J., Eckstein, M., Botvinick, M. M., and Kurth-Nelson, Z. (2023). Cognitive model discovery via disentangled rnns. *bioRxiv*, pages 2023–06.
- Miller, K. J. and Venditto, S. J. C. (2021). Multi-step planning in the brain. *Current Opinion in Behavioral Sciences*, 38:29–39.
- Miron-Shatz, T. (2009). Evaluating multiepisode events: Boundary conditions for the peak-end rule. *Emotion*, 9(2):206.
- Mitroff, S. R., Biggs, A. T., Adamo, S. H., Dowd, E. W., Winkle, J., and Clark, K. (2015). What can 1 billion trials tell us about visual search? *Journal of experimental psychology: human perception and performance*, 41(1):1.
- Molinaro, G. and Collins, A. G. (2023). A goal-centric outlook on learning. *Trends in Cognitive Sciences*.
- Montague, P. R., Dayan, P., and Sejnowski, T. J. (1996). A framework for mesencephalic dopamine systems based on predictive hebbian learning. *Journal of neuroscience*, 16(5):1936–1947.
- Moore, A. W. and Atkeson, C. G. (1993). Prioritized sweeping: Reinforcement learning with less data and less time. *Machine learning*, 13:103–130.
- Moreno-Bote, R., Ramírez-Ruiz, J., Drugowitsch, J., and Hayden, B. Y. (2020). Heuristics and optimal solutions to the breadth–depth dilemma. *Proceedings of the National Academy of Sciences*, 117(33):19799–19808.
- Newell, A. et al. (1980). Mechanisms of skill acquisition and the law of practice.
- Newell, A., Shaw, J. C., and Simon, H. A. (1959). Report on a general problem solving program. In *IFIP congress*, volume 256, page 64. Pittsburgh, PA.

- Newell, A. and Simon, H. (1956). The logic theory machine—a complex information processing system. *IRE Transactions on information theory*, 2(3):61–79.
- Oberauer, K., Lewandowsky, S., Awh, E., Brown, G. D., Conway, A., Cowan, N., Donkin, C., Farrell, S., Hitch, G. J., Hurlstone, M. J., et al. (2018). Benchmarks for models of short-term and working memory. *Psychological bulletin*, 144(9):885.
- Okada, K., Vandekerckhove, J., and Lee, M. D. (2018). Modeling when people quit: Bayesian censored geometric models with hierarchical and latent-mixture extensions. *Behavior research methods*, 50:406–415.
- Otto, A. R., Gershman, S. J., Markman, A. B., and Daw, N. D. (2013). The curse of planning: dissecting multiple reinforcement-learning systems by taxing the central executive. *Psychological science*, 24(5):751–761.
- Oulasvirta, A., Jokinen, J. P., and Howes, A. (2022). Computational rationality as a theory of interaction. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–14.
- Padoa-Schioppa, C. and Assad, J. A. (2006). Neurons in the orbitofrontal cortex encode economic value. *Nature*, 441(7090):223–226.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- Payne, J. W. (1976). Task complexity and contingent processing in decision making: An information search and protocol analysis. *Organizational behavior and human performance*, 16(2):366–387.
- Payzan-LeNestour, E. and Bossaerts, P. (2011). Risk, unexpected uncertainty, and estimation uncertainty: Bayesian learning in unstable settings. *PLoS computational biology*, 7(1):e1001048.
- Pearl, J. (1984). *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley Longman Publishing Co., Inc.
- Pedersen, C., Togelius, J., and Yannakakis, G. N. (2010). Modeling player experience for content creation. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(1):54–67.
- Pedersen, M. K., Díaz, C. M. C., Alba-Marrugo, M. A., Amidi, A., Basaiawmoit, R. V., Bergenholz, C., Christiansen, M. H., Gajdacz, M., Hertwig, R., Ishkhanyan, B., et al. (2021). Cognitive abilities in the wild: Population-scale game-based cognitive assessment. *Preprint at <http://arxiv.org/abs/2009.05274>*.
- Peterson, J. C., Bourgin, D. D., Agrawal, M., Reichman, D., and Griffiths, T. L. (2021). Using large-scale experiments and machine learning to discover theories of human decision-making. *Science*, 372(6547):1209–1214.

- Pezzulo, G., Donnarumma, F., Maisto, D., and Stoianov, I. (2019). Planning at decision time and in the background during spatial navigation. *Current opinion in behavioral sciences*, 29:69–76.
- Pfeiffer, B. E. and Foster, D. J. (2013). Hippocampal place-cell sequences depict future paths to remembered goals. *Nature*, 497(7447):74–79.
- Popp, P. O. and Gureckis, T. M. (2020). Ask or tell: Balancing questions and instructions in intuitive teaching. In *CogSci*.
- Rescorla, R. A. (1972). A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement. *Classical conditioning, Current research and theory*, 2:64–69.
- Rubinstein, R. Y. and Kroese, D. P. (2016). *Simulation and the Monte Carlo method*. John Wiley & Sons.
- Russek, E., Acosta-Kane, D., van Opheusden, B., Mattar, M. G., and Griffiths, T. (2022). Time spent thinking in online chess reflects the value of computation.
- Russell, S. and Wefald, E. (1991). Principles of metareasoning. *Artificial intelligence*, 49(1-3):361–395.
- Saariluoma, P. (1992). Visuospatial and articulatory interference in chess players' information intake. *Applied cognitive psychology*, 6(1):77–89.
- Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247.
- Schultz, W., Dayan, P., and Montague, P. R. (1997). A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599.
- Schulz, E., Bhui, R., Love, B. C., Brier, B., Todd, M. T., and Gershman, S. J. (2019). Structured, uncertainty-driven exploration in real-world consumer choice. *Proceedings of the National Academy of Sciences*, 116(28):13903–13908.
- Seo, H., Cai, X., Donahue, C. H., and Lee, D. (2014). Neural correlates of strategic reasoning during competitive games. *Science*, 346(6207):340–343.
- Sezener, C. A., Dezfouli, A., and Keramati, M. (2019). Optimizing the depth and direction of prospective planning using information values. *PLOS Computational Biology*.
- Shadlen, M. N. and Newsome, W. T. (1998). The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *Journal of neuroscience*, 18(10):3870–3896.
- Shaker, N., Yannakakis, G., and Togelius, J. (2010). Towards automatic personalized content generation for platform games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 6, pages 63–68.

- Shallice, T. (1982). Specific impairments of planning. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 298(1089):199–209.
- Shannon, C. E. (1950). Xxii. programming a computer for playing chess. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41(314):256–275.
- Shepard, R. N. and Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703.
- Shin, M., Kim, J., van Opheusden, B., and Griffiths, T. L. (2023). Superhuman artificial intelligence can improve human decision-making by increasing novelty. *Proceedings of the National Academy of Sciences*, 120(12):e2214840120.
- Siegelmann, H. T. and Sontag, E. D. (1995). On the computational power of neural nets. *Journal of computer and system sciences*, 50(1):132–150.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.
- Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *nature*, 550(7676):354–359.
- Snider, J., Lee, D., Poizner, H., and Gepshtein, S. (2015). Prospective optimization with limited resources. *PLoS Comput Biol*, 11(9):e1004501.
- Solway, A. and Botvinick, M. M. (2012). Goal-directed decision making as probabilistic inference: a computational framework and potential neural correlates. *Psychological review*, 119(1):120.
- Solway, A. and Botvinick, M. M. (2015). Evidence integration in model-based tree search. *Proceedings of the National Academy of Sciences*, 112(37):11708–11713.
- Stafford, T. and Dewar, M. (2014). Tracing the trajectory of skill learning with a very large sample of online game players. *Psychological science*, 25(2):511–518.
- Stern, D., Herbrich, R., and Graepel, T. (2006). Bayesian pattern ranking for move prediction in the game of go. In *Proceedings of the 23rd international conference on Machine learning*, pages 873–880.
- Steyvers, M. and Benjamin, A. S. (2019). The joint contribution of participation and performance to learning functions: Exploring the effects of age in large-scale data sets. *Behavior research methods*, 51(4):1531–1543.

- Steyvers, M., Hawkins, G. E., Karayannidis, F., and Brown, S. D. (2019). A large-scale analysis of task switching practice effects across the lifespan. *Proceedings of the National Academy of Sciences*, 116(36):17735–17740.
- Steyvers, M., Lee, M. D., and Wagenmakers, E.-J. (2009). A bayesian analysis of human decision-making on bandit problems. *Journal of Mathematical Psychology*, 53(3):168–179.
- Steyvers, M. and Schafer, R. J. (2020). Inferring latent learning factors in large-scale cognitive training data. *Nature Human Behaviour*, 4(11):1145–1155.
- Sukhov, N., Dubey, R., Duke, A., and Griffiths, T. (2023). When to keep trying and when to let go: Benchmarking optimal quitting.
- Sutskever, I. and Nair, V. (2008). Mimicking go experts with convolutional neural networks. In *International Conference on Artificial Neural Networks*, pages 101–110. Springer.
- Sutton, R. and Barto, A. (2018). *Reinforcement learning: An Introduction*, 2nd edition. MIT Press.
- Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine learning*, 3:9–44.
- Sutton, R. S. (1991). Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163.
- Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.
- Tajima, S., Drugowitsch, J., Patel, N., and Pouget, A. (2019). Optimal policy for multi-alternative decisions. *Nature neuroscience*, 22(9):1503–1511.
- Ten, A., Kaushik, P., Oudeyer, P.-Y., and Gottlieb, J. (2020). Humans monitor learning progress in curiosity-driven exploration.
- Tesauro, G. et al. (1995). Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68.
- Thompson, B., Van Opheusden, B., Sumers, T., and Griffiths, T. (2022). Complex cognitive algorithms preserved by selective social learning in experimental populations. *Science*, 376(6588):95–98.
- Togelius, J., De Nardi, R., and Lucas, S. M. (2006). Making racing fun through player modeling and track evolution.
- Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological review*, 55(4):189.
- Tolpin, D. and Shimony, S. (2012). Mcts based on simple regret. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 570–576.

- Török, G., Pomiechowska, B., Csibra, G., and Sebanz, N. (2019). Rationality in joint action: Maximizing coefficiency in coordination. *Psychological science*, 30(6):930–941.
- Treisman, A. M. and Gelade, G. (1980). A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136.
- Tromp, J. (2016). The number of legal go positions. In *International Conference on Computers and Games*, pages 183–190. Springer.
- Tsividis, P. A., Loula, J., Burga, J., Foss, N., Campero, A., Pouncy, T., Gershman, S. J., and Tenenbaum, J. B. (2021). Human-level reinforcement learning through theory-based modeling, exploration, and planning. *arXiv preprint arXiv:2107.12544*.
- Turing, A. M. (2009). *Computing machinery and intelligence*. Springer.
- Uiterwijk, J. W. (2019). Solving strong and weak 4-in-a-row. In *2019 ieee conference on games (cog)*, pages 1–8. IEEE.
- Van Harreveld, F., Wagenmakers, E.-J., and Van Der Maas, H. L. (2007). The effects of time pressure on chess skill: an investigation into fast and slow processes underlying expert performance. *Psychological research*, 71:591–597.
- van Opheusden, B., Acerbi, L., and Ma, W. J. (2020). Unbiased and efficient log-likelihood estimation with inverse binomial sampling. *PLoS computational biology*, 16(12):e1008483.
- van Opheusden, B., Kuperwajs, I., Galbiati, G., Bnaya, Z., Li, Y., and Ma, W. J. (2023). Expertise increases planning depth in human gameplay. *Nature*, pages 1–6.
- van Opheusden, B. and Ma, W. J. (2019). Tasks for aligning human and machine planning. *Current Opinion in Behavioral Sciences*, 29:127–133.
- Vélez, N., Christian, B., Hardy, M., Thompson, B. D., and Griffiths, T. L. (2023). How do humans overcome individual computational limitations by working together? *Cognitive Science*, 47(1):e13232.
- Wang, R. E., Wu, S. A., Evans, J. A., Tenenbaum, J. B., Parkes, D. C., and Kleiman-Weiner, M. (2020). Too many cooks: Coordinating multi-agent collaboration through inverse planning.
- Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8:279–292.
- Wilson, R. C., Geana, A., White, J. M., Ludvig, E. A., and Cohen, J. D. (2014). Humans use directed and random exploration to solve the explore-exploit dilemma. *Journal of Experimental Psychology: General*, 143(6):2074.
- Wise, T., Emery, K., and Radulescu, A. (2023). Naturalistic reinforcement learning. *Trends in Cognitive Sciences*.

- Wunderlich, K., Dayan, P., and Dolan, R. J. (2012a). Mapping value based planning and extensively trained choice in the human brain. *Nature neuroscience*, 15(5):786–791.
- Wunderlich, K., Smittenaar, P., and Dolan, R. J. (2012b). Dopamine enhances model-based over model-free choice behavior. *Neuron*, 75(3):418–424.
- Yannakakis, G. N., Spronck, P., Loiacono, D., and André, E. (2013). Player modeling.
- Yannakakis, G. N. and Togelius, J. (2018). *Artificial intelligence and games*, volume 2. Springer.
- Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., and Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.
- Yoo, A. H. and 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.
- Yoshida, W., Dolan, R. J., and Friston, K. J. (2008). Game theory of mind. *PLoS computational biology*, 4(12):e1000254.
- Zhang, S. and Yu, A. J. (2013). Forgetful bayes and myopic planning: Human learning and decision-making in a bandit setting. *Advances in neural information processing systems*, 26.
- Zheng, Z. S., Lin, X. D., Topping, J., and Ma, W. J. (2022). Comparing machine and human learning in a planning task of intermediate complexity. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.