Learning and Decision Making in Social Contexts: Neural and Computational Models

by

Peter Duggins

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2022

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

Supervisor:	Chris Eliasmith Professor, Dept. of Systems Design Engineering, University of Waterloo
Internal Member:	Bryan Tripp Professor, Dept. of Systems Desigin Engineering, University of Waterloo
Internal Member:	Kerstin Dautenhahn Professor, Dept. of Electrical and Computer Engineering, University of Waterloo
Internal-External Member:	John McLevey Professor, Dept. of Knowledge Integration, University of Waterloo
External Examiner:	Rajesh P.N. Rao Professor, Paul G. Allen School of Computer Science and Engineering University of Washington

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Social interaction is one of humanity's defining features. Through it, we develop ideas, express emotions, and form relationships. In this thesis, we explore the topic of social cognition by building biologically-plausible computational models of learning and decision making. Our goal is to develop mechanistic explanations for how the brain performs a variety of social tasks, to test those theories by simulating neural networks, and to validate our models by comparing to human and animal data.

We begin by introducing social cognition from functional and anatomical perspectives, then present the Neural Engineering Framework, which we use throughout the thesis to specify functional brain models. Over the course of four chapters, we investigate many aspects of social cognition using these models. We begin by studying fear conditioning using an anatomically accurate model of the amygdala. We validate this model by comparing the response properties of our simulated neurons with real amygdala neurons, showing that simulated behavior is consistent with animal data, and exploring how simulated fear generalization relates to normal and anxious humans. Next, we show that biologically-detailed networks may realize cognitive operations that are essential for social cognition. We validate this approach by constructing a working memory network from multi-compartment cells and conductance-based synapses, then show that its mnemonic performance is comparable to animals performing a delayed match-to-sample task. In the next chapter, we study decision making and the tradeoffs between speed and accuracy: our network gathers information from the environment and tracks the value of choice alternatives, making a decision once certain criteria are met. We apply this model to a two-choice decision task, fit model parameters to recreate the behavior of individual humans, and reproduce the speed-accuracy tradeoff evident in the human population. Finally, we combine our networks for learning, working memory, and decision making into a cognitive agent that uses reinforcement learning to play a simple social game. We compare this model with two other cognitive architectures and with human data from an experiment we ran, and show that our three cognitive agents recreate important patterns in the human data, especially those related to social value orientation and cooperative behavior. Our concluding chapter summarizes our contributions to the field of social cognition and proposes directions for further research.

The main contribution of this thesis is the demonstration that a diverse set of social cognitive abilities may be explained, simulated, and validated using a functionally-descriptive, biologically-plausible theoretical framework. Our models lay a foundation for studying increasingly-sophisticated forms of social cognition in future work.

Acknowledgements

I would like to thank the following people, without whom this work would have been impossible

- My advisor, Chris Eliasmith, for introducing me to computational neuroscience and guiding my academic journey
- My colleagues in the Computational Neuroscience Research Group, especially Terry Stewart, for our many conversations and collaborations
- My friends and partners, near and far, for bringing joy to my life and supporting me in difficult times
- My parents, Megan Dethier and David Duggins, for their unerring guidance and unconditional love

Dedication

This thesis is dedicated to the emerging field of computational social science. It is my hope that a better understanding of human social cognition will help us come together, as a global community of independent individuals, and confront the major challenges of our time.

This thesis is also dedicated to the future of artificial intelligence. I hope that the path towards AI sentience and independence is mutually beneficial, and that we will treat each other with compassion and respect.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Social Cognition

Social intelligence is a defining feature of *homo sapiens*. From our hunter-gatherer origins to our modern societies, social interactions are essential for survival and well-being, both for individuals and for communities. Every day, we communicate to exchange information and form opinions, compete for limited resources, and coordinate with each other to solve complex problems. We create products for human use and entertain ourselves with multiplayer games and interpersonal stories. Many of humanity's greatest accomplishments, from art and architecture to technology and sciences, exemplify our ability to bring together talented individuals into collaborative projects that produce public goods. However, the greatest risks facing our species, from climate change to international war, also partly stem from difficulty accepting and working with each other; on a societal scale, we have historically struggled to coordinate our actions in ways that promote desirable collective outcomes. Understanding social cognition thus has immense value, both from a scientific and a societal perspective.

Social psychology, the formal study of social intelligence through the systematic measurement of human thoughts, feelings, and behaviors [134], began in the early 1900s with experiments on group behavior [241] and the publication of the first social psychology textbook [162]. The field grew substantially in the World War II era, first with studies on persuasion, propaganda, and psychological warfare from the US military, and later with studies seeking to understand the conformity, obedience, and social pressures that enabled wartime atrocities. While rife with methodological and ethical problems, many of the experiments conducted during this period, including Asch's conformity study [10], the

Stanford Prison Experiment [95], and Milgram's obedience study [166], captured the public attention and inspired renewed interest in social psychology. In the 1970s and 80s, the field became more cognitively oriented, focusing more on the mental and neural processes underlying social thinking while developing tools to address the methodological failures of early experiments. Modern social psychology studies numerous intrapersonal phenomena, such as attitudes, persuasion, and self-concepts, and interpersonal phenomena, such as social influence, group dynamics, and interpersonal attraction, using a diverse set of experimental, statistical, and computational tools.

At a high level, social intelligence can describe any thought, feeling, or behavior that is influenced by the presence of others or by internalized social norms. Given the pervasiveness of social interaction described above, it is easy to see how a wide variety of cognitive operations fall within this category. While this breadth makes it difficult to constrain the study of social intelligence, it also reflects an important principle: social cognition includes and builds upon existing non-social cognitive components. Many of the functions and operations performed by the brain, such as classification, working memory, valuation, and decision making, will be useful for social thinking in a general sense. What's more, evolution has a tendency to leverage existing functional mechanisms when designing new cognitive tools: it seems likely that, as humans evolved and refined their socio-cognitive abilities, natural selection favored mutations that made incremental changes and improvements to existing neural systems, rather than inventing new social cognition systems from scratch. For example, the amygdala is known to interpret social information and initiate appropriate social behaviors in social primates [69], a role that may have evolved by functionally extending its existing ability to coordinate approach and avoidance behaviors in non-social settings [2]. Similarly, the vmPFC is thought to evaluate the quality of potential actions for an individual, but neural and behavioral evidence suggest that social animals also use this area (alongside more dedicated structures) to estimate the social value of potential actions, leading to a combined utility estimate that is used to direct social behavior [47]. In studying social cognition, it is therefore important to recognize that the cognitive operations and functional brain areas under discussion exist within a larger cognitive system: it is not always possible to differentiate social operations from domain-general operations like associative learning, working memory, or decision making. Labelling cognitive mechanisms as "social" is a useful convention rather than a clear-cut scientific distinction.

The overarching goal of this thesis is to explore social cognition from the perspective of learning and decision making. We believe that, by focusing on the adaptive nature of social intelligence, we can understand the cognitive mechanisms that underlie a wide range of social phenomena across many disciplines within social psychology. This belief

stems from the observation that most social environments are fluid: they contain intelligent agents whose thoughts and behaviors change across time, culture, context, and more. To successfully navigate such environments, brains have evolved learning systems that model the social world and make smart decisions within it. Rather than study the models that individuals have built over their lifetimes, the heuristics that groups rely upon as cognitive shortcuts, or the statistical regularities of behaviors within particular cultures, we will study the processes through which these phenomena are implemented in the brain. Thus, our goal is to describe the cognitive algorithms that individuals use to learn about, and make decisions within, environments that contain other intelligent agents. We strive to identify the neural and anatomical correlates of these algorithms and to demonstrate that biological brains can (and do) realize them. To do so, we rely on computational models that rigorously define algorithms for learning and decision making, train these models to perform simple cognitive tasks from social psychology, and validate them by comparing model outputs to neural and behavioral data from social animals.

1.2 Computational Models

Computational models are powerful tools for studying social cognition. In contrast to many methods in social psychology, economics, and neuroscience, computational methods provide a mechanistic account of brain function, which can inform our understanding of how particular cognitive processes are implemented in the brain. Computational models can also be used to rigorously specify cognitive theories and contrast competing hypotheses: outputs from competing models can be compared to neural and behavioral data, providing an empirical basis for comparison. Notably, models make precise quantitative predictions about social decision making across people, context, time, and other experimental confounds. Parameter fitting also allows researchers to identify how individual differences in learning variables or decision variables uniquely interact to determine behavior. Finally, computational models can synthesize theories from diverse scientific fields using a common framework, and can be incrementally expanded by adding additional mechanisms and variables; this provides a grounded basis for theoretical development with the potential to explain an expanding body of data. These benefits have been widely recognized within the cognitive science community [243], leading to an encouraging increase in the quantity and quality of computational models for social cognition [36, 150].

Of course, computational models are incredibly diverse with respect to the mechanisms they simulate, the parameters they include, and the data they produce. Before embarking on the task of building models to study social cognition, it is important to think about

which features of computational models are scientifically valuable; that is, which features meaningfully enhance our ability to understand the relationship between the biological brain and our social and cognitive capabilities. We believe that three features, in particular, characterize a good computational model of social cognition: biological fidelity, functional capacity, and explainability.

First, the components of a computational model should resemble components of the biological brain. While high-level models of artificial intelligence may perform at human levels on various cognitive tasks, they do not necessarily inform our understanding of how humans perform those tasks. All the models presented here include several biological constraints that ensure they correspond to brains. For instance, our models are constructed out of spiking neurons connected with synaptic weights: while more detailed biophysical models of neural physiology and connectivity exist, these constraints impose a basic structure that is qualitatively more brain-like than many other approaches to artificial intelligence, including deep neural networks and symbolic production rules. Furthermore, all cognitive operations performed by our models are computed dynamically through neural connection weights; this ensures that the model has not offloaded any key computational components to non-neural platforms, and ensures that the problems of representational noise and algorithmic specification are directly addressed by the model, as they must be by brains. Finally, we identify the anatomical areas associated with our model components wherever possible, and tag components as biologically-questionable otherwise. While the biological realism of a neurocomputational model can always be expanded, we believe these biological constraints strike an acceptable balance between realism and coherence.

Second, computational brain models must have functional capacity: that is, they must represent external inputs, perform functional operations on those representations, and produce behavioral outputs. Clearly, the scientific understanding of social cognition has not yet reached a stage where we can expect models to navigate complex social environments in humanlike ways. While many algorithms play multiplayer games at superhuman levels [214] or display remarkable linguistic abilities on standardized tasks [215], much work remains to build agents that can flexibly interact with humans across many social contexts. As mentioned above, we believe that the best approach to developing flexible cognitive agents is to achieve a deeper understanding of the basic mechanisms of learning and decision making. For us, this means constructing agents endowed with learning and decision making systems and showing that they can, through experience, learn to complete various social tasks. We say that an agent has functional capacity if it can learn to perform a wide array of such tasks, and if its behavior resembles human behavior.

Finally, computational models should be explainable with regards to what information is being represented and what operations are being applied. All else being equal,

a model whose internal processes can be clearly understood is preferable to a black-box model: while both models may predict the data equally well, the former will have more scientific value with respect to characterizing how social intelligence arises. For example, the internal representations and computations learned by deep neural networks are often obscure, requiring independent tools to interpret [168] or to correct learned biases [252]. However, it is possible to overstep when designing models to meet this criteria: we should not expect that everything the brain does will be amenable to linguistic or mathematical description using our current scientific toolkit. Forcing models to use a specific set of high-level symbols, as was common in the 1960s, 70s, and 80s with connectionist models [204] and good-old-fashioned AI [98], may severely limit the model's ability to generalize to novel tasks or scale to larger symbolic vocabularies. In summary, models with comprehensible internal representations and operations are useful for describing the mechanisms of social cognition, but subsymbolic representations and operations are required to capture the flexible nature of social cognition.

One persistent problem with computational models is overspecification and overfitting. If a model includes enough parameters and mechanisms, it can recreate any dataset, without necessarily informing our understanding of social cognition. Many models are built to explain particular social phenomenon, and do so convincingly; the greater challenge, and the more important scientific test, is whether these models can also explain additional, comparable social phenomenon. If a model must be significantly reorganized to apply to other forms of social cognition, we should be wary of its general explanatory power. While many tools can be used to combat the overfitting of model parameters to a particular dataset [243], the problem of overspecifying the structure or mechanisms of a model to a narrow scientific domain is more insidious. We believe this problem is best addressed through the use of cognitive architectures, which are theories about the structure of the mind that focus on generality and usability: they intend to explain a wide variety of cognitive phenomena using a common set of mechanisms that may work together to perform many cognitive tasks. The most successful architectures, such as Adaptive Control of Thought - Rational (ACT-R) [5] and the Semantic Pointer Architecture (SPA) [64], specify subsystems for common cognitive operations such as working memory, attention, association, and decision making; because the mechanisms underlying the operation of these subsystems are compatible, researchers can connect subsystems together to form larger systems that perform more complex cognitive tasks. Both ACT-R and SPA have been used this way for decades, producing a wide body of research modelling diverse cognitive phenomena. In this thesis, our computational models derive from one or more cognitive architectures.

1.3 Functional Neuroanatomy

Recall that the goal of this thesis is to study the neural and cognitive mechanisms of learning and decision making in social contexts, and to recreate them using computational models. Doing so clearly requires an investigation of the brain itself. By observing the brain's activity while it engages in social tasks, we can identify which parts of the brain are involved in which social operations, then develop more detailed representational and functional hypotheses. These theories can be rigorously specified using formal computational models that are representationally and functionally segregated (i.e., have components that correspond to the identified brain systems) and perform social tasks (i.e., produce simulated activity and behavior that may be compared to empirical data). The successes and failures of these hypotheses and models should inform the design of new experiments, observations, and interpretations.

In this section, we review the brain's functional neuroanatomy, with the aim of developing a high-level picture of the role played by different brain areas during social cognition. In subsequent chapters, we provide greater detail about the functional organization of particular regions, and discuss existing computational models of their behavior. In this review, we generally reference recent review papers and meta analyses, rather than citing primary sources that detail particular empirical experiments. Review papers synthesize an overwhelming body of literature into coherent theories about brain organization. More importantly, these papers are less likely than the primary papers to draw overgeneralized conclusions; individual studies frequently claim that certain brain areas represent a particular quantity, or perform a particular function, based only on a single experiment or dataset. We report the conclusions drawn by several recent reviews, and discuss the extent to which they agree or disagree about functional neuroanatomy.

Despite these precautions, there are numerous empirical and theoretical challenges to functional neuroanatomy. In Appendix A, we discuss five of these challenges at length, including: the difficulty of dividing the brain into discrete regions; the problem of ascribing specific symbolic representations to neural activities; the confusion of functional redundancy between brain areas; the intractability of designing controlled social experiments; and the prevalence of vague terminology. Readers who are troubled by the breadth of our neuroanatomical claims should refer to these notes, which qualify many of our statements.

1.3.1 The Value Based Framework

When making social decisions, an individual must weight competing alternatives and choose an option that satisfies various criteria. Most cognitive scientists who study decision making describe this process using the "value based framework" (our terminology), in which an individual, or "agent", assigns each potential action a "value" and selects the action with the highest value when making a decision [231]. In the value based framework, value is a catch-all quantity: it incorporates every feature than an agent potentially cares about, with respect to deciding between candidate actions. While conceptually simple, this framework is consistent with the many complexities of learning and decision making in social contexts. Agents contain sophisticated mental models that evaluate potential actions based on a wealth of social and nonsocial information. These mental models make predictions about the outcome of taking various actions, and agents learn to improve their value estimates by comparing their predictions to the outcomes their actions actually produce in the social world. Agents may also adapt their mental models to accommodate high-level goals, emotional states, and social contexts through complex modulatory mechanisms. Overall, the value based framework is consistent with behavioral and neuroimaging data [180], can easily be realized in computational models [36], and is closely associated with reinforcement learning, another successful theory of learning and decision making [230]. For these reasons, we believe that it is appropriate for studying learning and decision making in social contexts, and we organize our review of functional neuroanatomy accordingly.

In the following sections, we describe how value based decision making is organized in the human brain. We begin by reporting regions that are involved in representing social information, processing it, and computing value estimates. We discuss how these estimates are integrated into a common currency for valuation, then describe how this process is modulated by other regions that assess goals, norms, and context. We also identify regions that are crucial to error monitoring and learning, then finally point out several regions that facilitate action selection. Fig. 1.1 provides an overview of this anatomical mapping.

1.3.2 Value Estimation

In the last decade, a great deal of research has focused on the neural basis of valuation, and many excellent meta analyses and review articles have summarized empirical data from fMRI studies, behavior experiments, neurochemical and lesion studies, studies of neurological disorders, and more [231, 205, 198, 194, 180, 177, 154, 122, 81, 74, 46]. A common theme among these analyses is the distinction between "proself" and "prosocial"

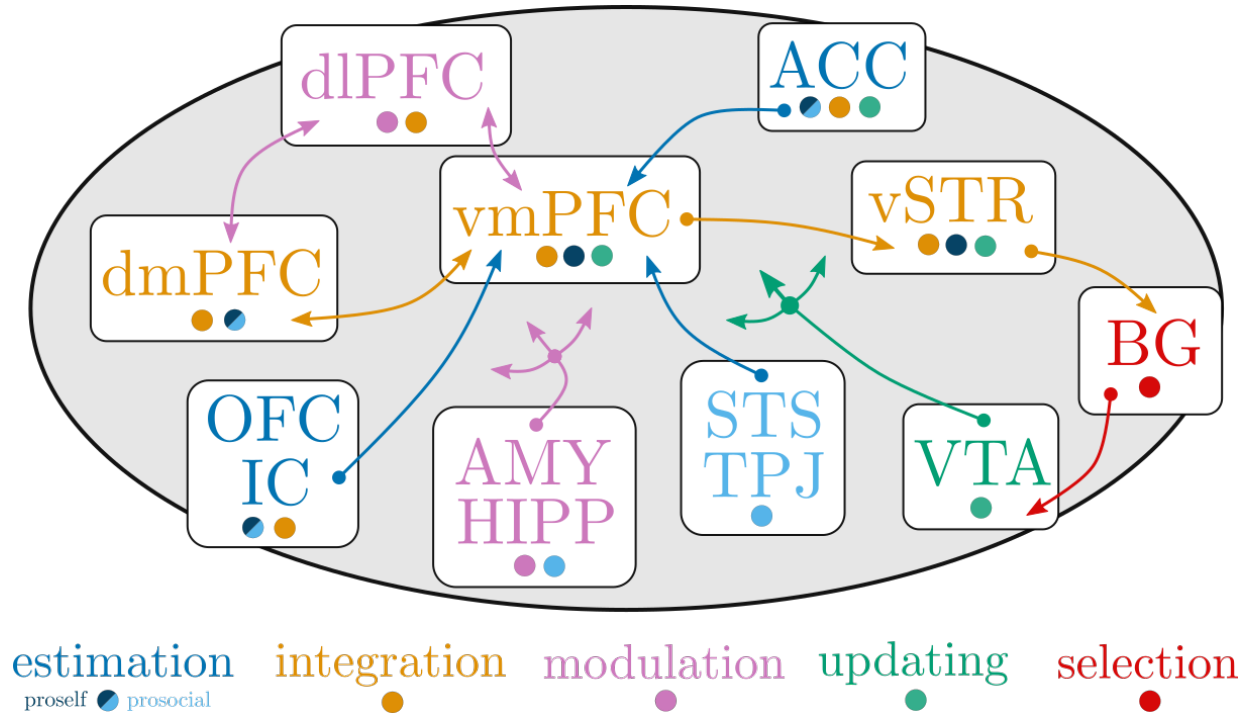

Figure 1.1: Anatomical mapping of the value based framework in the human brain. Colored dots indicate the functional roles of each region, with the primary role indicated by text color. Directed arrows indicate specific anatomical connections, and multi-pronged arrows indicate diffuse connectivity to multiple areas within the region. Abbrevations: dorsolateral prefrontal cortex (dlPFC), dorsomedial prefrontal cortex (dmPFC), ventromedial prefrontal cortex (vmPFC), anterior cingulate cortex (ACC), orbitofrontal cortex (OFC), insular cortex (IC, particularly anterior insula), amygdala (AMY), hippocampus (HIPP), superior temporal sulcus (STS), temperoparietal junction (TPJ), ventral striatum (vSTR), vental tegmental area (VTA), basal ganglia (BG).

cognition. Proself cognition estimates action values with reference to the individual and their personal goals. Prosocial cognition, on the other hand, is directly concerned with understanding other agents, predicting social behaviors, and evaluating actions with respect to social goals; it encompasses trust, social status, emotional recognition, empathy, motivational inference, mentalizing, social norms, norm enforcement, and more. Converging evidence strongly suggests that proself and prosocial cognition occur in distinct brain regions [231, 180]. Prosocial areas process social information and produce rich social representations, then use these representations to evaluate candidate actions; these cognitive processes are typically required for an individual to exhibit prosocial behaviors, including cooperative, norm-compliant, and altruistic behaviors [194].

Several brain regions in and around the medial prefrontal cortex (mPFC) and the temporal cortex are involved in prosocial cognition. As a whole, activity in the mPFC reliably correlates with multisensory integration, distinguishing self from other, processing social emotions, estimating social status and hierarchy, mentalizing, and adhering to social norms [198, 239, 81, 243, 74, 36]. Given the size and interconnectivity of the mPFC, this area likely performs various social operations related to analyzing social relationships and predicting the social impact of actions. For instance, the dorsomedial prefrontal cortex (dmPFC) tracks other agents' beliefs and intentions and responds to deviations between individual and group preferences, suggesting that it may be a critical region for assessing conflicts between proself and prosocial behavior [36]. Some researchers believe that dmPFC also facilitates value integration [231].

The orbitofrontal cortex (OFC) is a brain region adjacent to the mPFC that also performs prosocial valuation. OFC activity correlates with proself and prosocial value, distinctions between self and other, and social rewards [239, 198]. Some researchers believe the OFC is also involved in value integration [81, 180], while others contend it is critical for emotional valuation of reward signals [200]. While the OFC is clearly important for prosocial valuation, more work is needed to specify its exact functional role.

The anterior cingulate cortex (ACC) also lies adjacent to the mPFC and manages a variety of features and computations related to prosocial valuation; it is important for empathy and mentalizing [198, 239, 154, 205], for monitoring conflict between cognitive and emotional processes and accommodating social uncertainty [194, 154], and for facilitating prosocial learning through reward prediction errors (RPEs) [198, 18, 194, 36, 205]. It may also be directly involved in assigning value to actions [81, 231]. Other areas in the cingulate cortex also process social information: the dorsal posterior cingulate cortex facilitates mentalizing [194], the midcingulate cortex facilitates empathy [243, 194], and the paracingulate cortex facilitates the development of trust and inference of other agents' intentions [154]. As with the OFC, more work is needed to functionally subdivide the

cingulate cortex and clarify how the ACC coordinates learning and prosocial valuation.

Another area that seems to perform multiple functions related to prosocial valuation is the insular cortex (IC), particular the anterior insula (AI). IC facilitates empathy [243, 194, 154] and processes social emotions, including those related to social norms like fairness and cooperation [136]. Some researchers believe the insula is a critical hub for the integration of social information, bringing together external sensory information, internal autonomic information, and signals encoding reward to facilitate social learning and affective decision making [198, 177]. In particular, the insula may coordinate social behavior by managing a network of subcortical areas that includes behavioral systems, emotional systems, and reward systems [198]. In nonhuman animals ranging from mammals to birds to reptiles, these structures facilitate simpler forms of learning and decision making in social contexts, including sexual displays, aggression, parental care, foraging, and habitat selection [177]. We suspect that the cortical decision making network evolved to complement this system, but that the original system continues to influence a variety of social behaviors in humans.

The temperoparietal junction (TPJ) and superior temporal sulcus (STS) are two areas located in the temporal cortex that are frequently implicated in mentalizing, the process of taking the perspective of other agents and predicting the social consequences of actions [239, 243, 94, 36, 154]. Other temporal areas are also important for processing social information, but are not directly involved in valuation: these include the temporal poles (which are implicated in mentalizing [243]), and the medial temporal lobes and fusiform gyrus (which facilitate the recognition of known agents and the recall of previous interactions [239]). TPJ and STS project to cortical areas like the mPFC and ACC, allowing prosocial valuations estimated through mentalizing to integrate with other specialized value estimates.

Finally, proself valuation is important for social decision making; without it, individuals would be unable to pursue their own goals during social interactions. Neural estimates of "economic value" have been widely studied, both in nonsocial contexts [30] and social contexts [231]. These studies reveal a common set of brain regions that are consistently activated during prosocial valuation. The two most commonly cited regions are the ventromedial prefrontal cortex (vmPFC) and the ventral striatum (vSTR), but many other regions also correlate with proself valuation, including the dmPFC, OFC, posterior cingulate cortex, and insula [30]. As we discuss in the next section, many of these areas are implicated in value integration, so labelling these regions as exclusively "proself evaluators" is misleading. Nonetheless, it is important to recognize their heightened role in computing the self-interested value of actions, given a mixture of economic and social information.

1.3.3 Value Integration

In cognitive science and neuroeconomics, there is a widely accepted view that different regions in the brain use different mechanisms to compute multiple independent value estimates for potential actions, but that these valuations are eventually integrated into a common currency that determines subsequent decisions [231, 180, 30]. The synthesis of value estimates occurs primarily in the vmPFC and vSTR [239, 243, 180, 205, 194]. Neural activities in these regions reliably correlate with both proself and prosocial features across a variety of cognitive tasks and social contexts, and these activities robustly predict individuals' subsequent behaviors. Notably, when a social context presents a tradeoff between individual goals and collective goals, vmPFC and vSTR seem to combine proself and prosocial evaluations into more complex combined representations that determine decision making. This suggests that value integration may be a complex function that combines specialized value estimates in a flexible, goal-driven manner. In the next section, we review how other systems may modulate and control the vmPFC to achieve this flexibility.

Interestingly, the activity of vmPFC and vSTR also correlate with RPEs, and with changes in value estimates during cognitive tasks where participants learn action values [36, 18, 81, 150]. While these observations are consistent with the value based framework in general, they suggest that vmPFC and vSTR may be involved in the primary valuation of actions, rather than the secondary integration of precomputed values. As discussed in the previous section, vmPFC and vSTR are directly involved in proself valuation, which would explain why they represent and respond to RPEs: they are actively refining their proself estimates in response to external feedback. However, it is unclear whether these regions also directly evaluate prosocial actions using prosocial RPEs, or whether they use existing prosocial estimates originating in regions like OFC, ACC, STS, and TPJ. It seems likely that these regions perform both functions to some degree, but for conceptual clarity, we will assume that vmPFC and vSTR compute proself values directly, then combine these estimates with external prosocial estimates to produce an overall value signal.

Further evidence of value integration in vmPFC and vSTR comes from their connectivity to other brain regions. Many of the value estimating regions above, including mPFC, ACC, OFC, STS, and TPJ, project to one or both of these areas, and vmPFC and vSTR are themselves interconnected. These connections facilitate the transmission of value estimates into neural systems that can compare and integrate them, allowing these systems to evaluate actions according to many relevant features [231]. Furthermore, connectivity between vmPFC, dmPFC, and dorsolateral prefrontal cortex (dlPFC) suggests that these regions may work together to ensure value integration aligns with high-level goals, which may depend on the current social environment [231, 81, 243, 180]. Finally, these systems

project to areas that are thought to compare candidate actions and execute the selected action, including basal ganlgia, cerebellum, and motor cortex [192, 72]. Thus, vmPFC and vSTR are well-positioned to act as a "hub" of value integration.

Of course, many other regions are implicated in value integration. Activity in the dmPFC and ACC, which lie adjacent to the vmPFC, also correlate with multiple value features, and their connectivity is similarly conducive to integration [231, 94]. Similar statements can be made about OFC [198, 180, 81]. As we discussed in Appendix A, the divisions between these brain areas may be more confusing than helpful when identifying the region of the brain that is responsible for value integration. Regardless of where exactly value integration takes place, it is clear from the behavioral, neural, and anatomical evidence that the brain's activity is well characterized by the value based framework.

1.3.4 Value Updating

A critical component of valuation is the ability to improve value estimates in response to external feedback from the environment. Given the diversity of value estimates, social goals, and types of social feedback, there are undoubtedly many neural mechanisms for updating the mental models that drive social decision making; we investigate many of these mechanisms, and the theories behind them, throughout the remainder of this thesis. In the value based framework, the brain tracks the expected value of performing an action (the standard value estimate), observes the impact of the action on the environment, and evaluates that outcome with respect to the individual's current goals. Other systems compare the expected value to the observed value and calculate a reward prediction error (RPE) based on the difference. RPEs are then used to update synaptic weights in specialized valuation systems, with the aim of reducing the gap between expected outcomes and observed outcomes.

Neural activities in many brain regions correlate with RPEs in both social and nonsocial contexts [81, 231, 84]. Unfortunately, it can be difficult to determine whether a brain region computes RPEs directly, or whether it uses RPEs computed by specialized systems. For instance, many scientists believe that vSTR directly encodes proself and prosocial RPEs [150], but these regions also seem to perform value estimation and integration. In contrast, the midbrain dopaminergic system, including the ventral tegmental area (VTA), seems crucial for encoding RPEs, but does not appear to encode action values; this system uses the neurotransmitter dopamine (DA) to facilitate synaptic plasticity in frontal cortex and basal ganglia [84], two areas that encode prosocial value and perform value integration. Furthermore, it seems as though specialized prosocial RPEs are generated and/or used by

prosocial valuations systems, most notably the ACC [198, 194, 205] and OFC [198, 199]. Overall, it seems quite likely that the computation of RPEs for value updating is distributed across several brain regions that are closely tied to value estimation.

Value updating is computationally realized in models of reinforcement learning (RL). In RL, an agent uses external feedback from the environment (rewards and punishments) to update its value estimates, with the goal of choosing more rewarding actions in the future [230]. A large body of empirical and computational work has investigated the relationship between neural activity in the regions mentioned above, and the RPE signals predicted by RL [84]. For instance, in a recent review paper that uses RL to analyze prosocial valuation and decision making [81], the authors discuss the correspondence between various neural systems and functional learning mechanisms. This work further supports the conclusion that the brain uses RPEs to update its value estimates and facilitate social decision making.

1.3.5 Value Modulation

A common feature of social decision making is that it is highly context-dependent: social norms differ between groups, emotions change rapidly, and goals are regularly reevaluated. While value updating based on RPEs is a powerful technique for updating our mental models of the social world, it is a slow process: in order to produce enduring synaptic changes, evaluation systems typically require many instances of value updating spread across time and place. In many social contexts, the brain needs to adapt on short timescales, temporarily modifying its evaluative processes to produce the best outcomes. Rather than directly updating the underlying mental models, systems that modulate social valuation may flexibly alter how those systems are expressed relative to one another. This allows an individual to maintain several sets of effective mental models, and switch between them as needed. We refer to this process as value modulation.

The primary characteristic of value modulation is that it can rapidly reorganize cognitive processing (e.g., value estimation, integration, and updating) in response to external changes (e.g., other agents' emotions and behaviors) or internal changes (e.g., goals). The brain region primarily associated with such reorganization is the dorsolateral prefrontal cortex (dlPFC): its activity correlates with executive, goal-directed control over value integration. For example, dlPFC dynamically weighs proself versus prosocial value [81, 46], overrides existing evaluations [180], and aligns goals with contextual factors [243]. dlPFC is therefore critical in coordinating an individual's response to social norms: it is involved in norm compliance and enforcement [239, 154] and in modulating valuation based on abstract rules [194]. The strong connectivity between neurons in dlPFC and vmPFC further supports the idea that value estimates, which are collected and integrated in vmPFC, may be

interpreted and modified by dlPFC, which has greater access to high-order representations of an individual's goals and its social context.

Systems involved in emotional processing also influence social decision making through value modulation. The amygdala (AMY) coordinates various emotional responses in the brain [69, 2], and is implicated in numerous social emotions, especially fear and trust [154, 177]. As input, AMY receives both social and nonsocial information: it has access to facial recognition through the temporal cortex (fusiform gyrus), episodic memory through hippocampus and medial temporal lobes, and negative stimuli (sensory or social) through brainstem, thalamus, and various cortical areas. As output, AMY connects to several areas whose neurochemical projections modulate the rest of the brain, including hypothalamus, nucleus accumbens, and VTA. AMY is therefore able to orchestrate emotional responses to salient stimuli and modulate cognitive processes including perception, attention, and valuation [122, 177, 239]. One central feature of AMY is that it learns associations between social features and aversive (or rewarding) outcomes [81]. While this could be considered a form of value estimation, AMY outputs are not compared and integrated in the same way as other valuation regions; instead, AMY's influence is realized through salience detection, reallocation of attention, and reorientation of emotional state. These changes affect value estimation, integration, and updating in indirect ways. We discuss the mechanisms of AMY learning, AMY's effects on social decision making, and the role of the neurotransmitter oxytocin more extensively in Ch. 2.

The hippocampus (HIPP) is important in forming and retrieving episodic memories, an operation that is used by many systems concerned with either valuation or emotional modulation [180]. With regards to social decision making, HIPP helps form, update, and flexibly manipulate representations of social relations; these cognitive processes help individuals navigate complex social environments [203]. There are several interesting parallels between these cognitive processes and spatial navigation, another function performed HIPP. However, more research is needed to understand exactly how these abilities fit in with the value based framework.

1.3.6 Action Selection

Once final values have been assigned to each candidate action, the brain must decide which of these actions to implement. In the value based framework, this simply requires choosing the action with the greatest final value; all the work of evaluating actions and weighing their tradeoffs has already been completed. Although choosing the option with the greatest value is mathematically trivial, it is rather difficult to implement in biological neural networks.

Many computational neuroscientists believe that action selection is implemented through some form of "competition" between candidate actions, such as mutual inhibition between groups of neurons representing different actions.

Several brain regions receive inputs from value integration areas, are recurrently connected by inhibitory synapses, and encourage mutual competition between internal representations. Several nuclei within the basal ganglia (BG), including STR, globus pallidus, and the subthalamic nuclei, form a network that has been shown, both empirically and in computational models, to realize action competition [192, 92, 219]. Consistent with the idea that BG selects between candidate actions and recommends one for execution, many BG nuclei project back to motor cortex and the cerebellum, areas that plan actions and generate motor commands. BG also projects back to the prefrontal cortex, and many cognitive scientists believe that these projections may transmit "cognitive commands" that coordinate communication between cognitive systems, effectively routing information between memory systems, evaluation systems, and motor systems [222, 5, 64]. Other researchers have suggested that simpler social actions, such as social play and sexual behaviors, are governed by networks of subcortical nuclei [177], but it is not clear how these areas would accommodate cortical value estimates. Similarly, while specialized brain areas leverage mutual inhibition for domain-specific action selection [130], it is not clear whether this mechanism generalizes to social decision making in brain regions outside the BG.

Overall, action selection has received far less attention than valuation in the social decision making literature. We view biological action selection as a difficult and interesting problem, one that requires coordination between several functional systems and introduces constraints that affect social decision making. In addition to action selection, the decision making literature rarely addresses action proposition or planning. Until now, we have assumed that candidate actions exist ready-made, but the process of actually generating reasonable candidates must be highly complex, experience-driven, and context-dependent. We mention this concern here because action proposal and action selection likely form an iterative and/or hierarchical feedback cycle. Once an action is chosen, the brain must specify that action in greater detail, which requires proposing, evaluating, and selecting additional candidate actions that expand upon the chosen action [107, 189]. The neural circuit referred to as the "cortico-basal ganlgia-thalamo-coritcal loop" likely supports hierarchical and iterative action selection in the brain [93]. More empirical, theoretical, and computational research is needed to understand the functionality of this circuit.

1.3.7 Working Memory

Working memory is a system for actively maintaining and manipulating information, and is consequently essential for a variety of social and nonsocial cognitive tasks. Unlike action selection, the neuroanatomy of working memory has been extensively studied. For many years, the consensus was that working memory was mainly associated with activity in the lateral prefrontal cortex (lPFC) [45]. This theory was based on repeated observation of "delay period activity" in dlPFC neurons: in working memory tasks, neurons that represent task-relevant information continue firing after an external stimuli is removed, if the task requires an animal to actively maintain information about the stimuli. These activities indicate that animals actively maintain representations of task relevant information over time, even when the original stimulus is removed. In recent years, this theory has been refined; modern accounts of working memory suggest that it is highly distributed throughout prefrontal cortex, sensory cortex, and parietal cortex, with memory buffers existing at many hierarchical levels for many different sensory modalities [217, 39]. PFC, in particular, may collect and maintain task relevant information originating from other memory buffers. While there are relatively few studies that investigate the relationship between working memory systems and valuation systems, it seems reasonable to assume that value estimators, integrators, and updaters use working memory networks whenever information needs to be maintained over time.

1.3.8 Summary

The brain is an overwhelmingly complex system, and our characterization of how neuroanatomy relates to value based decision making is necessarily oversimplified. To help summarize this perspective further, we provide a brief, high-level hypothesis for how information flows within this system, allowing the brain to learn about the social world make appropriate decisions; Fig. 1.1 provides a graphical overview.

The brain begins by representing features of the external world, especially those related to other agents and social groups. This involves sensory processing, such as the identification of group members by facial recognition, and linguistic processing, such as parsing the semantic and emotional content of dialogue. The brain uses this information to build rich representations of the current social context. For example, the brain estimates the trustworthiness of other individuals based on previous interactions and their social reputation. Simultaneously, the brain sets goals at various levels of abstraction: these goals reflect a mix of short- and long-term objectives, as well as proself and prosocial orientations. Finally, the brain generates candidate actions that have the potential to realize its

goals given the current social environment.

With information readily available, goals in mind, and actions to consider, the brain begins the process of value based decision making. It first uses its rich representations of the social world to evaluate candidate actions. This process appears to be distributed across many distinct evaluation systems in cortex, which are functionally specialized to compute specific types of action value, notably proself and prosocial value. Once computed, these estimates are routed to systems that integrate them into a holistic evaluation, a value that determines the overall quality of each action relative to the others. This integration is guided by other systems in the brain that represent goals and contexts: these modulate the integration process, emphasizing certain evaluations and depreciating others based on the current high-level needs of the individual. For instance, executive systems may encourage behaviors that will facilitate trust-building if an individual has joined a new social group, but may encourage selfish behaviors if the group has a reputation for treachery. Once value assignment is complete, the brain selects the action with the highest overall utility and implements it. Finally, as the individual observes the consequences of the action, the brain compares the expected outcome with the observed outcome, and computes prediction errors. These error signals drive synaptic plasticity in regions responsible for value estimation, changing how future evaluations will proceed.

1.4 Neural Engineering Framework

Now that we have reviewed the functional operations involved in learning and decision making, and discussed the brain areas that realize them, we can begin to build biologically-inspired computational models. The Neural Engineering Framework (NEF) is a theory of neural representation, originally developed by Eliasmith and Anderson [65], that provides the mathematical basis for the Semantic Pointer Architecture (SPA), and can be used on its own to construct cognitive models using spiking neural networks. The NEF characterizes spiking activity within populations of neurons as encoding information in a latent *state space*. While spikes are the physical means of communication between neurons, cognition can be analyzed as transformations of these lower-dimensional states, permitting a more abstract, computational, or symbol-like description of what brains do. We assume that this state space can be represented by a vector-valued signal $\mathbf{x}(t)$, and that the cognitive operations performed in the brain may be described through the dynamics of this state space $\dot{\mathbf{x}}(t)$. At the sensory periphery, neurons transduce external signals (light, sound, etc.) into spikes that represent the stimuli. These representations are transformed via connections within the brain, ultimately producing motor commands that activate the

body's muscles.

1.4.1 Representation, Transformation, and Dynamics

Broadly speaking, to build a cognitive model using the NEF, we must describe the relevant cognitive processes in terms of state space dynamics (a procedure facilitated by the SPA), then determine the connectivity within a neural network such that its spiking activity implements those dynamics. To do so, the theory defines methods for encoding and decoding between neural activity and the state space, and describes how synaptic connections implement state space transformations. Given a signal $\mathbf{x}(t)$ and a population of neurons, the signal must drive those neurons to fire in patterns that represent the signal. Each neuron spikes most frequently when presented with its particular "preferred stimulus" and responds less strongly to increasingly dissimilar stimuli (i.e., values of $\mathbf{x}(t)$). Each neuron i is accordingly assigned a preferred direction vector, or *encoder*, $\mathbf{e}_i$. At the sensory periphery, $\mathbf{e}_i$ determines how the external $\mathbf{x}(t)$ is transduced into electrical inputs to neuron i; within the network itself, $\mathbf{e}_i$ co-determines connection weights, which dictate how neuron i responds to spiking inputs from an upstream population whose activities represent $\mathbf{x}(t)$. Mathematically, both these processes can be summarized as

$$a_i(t) = G[\mathbf{e}_i \cdot \mathbf{x}(t)], \tag{1.1}$$

where $a_i(t)$ is the activity of neuron i, G is the neuron model with electrical current inputs $[\cdot]$, and $\mathbf{e}_i \cdot \mathbf{x}(t)$ is the dot product between the state space inputs and neuron i's encoder. The specifics of how inputs drive the cell, and how the cell dynamically responds, depends upon the neuron model G. A distributed encoding extends the notion of representation: if $\mathbf{x}(t)$ is fed into multiple neurons, each with a unique tuning curve defined by $\mathbf{e}_i$ and other parameters, then each neuron will respond with a unique spiking pattern $a_i(t)$, and the collection of all neural activities will robustly encode the signal. Eq. 1.1 is typically extended to include two additional parameters that determine the slope and x-intercept of each individual neuron in order to capture observed neuron tuning curves:

$$a_i(t) = G[\alpha_i \mathbf{e}_i \cdot \mathbf{x}(t) + \beta_i], \tag{1.2}$$

where α_i is referred to as the gain and scales the magnitude of the neural response to an input $\mathbf{x}$, and β_i is referred to as the bias, realized by injecting a constant bias current into the neuron, and can be chosen to effectively set the state-space values $\mathbf{x}^{\text{int}}$ where the neuron will begin firing.

The NEF is agnostic about the choice of G, but we generally use spiking leaky-integrate-and-fire (LIF) neurons in our models, whose membrane dynamics $\dot{v}(t)$ are given by

$$C_m \ \dot{v}(t) = g_L(E_L - v(t)) + J(t) \tag{1.3}$$

where C_m is the membrane capacitance, g_L is the leak conductance, E_L is the leak reversal potential, and $J(t)$ is an external current injected into the membrane. This external current originates from the synapse, and its magnitude is given by the argument for G in Eq. 1.2 (i.e., $J_i(t) = \alpha_i \mathbf{e}_i \cdot \mathbf{x}(t) + \beta_i$). An additional mechanism implements the spiking behavior in the LIF neuron: once $v(t)$ reaches a threshold v_{th}, a spike event is recorded and v is held at the reset potential v_{reset} for the duration of the refractory period v_{ref}. The LIF neuron is frequently used in computational models where both biological plausibility and scalability are a concern: previous research has shown that LIF neurons (and extensions thereof) reproduce the spiking behaviors of a wide class of biological neurons [232], and it is possible to simulate networks of millions of LIF neurons on the appropriate hardware [66].

For state space representation to be useful, there must be methods to recover, or decode, the original vector from the neuron activities. The NEF defines *decoders* $\mathbf{d}_i$ that either perform this recovery or compute arbitrary functions, $f(\mathbf{x})$, of the represented vector. A functional decoding with $\mathbf{d}_i^f$ allows networks of neurons to *transform* the signal into a new state, which is essential for performing cognitive operations. To compute these transformations, a linear decoding is applied to the neural activities:

$$\hat{f}(\mathbf{x}(t)) = \sum_{i=0}^{n} a_i(t) \ \mathbf{d}_i^f, \tag{1.4}$$

where a_i is the activity of neuron i, n is the number of neurons, and the hat notation indicates that the computed quantity is an estimate of the target function. At the motor periphery, $\mathbf{x}(t)$ is retrieved using appropriate decoders, driving the model's behavioral output via muscles; within the network itself, $\mathbf{d}_i^f$ co-determines connection weights, which dictate how downstream neurons respond to spikes produced by neuron i. Connection *weights* between each presynaptic neuron i and each postsynaptic neuron j combine encoders and decoders into a single value

$$\mathbf{w} = \mathbf{e} \times \mathbf{d}^f. \tag{1.5}$$

Discrete spikes produced by G are transformed into real-valued activities a (which also correspond to post-synaptic currents) by convolving the spike train with a filter,

$$a_i(t) = \sum_T h(t - T) * \delta_i(T) \tag{1.6}$$

where T are the spike times, h is an impulse response function for the filter, and δ_i is a dirac-delta function describing the spike train for neuron i. These filters h describe the effects current-based synapses within the network: when a neuron j receives a spike from neuron i, this spike is translated into a post-synaptic current $J(t)$ that drives neuron j, perturbing its membrane voltage according to Eq. 1.3. We typically use a low-pass filter, or exponential synapse, defined by an instantaneous rise and a decay time τ.

State space transformations in the NEF are specified by describing the dynamics of the state variables,

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t) \tag{1.7}$$

where $\mathbf{x}$ is the represented state, $\dot{\mathbf{x}}$ is its derivative, $\mathbf{u}$ are inputs (external signals or upstream representations), and t is time. In our models, we typically specify these dynamics using control theory,

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \tag{1.8}$$

where A and B are transformation matrices. B governs how feedforward inputs affect the current representation (e.g., scaled addition), while A governs how recurrent inputs affect the representation (e.g., simple harmonic oscillation); both A and B are implemented through weighted synaptic connections between neurons in populations representing $\mathbf{u}(t)$ and $\mathbf{x}(t)$. These matrices can be found by solving for the appropriate decoders, as described below. While Eq. 1.8 describes linear systems, NEF methods work for nonlinear dynamics as well.

1.4.2 Optimizing or Learning Encoders and Decoders

Given this formulation, the challenge of building cognitive neural systems using the NEF reduces to (a) describing a cognitive algorithm as a dynamical system (e.g., Eq. 1.8), and (b) finding encoders and decoders such that neural connection weights implement the transformations dictated by the dynamical system. The NEF includes both top-down and bottom-up methods for finding these weights; we use both methods throughout this thesis. In the top-down method, encoders are randomly distributed across (or within) a unit hypersphere with dimensionality D (for a state space signal $\mathbf{x} \in \mathbb{R}^D$), ensuring that neurons in the population will efficiently tile the space with their tuning curves. Encoders may also be fit to neural data. Given this encoding, the NEF uses a least-squares optimization to solve for decoders that will minimize the error E between the neural estimate and the target function

$$E = \frac{1}{2}\int_{-1}^{1} \left(f(\mathbf{x}) - \hat{f}(\mathbf{x})\right)^2 dx. \tag{1.9}$$

Substituting Eq. 1.4 for $\hat{f}(\mathbf{x})$, this can be reformulated as a least-squares problem

$$A\ \mathbf{d} = f(\mathbf{x}) \tag{1.10}$$

which can be approximately solved by taking the matrix pseudo-inverse

$$\mathbf{d} = (A^T A + I\sigma^2)^{-1} A^T f(\mathbf{x}) \tag{1.11}$$

with regularization term σ. In the NEF, a solution is aquired by specifying the activity matrix A and the target vector $f(\mathbf{x})$ at numerous evaluation points which tile $\mathbb{R}^D$: A is calculated by estimating the neural activities (according to Eq. 1.1) for each neuron n at each value of $\mathbf{x}$, while $f(\mathbf{x})$ is computed directly at these points,

$$A = \begin{bmatrix} a_1(\mathbf{x}=-1) & \dots & a_n(\mathbf{x}=-1) \\ \vdots & \ddots & \vdots \\ a_1(\mathbf{x}=1) & \dots & a_n(\mathbf{x}=1) \end{bmatrix} \tag{1.12}$$

This top-down method allows modellers to analytically solve for the connection weights in a neural network that will compute a target function: it has the advantage of being pre-computed, removing any need to train the network online or with labelled examples. We use this method by default when building our neural networks, especially when we need the network to realize specific, mathematically well-defined operations, such as rescaling an input or maintaining a representation in working memory (see below).

However, the function that a neural network should compute within a larger cognitive model is not always clear, especially if the network's goal is to learn associations between stimuli or to evaluate the utility of possible actions based on environmental feedback. To deal with these situations, the NEF also includes bottom-up methods to train encoders and decoders during the simulation itself. To update the decoders, we use the Prescribed Error Sensitivity (PES) learning rule [156]:

$$\Delta\mathbf{d} = \frac{\epsilon}{n}\ a(t)\ (\hat{\mathbf{x}}(t) - \mathbf{x}(t)) \tag{1.13}$$

where ϵ is the decoder learning rate, n is the number of presynaptic neurons, $a(t)$ are the filtered activities from presynaptic neurons, $\mathbf{x}(t)$ is the state space target, and $\hat{\mathbf{x}}(t)$ is the estimate decoded with Eq. 1.4. Conceptually, Eq. 1.13 uses the error between the decoded estimate and state space target to update the decoders during the simulation. Extensive work with the PES rule in NEF networks has shown it is capable of learning decoders to compute a wide range of functions [249].

To update encoders, we sometimes use the Voja learning rule:

$$\Delta \mathbf{e} = \kappa \; a(t) \; (\mathbf{x}(t) - \mathbf{e}) \tag{1.14}$$

where κ is the learning rate. The Voja learning rule effectively shifts the encoders in a neural population towards the inputs observed by the network, ensuring that neurons will respond to the inputs present for a particular task. This is especially useful for storing and recalling associations, or for learning associative memories, in spiking neural networks [251].

1.4.3 Two Example Networks: Working Memory and Action Selection

In addition to these learning rules, researchers using NEF networks for cognitive modelling have developed a number of network components that are frequently reused to perform particular functions. Two such subsystems that we use throughout this thesis are a working memory, which is used to actively maintain a representation for manipulation by downstream subsystems, and a decision network, which is used to select the representation with the highest value among several possibilities. These subsystems are also a core part of the SPA and have been widely used in other functional brain models [64].

For working memory, we construct a "gated working memory" (gWM) originally developed by Choo [38]: this system receives an input vector $\mathbf{u}$ and maintains an estimate $\mathbf{x}$, with the relationship between $\mathbf{x}$ and $\mathbf{u}$ determined by an input gate:

$$\dot{\mathbf{x}} = \begin{cases} \mathbf{u} - \mathbf{x} & \text{if } \textit{gate} \text{ is open} \\ 0 & \text{if } \textit{gate} \text{ is closed} \end{cases} \tag{1.15}$$

When the "gate" is open, the system shifts its current representation towards the input value, effectively "loading" the current input into working memory; when the gate is closed, the system attempts to maintain the current representation without any change. Downstream subsystems may read the currently represented estimate $\hat{\mathbf{x}}$ in order to recall previous states of the system, while control subsystems may open or close the gate to load or empty the contents of working memory in a task-appropriate manner. Such a network may perform mnemonic tasks such as the oculomotor delayed response task (DRT), in which an animal must remember the 2D location of a briefly-presented visual cue for a short period of time before recalling its location [208].

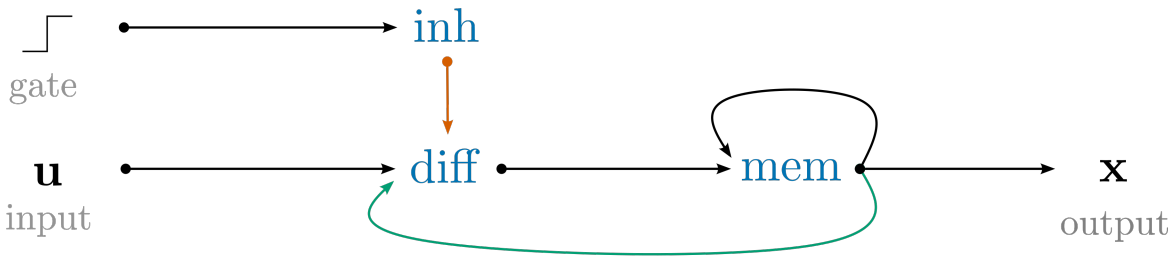

Figure 1.2: Network architecture for the gated working memory. Gray text indicates state space inputs and outputs, while blue text indicates neural populations. Black connections compute the identity function, while the green connection computes the negative of the identity function and the red connection inhibits downstream neurons.

The network used to realize the gWM is shown in Fig. 1.2. The input signal $\mathbf{u}(t)$ is represented by a population *diff*, while the gating signal $g(t)$ is represented by an inhibitory population *inh*. The gains and biases (Eq. 1.2) of the neurons in *inh* are set such that all neurons are inactive when the gate is open, but when the gate is closed ($g(t) > 0$), these neurons activate and inhibit *diff*. When the gate is open, *diff* receives both the input $\mathbf{u}(t)$ and the current representation stored in the central memory population *mem*, $\hat{\mathbf{x}}(t)$. Functional decoders $\mathbf{d}^f$ are used to compute the negative of the current memory representation, $\hat{f}(\mathbf{x}) = -\hat{\mathbf{x}}$, such that *diff* sees an effective input of $\mathbf{u}(t) - \hat{\mathbf{x}}(t)$. As long as the gate remains open, this signal is communicated to *mem*, driving the currently represented value $\hat{\mathbf{x}}(t)$ towards the input $\mathbf{u}(t)$. When the gate is closed, *diff* becomes quiet, and the recurrent connection on *mem* maintains $\hat{\mathbf{x}}(t)$ as best it can (given the imperfect representations and spike noise inherent in spiking neural networks, [65]). To access the memory, downstream systems can simply decode $\hat{\mathbf{x}}(t)$ from the spikes generated in *mem*.

For action selection or decision making, we use a network whose anatomy and function are modelled off the basal ganglia (BG), a subcortical structure that is thought to play a central role in selecting between candidate actions (for instance, between motor plans to execute a reaching movement towards one of several visual targets) [192]. Our network was originally developed by Stewart et al. [219] based off a model of BG function proposed by Gurney et al. [92]. The network is shown in Fig. 1.3: it includes separate populations for the subthalamic nuclei (STN), globus pallidus internus and externus (GPi and GPe), and striatal D1 and D2 neurons (STR D1 and STR D2). We assume that the inputs to this network originate in cortex and encode the values, or utilities, of candidate actions (see the neuroanatomical discussion in Sec. 1.3). These inputs travel through several streams within the BG, commonly referred to as the direct, indirect, and hyperdirect pathways.

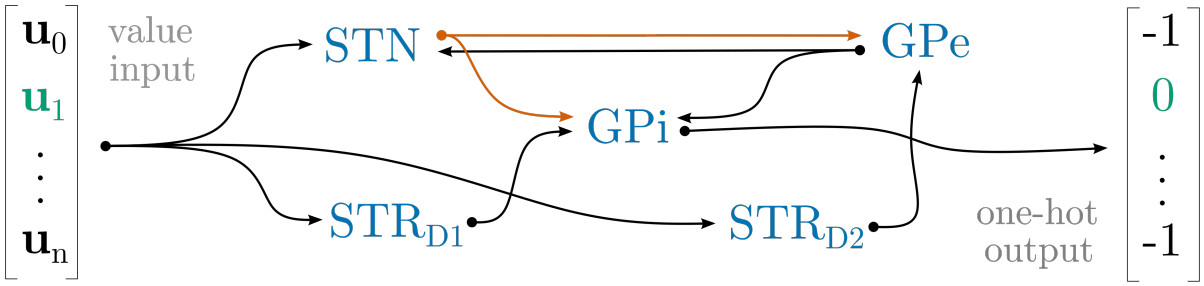

Figure 1.3: Network architecture for the basal ganglia network used for WTA selection. Text coloration and inset are as in Fig. 1.2. Functionally, this network takes a vector of values (representing, for instance, the utility of potential actions) and, through a series of recurrent connections with mutual inhibition, selects the value with the greatest magnitude. The default output is an inhibitory vector with -1 for each of the non-winning components and 0 for the winning component; this signal may be used to inhibit the non-selected action. See [219] for a detailed description of the anatomical and functional connectivity that inspired this circuit.

A detailed discussion of the interaction between these pathways, especially the balance of excitatory inputs and inhibitory control, is outside the scope of this thesis: interested readers should refer to [219] for further details. To summarize, the network as a whole acts as a winner-take-all circuit, identifying the element of the input vector with the greatest magnitude and outputting a vector that can be used to select the winning element. More specifically, for an input vector $\mathbf{x}^{\text{in}} = (0.2, 0.8, 0.5)$, the network will produce an output $\mathbf{x}^{\text{out}} = (-1, 0, -1)$, which can be used by downstream systems to inhibit all actions except the chosen one (via methods that will be discussed in subsequent chapters).

Previous empirical and theoretical work has advanced the hypothesis that inhibitory outputs from BG, in conjunction with inhibitory projections from thalamus back to cortex, are used to select between action plans by disinhibiting specific neural assemblies in cortex [133, 219]. However, recent work suggests that thalamic outputs to cortex and cerebellum may influence cortical dynamics, and the process of action selection, in more subtle ways (see [212] for a review). This work distinguishes two classes of cells within thalamus, core cells and matrix cells, which have distinct patterns of connectivity to the rest of the brain, and may facilitate different modes of processing with respect to action selection. Thalamic core cells may facilitate the execution of previously-learned action sequences by deepening basins of attraction within cortical network dynamics, effectively pushing the brain towards choosing actions that have worked well in the past. On the other hand, thalamic matrix cells may fulfil a more modulatory function through their diffuse

spatial projections to cortex, flattening attractor basins in cortex and effectively promoting variability (or injecting randomness) into cortical dynamics. Given that BG projects to matrix cells but not core cells, this new account of connectivity between BG, thalamus, and cortex challenges the assumption that inhibitory outputs from BG can be used to directly select between cortical action plans. Future work should investigate whether our BG model can be used in conjunction with more biologically plausible models of thalamus to achieve flexible control over cortical dynamics, for instance by promoting exploration versus exploitation via the differential activation of core versus matrix cells.

An alternative network for action selection is the independent accumulator, originally developed by Gosmann et al. [89]. This network is less focused on detailed anatomical reconstruction than the BG model, but has several properties that make this subsystem easier to functionally control and interpret. The network is shown in Fig. 1.4. As with the BG model, inputs to the network originate in cortex and encode the values of candidate actions. Each value is passed to a separate population (collectively labeled *acc*) that connects back to itself; this recurrent connection computes $f(\mathbf{x}) = R\mathbf{x}$, making each population act like an integrator. When driven by a vector of action utilities $\mathbf{x}^{\text{in}}$, the value represented by each population in *acc* will ramp (or "accumulate") at a rate proportional to that element of $\mathbf{x}^{\text{in}}$ and the ramp rate R. Each population also connects to a separate pool of inhibitory neurons (collectively labeled *inh*): the neuron parameters α and β for this population are set such that they only activate if their input exceeds some threshold $\mathbf{x}^{\text{thr}}$. Finally, each population in *inh* connects back to each *other* population in *acc* with an inhibitory connection. As a result, once the accumulated value represented in *acc* exceeds $\mathbf{x}^{\text{thr}}$, a WTA response is triggered, and all other values in *acc* are set to zero. Thus, for an input vector $\mathbf{x}^{\text{in}} = (0.2, 0.8, 0.5)$, the network will produce an output $\mathbf{x}^{\text{out}} = (0, 1, 0)$, a "one-hot" representation that identifies the chosen action.

The IA network is preferable to the BG network in some circumstances. The BG network simulates multiple populations and recurrent pathways in order to recreate the direct, indirect, and hyperdirect pathways. While these features make the network biologically realistic, they require a set of fixed neural and synaptic parameters. The Nengo library that we use to build and simulate neural networks [21] specifies these parameters automatically, but this leaves the user with few tools to manage representation and dynamics within the network. This is problematic because the BG network often fails to select the best action if the inputs fall outside a certain range or if they are very similar to one another; the user must add additional network components to properly rescale or separate inputs. In contrast, users can directly modify the parameters of the core IA network to realize these operations: in our models, we set (a) the neural parameters α, β, and $\mathbf{e}$ to ensure $\mathbf{x}^{\text{in}}$ is represented properly, and (b) the ramp rate R and threshold $\mathbf{x}^{\text{thr}}$ to choose the highest-

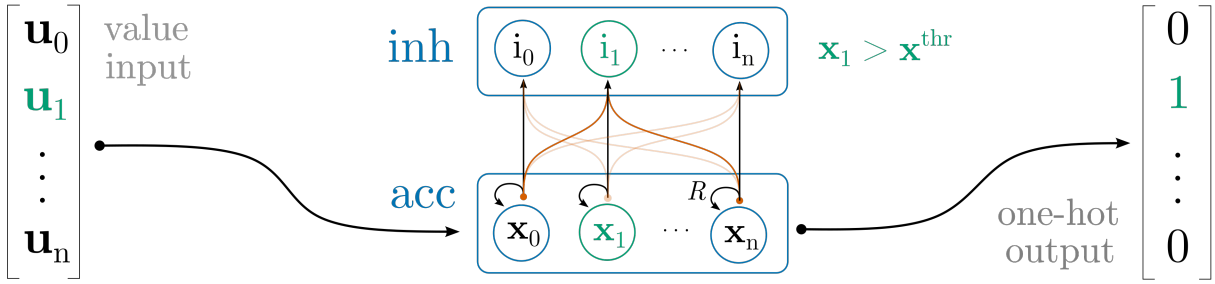

Figure 1.4: Network architecture for the independent accumulator network used for WTA selection. Blue boxes contain one neural population for each candidate action. As with the BG network in Fig. 1.3, this network takes a vector of input values and, after realizing the internal WTA dynamics, selects the action with the greates value. WTA dynamics are driven by the independent accumulation of each value: each population i in *acc* represents a value $\mathbf{x}_i$ that ramps over time according to $\mathbf{u}_i$ and a shared ramp rate R. *acc* drives populations of inhibitory neurons *inh*, which remain inactive until $\mathbf{x}$ crosses a threshold $\mathbf{x}^{\text{thr}}$. Once a population in *inh* is activated, it inhibits all other populations in *acc*, leaving only a single population in *acc* with nonzero activity. Decoding a one-hot output vector from these activities is trivial.

value action with greater probability. Manipulating the ramp rate and threshold of the IA network also provides an important form of cognitive control: we explore how these parameters relate to speed-accuracy tradeoffs in simulated decision makers in Ch. 3.

The IA model also bears many similarities to drift diffusion models (DDMs) [190], a popular family of decision making models that successfully predict simple perceptual decisions, account for reaction times, and correspond to brain activation [101, 191]; DDMs have even been applied to social decision making tasks [196]. DDMs assume that the decision maker continuously accumulates noisy information about competing choice options until an internal decision boundary is reached, at which point a choice is made. Information often takes the form of an internally-computed value signal, and thresholds are a subjective decision criteria that may be adjusted based on personal factors or task demands. Thus, although the IA does not recreate particular anatomical regions in the brain, its form and function have been extensively validated by way of DDMs, and it is consistent with general cortical connectivity.

1.5 Thesis Outline

In this chapter, we motivated the study of social decision making, reviewed the functional neuroanatomy of social cognition in the brain, and introduced a theoretical framework for building biologically-plausible computational models. In the next four chapters, we develop and analyze four cognitive models built as spiking neural networks. Each of these models investigates a different aspect of learning or decision making.

In Ch. 2, we model associative learning in the amygdala, then investigate the neural mechanisms of fear conditioning and extinction. We pay particular attention to anatomical detail: we recreate various nuclei within the amygdala, specify their connectivity, and simulate the effects of neurotransmitters including oxytocin, dopamine, and serotonin. We validate the model by comparing its behavior to conditioning studies in mice, and find that our pharmacological interventions recreate many known effects in the empirical literature.

In Ch. 3, we model biologically-detailed neurons and synapses and construct a detailed working memory model. Simulating and training this model requires extending the NEF, but allows us to investigate the effects of low-level biological perturbations, such as those thought to underlie certain forms of social neuromodulation. We validate this model by showing our detailed networks retain functional capacity, then show that the mnemonic properties of the working memory model resemble those of simple animals.

In Ch. 4, we model the speed-accuracy tradeoff in decision making by building a network that realizes value accumulation, value modulation, and action selection. This model is informed by theories of inference and decision making under time pressure, and includes several cortical and subcortical brain areas. We validate the model by comparing its behavior to a dataset from humans performing an identical task, and find that the model recreates several trends in the human data.

In Ch. 5, we model learning and decision making in a social game by building cognitive agents that realize value estimation, value updating, and action selection. Our models represent social information in high-dimensional spaces and use reinforcement learning to update proself and prosocial estimates of action values. We validate the model by conducting a human experiment, training a population of unique agents, and comparing human behaviors to agent behaviors; we find that our agents reproduce several trends related to social value orientation.

In Ch. 6, we summarize our contributions to the field of social cognition. We compare our work to other computational models, discuss its practical and theoretical limitations, and propose avenues for future research.

Chapter 2

Fear Conditioning and the Amygdala

Author's Note: some of the content in this chapter was previously published as a poster at the 2019 Society for Neuroscience conference [53]. Code is available on GitHub.

2.1 Introduction

To make intelligent social decisions, agents must learn which features of the environment, and which potential actions, are associated with positive and negative outcomes. The brain contains many distinct learning systems that work together to achieve this goal. In the previous chapter, we discussed how various cortical and subcortical areas might realize value-based decision making. An important step in evaluating actions and states is recognizing which features are associated with positive or negative outcomes. In "associative learning", agents build models that predict the co-occurrence of external features, including associations between neutral stimuli and positive or negative stimuli. These associative models can be used to evaluate basic actions, such as approach and avoidance, which manifest in numerous social and non-social contexts. Associative models can also be used in model-based computations, which use explicit knowledge about the dynamics and correlations within an environment to predict how an agent's actions will affect changes that promote pleasure or avoid pain. Thus, while associative learning does not directly evaluate the quality of potential actions, it is still essential for value-based decision making.

In this chapter, we develop and investigate a neural model of fear conditioning in the amygdala. Fear conditioning is a particular form of associative learning in which an agent learns to associate a neutral stimulus with a negative stimulus, causing the agent

to exhibit a fear response when the neutral stimulus is present. In evolutionary terms, fear conditioning has high adaptive value: agents that can learn to avoid life-threatening stimuli in the environment are more likely to survive. This makes fear conditioning one of the most common forms of associative learning in the animal kingdom, and one of the most widely studied (and modeled) types of learning in neuroscience and psychology. The amygdala (AMY) is an important locus of fear conditioning in the brain: it undergoes synaptic plasticity during fear conditioning, and lesions impair both learning and expression of conditioned fear responses. Furthermore, AMY is an affective system that coordinates emotional and hormonal responses to salient stimuli, effectively modulating value-based computations. With respect to social decision making, AMY learns associations between specific people (e.g., faces or identifying attributes) and valued outcomes, and is important in evaluating trustworthiness and directing primitive social actions (e.g., reproduction, parenting, aggression, and submission). For these reasons, AMY is widely understood to play an critical role in learning and decision making in social contexts.

Our model of fear conditioning in the amygdala seeks to be both biologically detailed and functionally capable. Anatomically, our network contains many nuclei, whose internal and external connectivity are appropriately constrained; we show that empirically-consistent response profiles of these neurons develop naturally over the course of training. Functionally, our network is trained using online, local, error-driven learning rules, and is capable of learning a variety of associations between neutral stimuli, negative stimuli, and environmental contexts. As a result, our network exhibits fear conditioning, fear extinction, and fear renewal when trained under various traditional experimental protocols. We show that externally activating or inactivating specific nuclei impairs these processes, and discuss the extent to which these results align with empirical experiments. Finally, we investigate fear generalization; we demonstrate that our model exhibits gradients of fear responses to similar stimuli that resemble the gradients observed in empirical studies.

We begin by introducing the functional and anatomical properties of the amygdala in greater detail. We then introduce two versions of the model: the first includes less anatomical detail and contains the minimum neural circuitry required to learn the desired functionality; the second includes greater anatomical detail and is able to reproduce a wider variety of empirical data. We describe the various training protocols we use, then simulate the networks and analyze the results, comparing to empirical data where possible. We conclude by discussing the successes and failures of the model, and by identifying directions for future work.

2.1.1 Fear Conditioning Protocols and Terminology

In a traditional fear conditioning experiment, an animal is placed into some environment, or "context". To establish a baseline for future testing, a neutral stimulus is presented for a short duration: stimuli may be auditory (tones of a specific frequency), visual (images of faces, objects, or natural scenes), or even olfactory (distinct odors). This stimulus is referred to as the "conditioned stimulus" (CS) because the animal will later be conditioned to fear its presence. The experimenter measures the baseline fear response of the animal; different animals display different fear responses in different settings, but common metrics include freezing (cessation of movement to avoid attracting predators), startle responses (pupil dilation, involuntary movements, etc.), and physiological measures (galvanic skin response, heart rate, etc.).

Once a baseline has been established, conditioning begins. During the "aquisition" or "conditioning" phase, the CS is presented for a duration, and a negative (or "aversive") stimulus is also presented for a duration. The negative stimulus, or "unconditioned stimulus" (US, so-named because the animal will exhibit the fear response to its presence by default), may be mild pain (electric shock), unpleasant odors, scary images, or any other stimuli that the animal would naturally avoid. The duration, ordering, and overlap of the CS and US are experimental parameters, but the most common setup is to present the CS for a duration, immediately followed by the US. Both stimuli are then removed. This constitutes a single "pairing" of the CS and US; typical fear conditioning experiments consist of multiple pairings to robustly reinforce the association (although one-shot learning is possible for some animals in some contexts). As acquisition proceeds, the animal learns that the CS predicts the onset of the US, and begins to exhibit the fear response to the CS itself. During a "fear expression" test, the CS is presented alone (without the US), and the animal's fear response is again measured; if the response is statistically greater than it was during the baseline test, fear conditioning has occurred. Fear conditioning may also occur to the context itself: when placed in the environment where acquisition occurred, the animal may exhibit a moderate fear response, even when no CS is presented.

Interestingly, conditioned responses to the CS are notoriously difficult to unlearn; they persist over a long duration (sometimes an entire lifetime for traumatic experiences), and explicit efforts to unlearn the association are only effective in certain situations. Training associated with such unlearning is referred to as "extinction", and proceeds as follows. The animal undergoes fear conditioning in a context (CTX+), then the animal is placed in a new context (CTX-, a new environment with readily distinguishable features), and a second round of training occurs. The CS is repeatedly presented without the US, and over time, the animal's fear expression to the CS diminishes (this typically requires more pairings

than does fear acquisition). However, neural and behavioral evidence shows that extinction training does not actually unlearn the original CS-US association: rather, the animal learns a positive (safety) association that competes with the negative (fear) association learned during acquisition, cancelling out the fear response in some circumstances, but not others. When fear expression is tested in CTX-, little or no fear response is observed; but when the animal is returned to CTX+ and presented the US, it exhibits the fear response at nearly full strength. This process is called "fear renewal", and demonstrates that fear extinction is context-specific.

Finally, "fear generalization" refers to the tendency for animals to exhibit fear responses to stimuli or contexts similar to those present during conditioning. In one popular paradigm, researchers show participants circles of two sizes during training: larger circles (CS+) are paired with the US, while smaller circles (CS-) are not associated with the US. During testing, the researchers present circles whose sizes range between the CS+ and the CS-: they typically observe a gradient of fear responses, indicating that fear expression diminishes as stimuli become increasingly dissimilar to the CS+ [145, 60]. Alternatives of this paradigm exist for contextual conditioning. In one experiment, researchers put humans in a virtual reality room (CTX+) containing various objects, and periodically presented a US as they explored the room; participants also explored a safe room (CTX-) containing different objects, in which the US was not presented. The researchers then placed participants in a new room that shared 50% of the objects from CTX+ and 50% from CTX-; they observed fear responses that indicated participants felt partly afraid and partly safe [7]. These experiments suggest that fear learning acts on a sub-symbolic level of abstraction: stimuli and context that share features with the CS+ and CTX+ elicit partial fear responses.

2.1.2 Neuroanatomy

The amygdala is a small structure located in the temporal lobes, adjacent to the hippocampus, that contains several nuclei, including the lateral amygdala (LA), basolateral amygdala (BLA), central lateral amygdala (CeL), and central medial amygdala (CeM). The internal and external connectivity of these nuclei are critical for fear conditioning. AMY nuclei contain a mix of excitatory pyramidal neurons and inhibitory interneurons; most nuceli contain various cell types and neurotransmitters, making the neural substrates of AMY resemble both cortex and basal ganglia [161]. Several dedicated populations of interneurons surround and separate the LA/BLA and the CeL/CeM; these "intercalated cells" (ITC, as well as interneurons distributed within the nuclei) are also important for

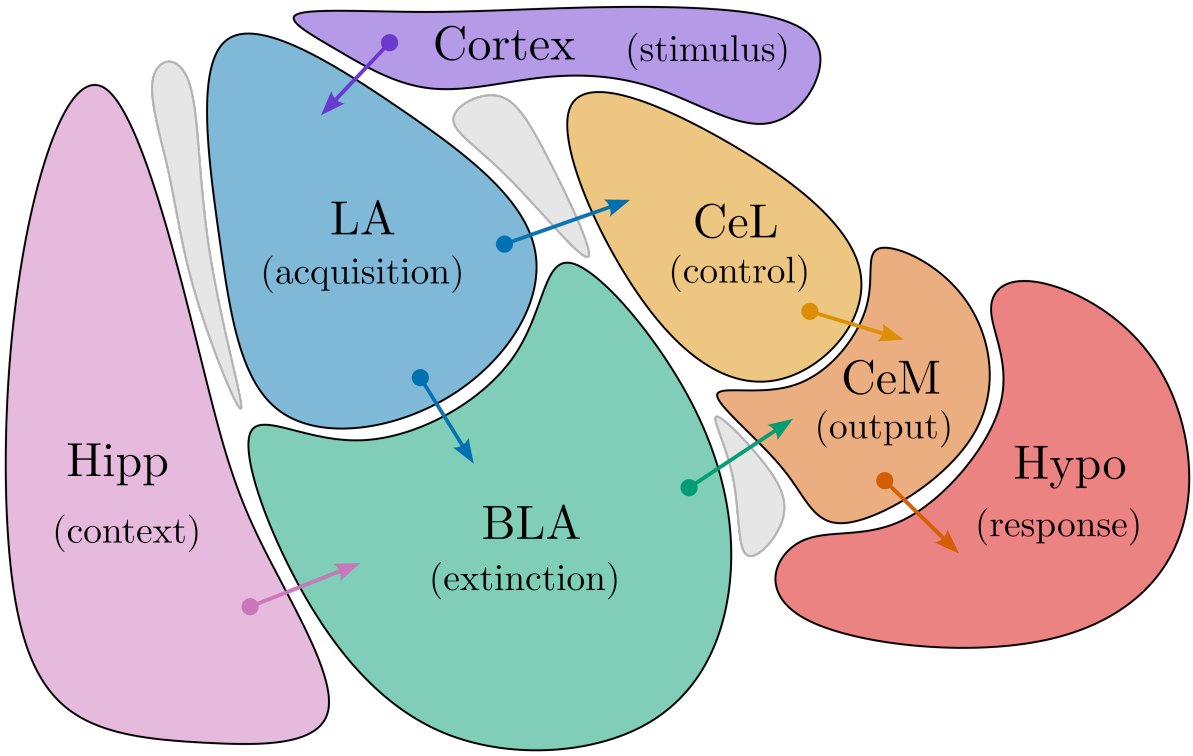

Figure 2.1: Sketch of amygdala neuroanatomy. Nuceli abbreviations: lateral amygdala (LA), basolateral amygdala (BLA), central lateral amygdala (CeL), central medial amygdala (CeM). External population abbreviations: hippocampus/MTL/mPFC (Hipp), sensory cortices (Cortex), hypothalamus/periaqueductal gray/sympathetic/parasympathetic (Hypo). The gray regions are intercalated cells (ITC).

fear conditioning [153, 131], but their exact functional role is less clear. Fig. 2.1 provides a rough sketch of the neuroanatomy and connectivity of these nuclei.

LA receives connections from sensory cortices (sometimes through intermediary structures) that convey information about the external world. The synapses between long-range cortical axons and the dendrites of pyramidal cells in LA are the primary site of fear learning; plasticity within these synapses induces LTP in response to coincident CS and US, increasing the sensitivity of (some) LA neurons to the CS and driving downstream fear responses [131, 62, 193]. Artificially inactivating LA neurons during training or testing impairs fear learning [173]. LA pyramidal neurons project almost exclusively to BLA and to CeL; they do not convey fear responses to anywhere outside AMY.

BLA receives few connections from sensory cortices, but is reciprocally connected to several structures in the medial temporal lobe (MTL) (including HIPP) [161], and mPFC (including the infralimbic and prelimbic cortices, IL and PL) [160]. These connections convey high-level contextual information to the BLA, including episodic memory from HIPP and personal identifiers from MTL. The BLA coordinates and controls fear responses based on this contextual information [62, 27, 248, 35]. In particular, BLA is the primary site of both context-dependent conditioning and extinction: synaptic plasticity on BLA pyramidal neurons and/or inhibitory interneurons is associated with increased fear responses in environments where painful stimuli were experienced, and with decreased fear responses in environments where painful stimuli were expected but not experienced. Pyramidal neurons within BLA exhibit various responses during fear conditioning and expression. BLA "fear neurons" become more active (responsive to CS inputs) during fear conditioning, but become less responsive during extinction training. In contrast, BLA "extinction neurons" do not respond to the CS after conditioning, but become responsive during extinction training. It seems likely that, following extinction, these neurons activate interneurons within BLA that inhibit fear neurons, suppressing the fear response in safe contexts. However, not all fear neurons in BLA are suppressed: "persistent neurons" are a small subset of BLA neurons that become CS-responsive during fear conditioning, but remain responsive after extinction. BLA neurons do project to brain areas outside AMY: apart from the reciprocal connections to temporal lobe mentioned above, BLA outputs also reach striatum, which in turn projects back to cortex, suggesting triangular connections between BLA, cortex, and striatum [161]. Through these connections, BLA is implicated in affective labelling, goal-directed behavior, planning, and decision making [167]. BLA also connects to CeM, and may connect indirectly to CeL through clusters of interneurons [62].

CeL receives some connections from outside amygdala, mostly from noncortical structures such as the thalamus and brainstem, as well as projections from the LA and from clusters of interneurons. The responsiveness of CeL neurons changes over the course of fear conditioning, with some neurons showing increased responses to the CS, and others showing decreased responses. Several lines of evidence suggest that these changes in responsiveness reflect synaptic plasticity within CeL, rather than simply transmitting learned responses in LA: CeL may have access to both CS and US information via extra- and intra-amygdalar inputs, and it displays synaptic and neurochemical changes following conditioning [120]. It is unclear whether these synaptic changes occur on connections from LA, or from areas outside amygdala. Unlike LA and BLA, CeL mostly consists of GABAergic interneurons; it appears that an inhibitory circuit within this nucleus controls fear responses, and damaging or inhibiting this area significantly impairs fear conditioning [120, 35]. Specifically, baseline activity within CeL (so-called CeL-off neurons) appears to inhibit CeM, preventing default

fear responses. However, artificially inactivating CeL-off neurons (or activating CeL-on neurons that inhibit CeL-off neurons) produces an unconditioned fear response (freezing) [40], suggesting that CeL acts as a controllable gate for CeM-driven fear responses.

CeM is the major output hub of AMY: it receives connections from BLA and CeL that convey fear and safety signals, and projects to numerous subcortical structures that drive basic fear responses. These targets include the hypothalamus, which controls endocrine responses in the brain and body; the periaqueductal gray, which directs behavioral responses (like freezing) to internal and external stressors; and numerous other structures in the sympathetic, parasympathetic, and hormonal systems that control bodily responses [161]. Through the CeM, the AMY is able to regulate the body's state, to modulate large portions of the brain, and to direct behavioral responses [167]. In fear conditioning experiments, CeM appears to integrate the responses learned in LA, BLA, and CeL to arrive at an overall fear estimate [62]. Consistent with this theory, activating CeM before training produces unconditioned freezing, while inactivating it after training reduces fear responses.

Many other neural populations play some role in fear conditioning, extinction, and expression. These include MTL and cortical structures like IL and PL, as well as clusters of inhibitory interneurons like the intercalated cells (ITC). For functional and anatomical simplicity, we will ignore these nuclei in our model, but we will comment on their possible contribution to associative learning in Sec. 2.4.

2.1.3 Function

Broadly speaking, AMY is part of the brain's "affective system": it helps organisms decide which of their goals to pursue in the current context. This is achieved by using associative learning to assign "salience" to objects, people, and events in the wider world. Organisms routinely experience stimuli that have intrinsic or extrinsic value: "appetitive" stimuli like food and sex are naturally desirable, while "aversive" stimuli like pain are undesirable. These "valenced" stimuli should be approached, avoided, or otherwise acted upon to increase an organism's chances of survival and reproduction. The AMY assigns salience to (previously) neutral stimuli by associating their presence with these valenced stimuli and with intrinsic behavioral responses to valenced stimuli. In fear conditioning, AMY learns to produce an intrinsic (or "unconditioned") fear response (UR, e.g., freezing) in response to the neutral CS when it is reliably paired with the aversive US. Conceptually, this process may be described as either associating the CS with the UR directly, or as associating the CS with the US, which is intrinsically associated with the UR [167]. CS-UR associative learning guarantees that an organism will rapidly develop responses to salient stimuli: because such learning is independent of contextual information, AMY responses will drive

the UR in many circumstances, guaranteeing a reliable "better safe than sorry" response. Learning within CeA (CeL and CeM microcircuits) may realize this type of associative learning: inputs from brainstem and thalamus convey simple representations of US and CS, while outputs to hormonal and basic behavioral systems direct the UR.

As cortex evolved to represent and plan around more sophisticated representations of the external world, AMY may have evolved more sophisticated mechanisms of assigning salience and controlling behavior. High-dimensional representations in sensory cortex are conveyed to LA, which can learn to associate these CS with various types of US; heightened responses in LA may then drive CeM (via CeL or BLA) to activate the UR. The advantage of this pathway is that it uses more complex representations of the CS, allowing an organism to distinguish stimuli that predict the US from those that are present by happenstance. Later, we will discuss how cortical complexity supports fear generalization.

A third layer of complexity to associative learning is introduced by the BLA, which helps AMY further distinguish salient features of the environment from irrelevant distractors. BLA receives sophisticated representations about an organism's overall context from MTL and mPFC: these signals may convey the organism's current goals (e.g., preserving social reputation by appearing brave when confronted with danger) or high-level assessments of the environment (e.g., reasoning that a climbing harness indicates protection from lethal falls, but only when properly equipped and supported by a trusted climbing partner). However, it is difficult to directly account for goal-related and contextual information when learning CS-US associations: learning three-way associations between CTX, CS, and US is more complicated (we suspect) than learning a series of two-way relations between CTX/CS and CS/US. More importantly, incorporating CTX information directly into CS-US learning rules may cause an organism to incorrectly identify situations as non-salient, which may be a life-threatening mistake. Neural and behavioral evidence suggest that AMY circuitry instead uses BLA to learn separate CTX associations that suppress fear responses: LA associations between CS and US are preserved during extinction training, but BLA learns a second memory trace that uses CTX information to correct false-positives (or otherwise supplement the primary CS-US association). This "better safe than sorry" approach ensures that, if the CTX association turns out to be wrong, or if more complicated contingencies arise, the organism does not have to relearn the original CS-US association; it simply amends the secondary memory trace in BLA.

This account of CTX learning in BLA explains several phenomenon in an evolutionarily-consistent manner. First, it minimizes the likelihood that the organism will have to repeatedly endure painful CS-US associations: these memory traces persist despite downstream learning. Second, the context-specificity and limited generalization of extinction training (evidenced by fear renewal and context generalization experiments) are consistent with the

principle that situational salience cues should not override stimulus-based salience cues, but merely suppress them in limited situations. Third, it explains the post-training response profiles of LA versus BLA neurons. LA neurons continue spiking in response to learned CS after extinction training, and this activity continues to drive a fear response. On the other hand, BLA neurons show diverse response profiles following extinction training (fear neurons, extinction neurons, and persistent neurons): their relative responses can be combined into a contextually-sophisticated salience assessment, possibly implementing a simple form of model-based control over fear expression [185]. Cortical areas like MTL and mPFC, whose hierarchical neural architectures and high-dimensional representations are well-suited for model-based reasoning, may also support complex associations between CS, US, and CTX, and relay associative information to AMY through the BLA.

Taken together, this functional hypothesis explains the roles of the various AMY nuclei, and shows how parallel learning rules may coordinate diverse responses to salient stimuli. This account is generally consistent with other models of AMY function, both conceptual [167] and computational [62, 35]. In the next two sections, we implement two spiking neuron models that realize this hypothesis: the first includes less biological detail, but has a clear functional correspondence to the populations and learning rules presented above; while the second expands this minimal model to include more biological detail, while keeping the same anatomical and functional divisions intact. After training the models and analyzing the results, we compare our model(s) to other computational and functional accounts of the amygdala.

2.2 Computational Models

2.2.1 Minimal Functional Model

We begin by describing a relatively-simple model of AMY function that should produce fear conditioning, extinction, and renewal. The anatomical organization of this model maps broadly onto the divisions between AMY nuclei discussed above, but the model's activity profiles and its responses to external perturbation lack empirical realism, motivating the design of a second model.

Fig. 2.2 gives the network architecture for model 1. Three neural populations, labelled $external_{\text{CS}}$, $external_{\text{US}}$, and $external_{\text{CTX}}$, lie outside AMY and coordinate CS, US, and CTX inputs; and one neural population labelled *fear* represents the overall fear response; its decoded activity constitutes the model's output. The model contains one central population labelled *BLA*: it is a three-dimensional population whose activities represent mixed

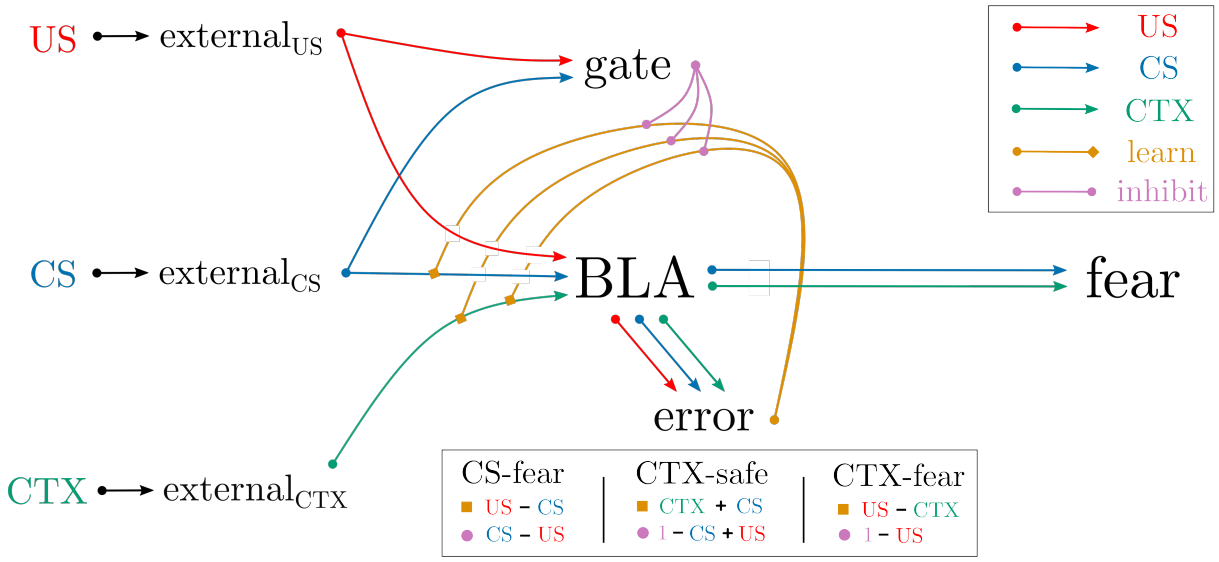

Figure 2.2: Minimal functional model of fear conditioning in the amygdala. Black text indicates neural populations containing 100-300 spiking LIF neurons. Coloration indicates the communication of information (red for US, blue for CS, green for CTX) or the application of a particular function (orange for PES learning, pink for direct inhibition). For simplicity, we draw *error* as a single population, but it actually consists of three distinct error populations, each of which implements a distinct learning rule and is gated by a distinct connection from the *gate* population. The quantities computed by these populations, and the conditions in which gate inhibits them, are indicated in the legend.

responses to CS, US, and CTX. Finally, there are three one-dimensional error populations that are used to update CS and CTX associations, plus a two-dimensional gate population *gate* that is used to control learning updates.

Input signals conveying CS, US, and CTX information drive the *external* populations. The CS signal has dimensionality dim_{CS} (default value is 3) and is presented for a duration of 1 second, followed by 1 second of silence (all zeros). The US signal is one-dimensional (indicating the presence or absence of the US), and is presented alongside the CS during acquisition training (1s on, then 1s off). The CTX signal has dimensionality dim_{CTX} (default value is 5) and remains constant during acquisition training, then changes during extinction training or renewal testing. To generate the vectors corresponding to CS and CTX, we drew samples from the surfaces of hyperspheres with dimensionality dim_{CS} or dim_{CTX}: this ensures that all CS and CTX vectors are unit length, which in turn ensures

that the responses of the *external* populations to different CS and CTX are approximately equivalent, even though different subsets of neurons within these populations will activate as the external cues change.

The external populations all connect to *BLA*: $external_{\text{CS}}$ and $external_{\text{CTX}}$ connection weights are initialized to zero, and these weights are learned during acquisition training or extinction training; while the $external_{\text{US}}$ connection into *BLA* simply computes the identity function (passes the US information directly). During acquisition training, the error population $error_{\text{CS-fear}}$ computes the difference between the US-induced response of *BLA* neurons and the CS-induced response of *BLA* neurons. If their difference (US minus CS) is positive, this indicates that a US is currently active, but that *BLA* is not responding to the coincident CS. In this case, learning changes connection weights from $external_{\text{CS}}$ to *BLA* such that *BLA*'s CS response increases. This is achieved by using the PES learning rule (Eq. 1.13) to update decoders on the $external_{\text{CS}}$ to *BLA* connection based on the error signal provided by $error_{\text{CS-fear}}$. Following fear conditioning, presenting the CS alone should produce a CS-response in *BLA*.

Unfortunately, with this setup, repeated presentation of the CS without the US will cause direct unlearning of the CS-US association. This is because $error_{\text{CS-fear}}$ represents *BLA*'s US response minus *BLA*'s CS response: when the former is absent and the latter is present, this signal will become negative, and a negative value sent to the PES learning will produce negative changes in the decoders, unlearning the original association. To prevent this, learning must cease when the CS is present without the US. This can be achieved in two ways: by initializing the neural tuning curves in $error_{\text{CS-fear}}$ such that this population only represents positive values (i.e., activities naturally drop to zero when the error becomes negative); or by placing an inhibitory gate on $error_{\text{CS-fear}}$, such that it is externally inhibited when the CS is present in isolation. We chose the later architecture, for reasons that will become apparent later.

Fig. 2.3 shows how these components interact to implement fear acquisition with repeated CS-US pairing. During the baseline test, the CS is presented in isolation, but neither the *BLA* nor the *fear* population shows a response. Next, five CS-US pairings are presented: *BLA*'s CS response initially starts at zero, leading to a large error and rapid updates of the $external_{\text{CS}}$ to *BLA* connection. By the fourth pairing, the CS response has risen to meet the US response, and learning becomes negligible as the error goes to zero. Next, we switch the context from CTX+ to CTX-, then repeatedly apply the CS for a fear expression test. When the CS is present without the US, the gate becomes active, and the error population is inhibited: the total lack of activity in $error_{\text{CS-fear}}$ is decoded as zero error. This prevents unlearning, so CS responses are retained throughout the test.

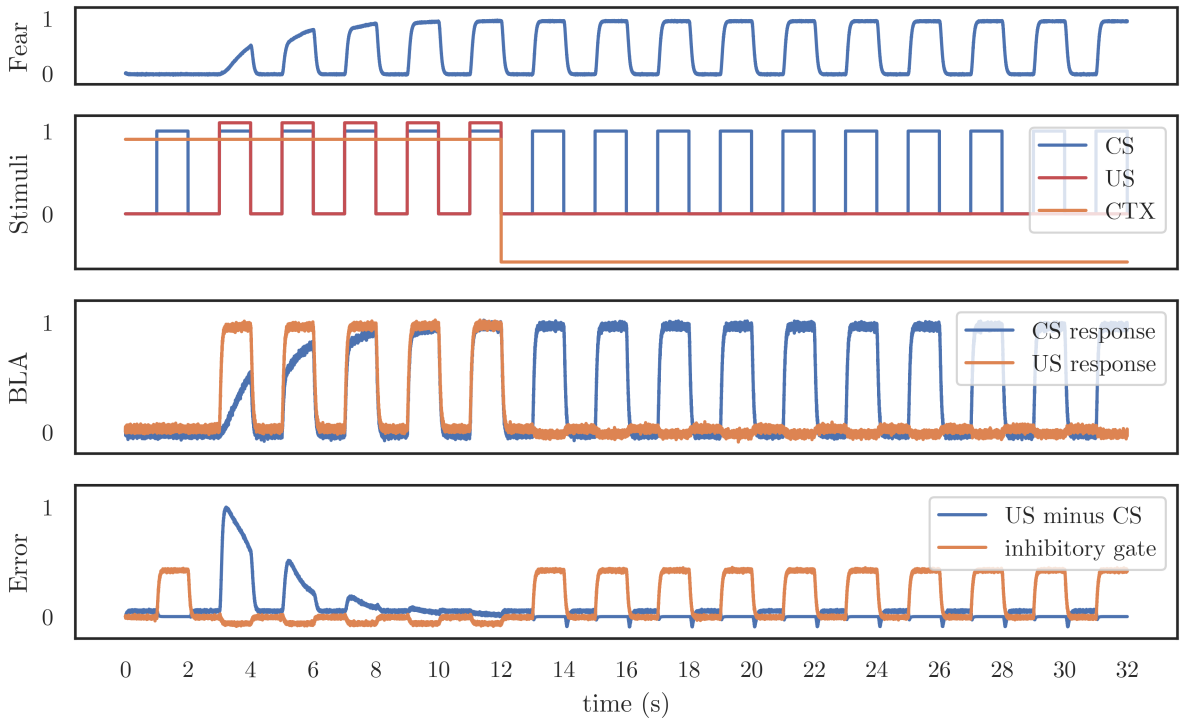

Figure 2.3: Fear conditioning in BLA during acquisition training and fear expression testing. The first panel plots the combined fear response produced by the network, while the second panel indicates when CS, US, and CTX are applied. Plotted CS values indicate the presence of the multidimensional CS signal, while plotted CTX values are the dot product between the current CTX and CTX+. The third and fourth panel indicate the decoded state from the *BLA* and $error_{\text{CS-fear}}$ populations. During acquisition, PES learning causes *BLA*'s CS to rise until the value represented in $error_{\text{CS-fear}}$ (US minus CS) reaches zero. After the context switch, no US is applied, and the gate population inhibits $error_{\text{CS-fear}}$, preventing unlearning.

Next, we add context learning between $external_{\text{CTX}}$ and *BLA*. If the combined CS and CTX responses of *BLA* is positive, this indicates a combined fear response; if it is zero, this indicates a lack of fear response. During acquisition, CS responses become high; during extinction, we need to train *BLA*'s CTX response to become negative, to suppress this fear response. The error population $error_{\text{CTX-safe}}$ computes the summation between the *BLA*'s CS response and its CTX response, contingent on the presence of the US, and drives PES learning on the decoders from $external_{\text{CTX}}$ to *BLA*. As a result, when (CS + CTX) is positive, the decoders from $external_{\text{CTX}}$ to *BLA* decrease, causing *BLA*'s CTX response to become more negative. However, this learning needs to be contingent on the US not being present: otherwise, this learning rule will cause contextual safety learning during acquisition training. To account for this contingency, we again use the *gate* population to control learning: *gate* inhibits $error_{\text{CTX-safe}}$ by default, and only releases inhibition when the CS is present but the US is not. Finally, *BLA*'s CS and CTX responses are summed and sent to the fear population.

Fig. 2.4 shows how this network realizes fear conditioning and extinction at different stages of training. The baseline and fear acquisition phases proceed as before: the extinction gate keeps any contextual learning from occurring before the context switch. During extinction training, the context input is switched to a new high-dimensional vector labelled "CTX-", and the CS is presented without an accompanying US. *BLA*'s CTX response begins at zero, but steadily decreases as $error_{\text{CTX-safe}}$ drives PES learning. By the end of the extinction phase, *BLA*'s CTX response has reached a value around -0.5; when added with the *BLA*'s CS response of +1.0, this produces only a minor fear response of +0.5. Changing the learning rate, or the degree of inhibition by the extinction gate, changes the speed of contextual extinction. Thus, the model may capture individual variability in contextual fear suppression with only a few parameters.

Finally, we introduce a third learning rule and error population, which also target the connection between $external_{\text{CTX}}$ and *BLA*. This third connection allows context-based fear conditioning, in which *BLA* learns to directly associate the CTX and US (no CS is required). Empirical data show that, when a US is intermittently delivered while an animal is present in CTX+, the animal will produce a moderate fear response whenever it is returned to CTX+. However, the extinction-based learning rule introduced above does not support this behavior, since it operates on differences between *BLA*'s CS and CTX response. To compensate, we add a third error population, $error_{\text{CTX-fear}}$, which computes the difference between *BLA*'s CTX and US responses, and drives PES learning on the $external_{\text{CTX}}$ to *BLA* connection. Unlike the CS-US learning rule, we want this connection to support both fear responses and safety responses: learning will not be restricted to the positive domain. Still, we do want fear learning to proceed faster than safety learning,

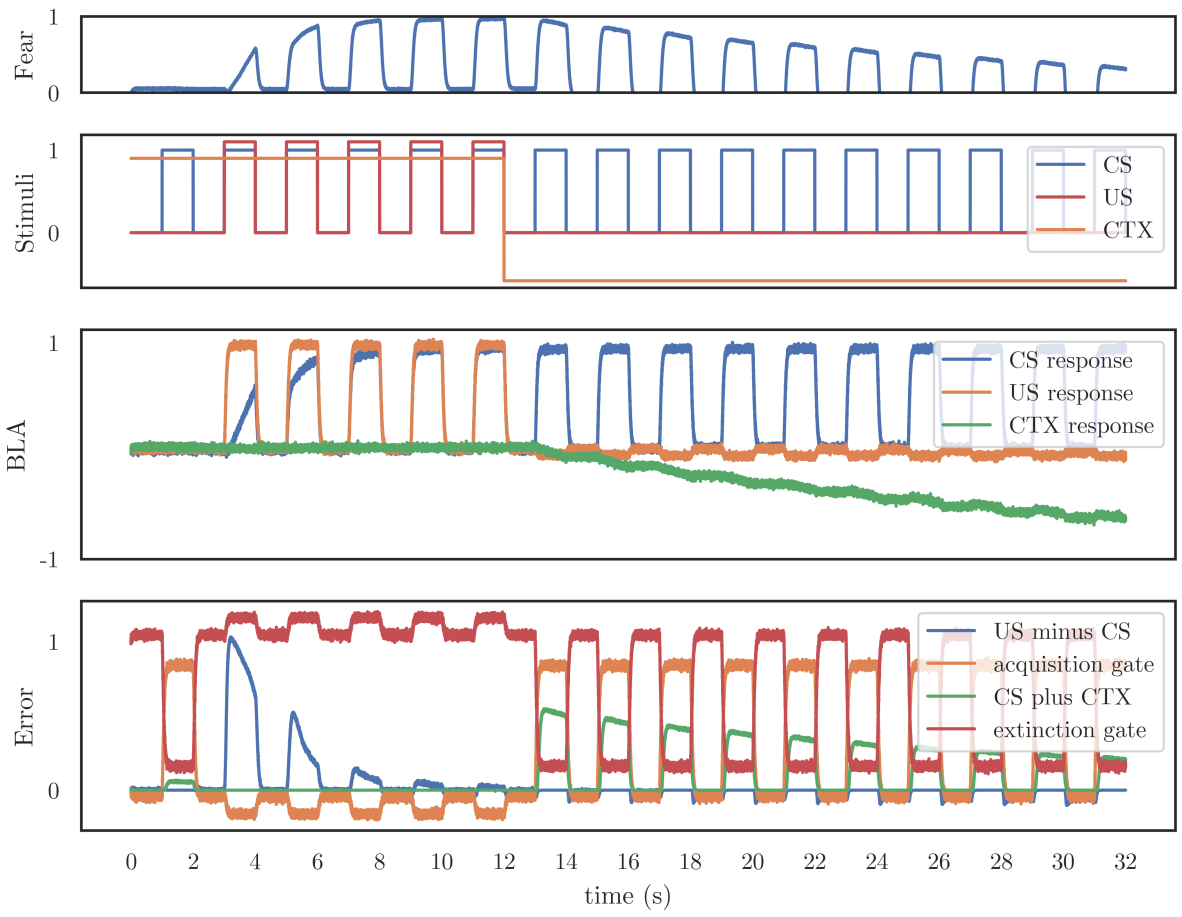

Figure 2.4: Fear extinction in BLA. After fear acquisition has occurred and the context switches from CTX+ to CTX-, $error_{\text{CTX-safe}}$ computes the value (CS + CTX) and drives PES learning between $external_{\text{CTX}}$ and *BLA*. This results in a gradual decrease of *BLA*'s CTX response, which competes with *BLA*'s CS response to suppress the overall fear response. The *gate* population inhibits this learning whenever the CS is not present, and whenever the US is present.

otherwise *BLA* may unlearn the CS-US associations during the 1s pauses between pairings we enforce during acquisition training. We achieve this by once again using the *gate* population: *gate* slightly inhibits $error_{\text{CTX-fear}}$ by default, but inhibition is removed when the US is present. This lets *BLA* learn CTX-US associations quickly, but also lets it unlearn them slowly (if the animal experiences CTX for a long duration without the US appearing).

Fig. 2.5 shows the behavior of the complete functional model. During acquisition in CTX+, *BLA* acquires both a strong CS and a moderate CTX+ response, leading to strong fear responses. During extinction, *BLA* acquires a moderate CTX- response that competes with the CS response to suppress fear, leading to a weak fear response. Following extinction, we simulate three fear expression tests: we present the CS without the US in CTX+, CTX-, and a novel CTX. Note that we externally stop all learning during the expression tests, to eliminate spurious ordering effects from continued learning during testing. As expected, baseline fear responses are minimal, while responses to the CS in CTX+ are maximal; fear responses in CTX- are low but nonzero, and fear responses in a novel context are high, but not as extreme as with CTX+.

We next generated a large dataset for analysis by simulating 100 unique instances of model 1. When initializing an NEF network, each population and connection is assigned a random number seed, which is used to sample encoders, gains, biases, etc. from the appropriate distribution. To generate many unique instances of model 1, we simply choose a different seed for each model, leading to unique tuning curve distributions within each neural population. We also assign each model instance a different learning rate for each of the three learned connections (by again sampling from distributions according to the random seed): these parameters determine how fast each model instance learns the various associations. Our models were trained using the same "ABX" protocol, which is a common protocol in fear conditioning experiments [151]: fear acquisition (5 pairings of CS and US) occurred in CTX A, followed by fear extinction (10 CS presentations) in CTX B, followed by tests in some context X. We collected spike data from *BLA* and measured fear response by decoding spikes from *fear*. In the following figures, we plot the variance of the measured quantities over the 100 model instances, visualized as 95% bootstrapped confidence intervals.

Fig. 2.6 shows mean fear response in response to the CS in CTX+, CTX-, and CTXn. As expected, fear responses are highest when CS is presented in CTX+, smallest when CS is presented in CTX-, and intermediate in a novel CTX. These results align with measured fear responses in the empirical literature [245, 103].

Fig. 2.7 shows the mean CS-evoked responses of *BLA* neurons. To generate this plot,

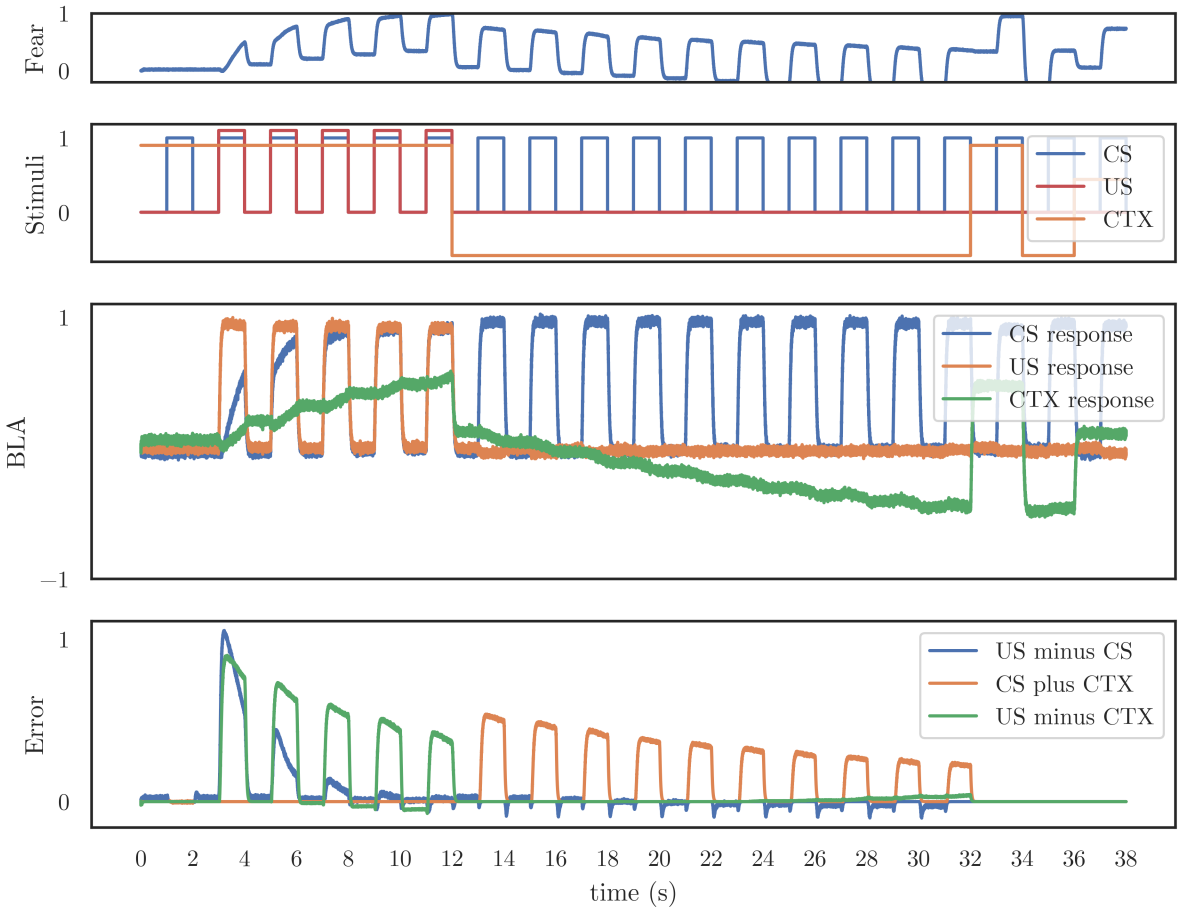

Figure 2.5: Fear learning in the complete functional model. During acquisition, *BLA* learns to associate both CS and CTX+ with US, leading to stronger fear responses. The contextual switch and extinction trials suppress this response in CTX-. Fear expression is then tested in three contexts: CTX+, CTX-, and a novel CTX that is partially similar to both CTX+ and CTX-.

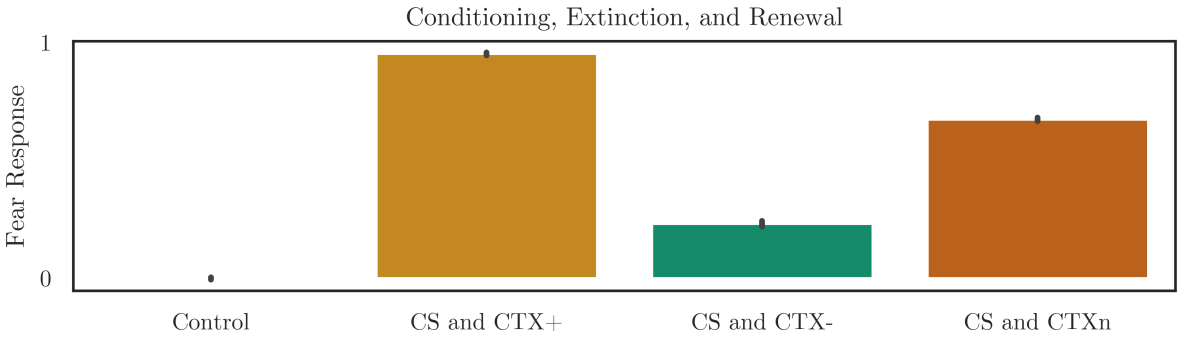

Figure 2.6: Fear responses across 100 unique instances of model 1. For each model instance, we initialized the model with a random network seed and three random learning rates for the three PES learning rules. We trained the models using an ABX experimental protocol, and measured their fear responses in CTX+, CTX-, and a novel CTX.

we collected spike data from all neurons in *BLA*, then categorized them based on their mean activity. We classified a neuron as a "fear neuron" if its CS-evoked mean activity (a) increased by at least 50% between baseline and the CTX+ test, and (b) decreased by at least 50% between the CTX+ and CTX- tests. We classified a neuron as an "extinction neuron" if its CS-evoked mean activity increased by at least 50% between the CTX+ and CTX- tests. We classified a neuron as a "persistent neuron" if its mean activity (a) increased by at least 50% between baseline and the CTX+ test, and (b) changed by less than 50% between the CTX+ and CTX- tests. Fig. 2.7 shows that model 1 contains neurons of all three classes, which have been repeatedly identified in empirical studies [161, 131].

Lastly, Fig. 2.8 shows how many neurons in *BLA* were classified according to the above scheme. We recorded the number of neurons in each class for an individual network, divided by the total number of neurons in *BLA* to calculate a normalized value, then plotted the distribution of this value as a histogram across the 100 unique networks. Estimates of these numbers vary widely across in empirical experiments and measurement criteria [90], ranging from 5% at the low end [135] to 40% at the high end [193]; estimates in the range of 20%-30% are quite common [27, 193]. In our model 1 experiments, we observed that 5%-20% of simulated neurons could be reliably classified as fear neurons or extinction neurons: these values are within the ranges reported in the literature.

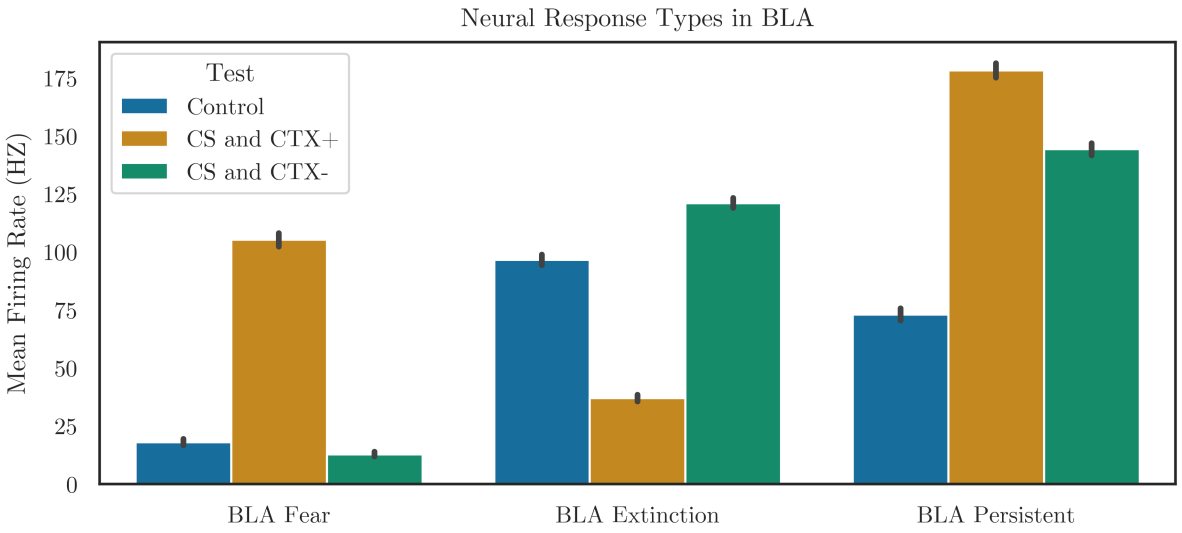

Figure 2.7: Mean firing rates of *BLA* neurons, classified by response profiles, across 100 unique instances of model 1. Note that the differences in mean firing rate between different experiments (for each neuron type) are largely determined by an arbitrary measurement threshold in the data analysis: a neuron had to change its firing rate by at least 50% to be counted towards inclusion in one of these categories.

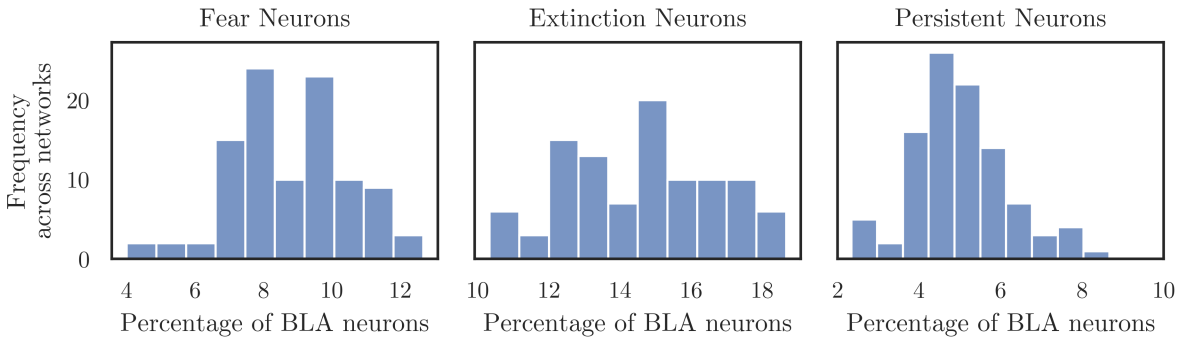

Figure 2.8: Histograms of the percentage of fear neurons, extinction neurons, and persistent neurons observed across 100 unique instances of model 1. After performing the neural response categorization for Fig. 2.7, we recorded the number of neurons in each category, divided by the total number of neurons in *BLA*, and recorded the response. This plot shows how consistent these percentages are across unique model instances.

2.2.2 Anatomically Detailed Model

Model 1 captured many of the important results in traditional fear learning experiments: fear responses changed over the course of acquisition and extinction; fear expression varied based on the experimental protocol used and the test context; and simulated neurons could naturally be classified into fear neurons, extinction neurons, and persistent neurons. However, model 1 lacks many anatomical details that define the biological AMY. In particular, model 1 contained one central population that handled all the incoming connections and computed all the important quantities. As we discussed in Sec. 2.1.2, the AMY is anatomically divided into several nuclei that receive specific external connections, perform specific functional roles, and have limited internal and external projections. Furthermore, model 1 does not make any novel experimental predictions: rather, it serves as a demonstration that our signals, learning rules, and representations are working as intended (i.e., they reproduce well-established experimental results) in a simplified network

Model 2 aims to recreate the essential anatomical divisions in AMY while preserving the core functionality of model 1. The purpose of this extension is twofold. First, a successful implementation of model 2 would show that our functional description of the AMY is not tied to an oversimplified model of its structure, and would increase the biological realism of our theory. Second, including more biological detail would open new avenues for measurement and experimentation, increasing the predictive power and applicability of our model. As with model 1, we will introduce the components of model 2 in a piecewise manner to clarify how they act in isolation, then perform a complete analysis on the full model and make several experimentally testable predictions. The complete model is shown in Fig. 2.9.

We begin with a model of CS-US association in the LA. LA contains two neural populations: LA_{pyr} are the excitatory pyramidal neurons, and LA_{inh} are the inhibitory interneurons. Both of these populations are two-dimensional and represent both CS and US information, which are again delivered through external populations $external_{\text{CS}}$ and $external_{\text{US}}$. LA_{pyr} receives both CS and US signals normally. LA_{inh} receives the high-dimensional CS signal using a fixed connection that detects the presents of any CS: when a CS is present, LA_{inh}'s CS response will be 1, and when a CS is absent (during every 1s pause between stimuli), LA_{inh}'s CS response will be zero. Finally, LA_{pyr} transmits the US signal to LA_{inh}. As in model 1, the connection between $external_{\text{CS}}$ and LA_{pyr} is learned using the PES learning rule, leading to a fear acquisition. Error is computed in a population LA_{error} and is equal to the difference between LA_{pyr}'s CS response and the presence of US; unlike in model 1, this US information is conveyed via LA_{inh}. Fig. 2.10 shows the dynamics of fear acquisition in LA. As expected, learning causes LA_{pyr} to become responsive to CS,

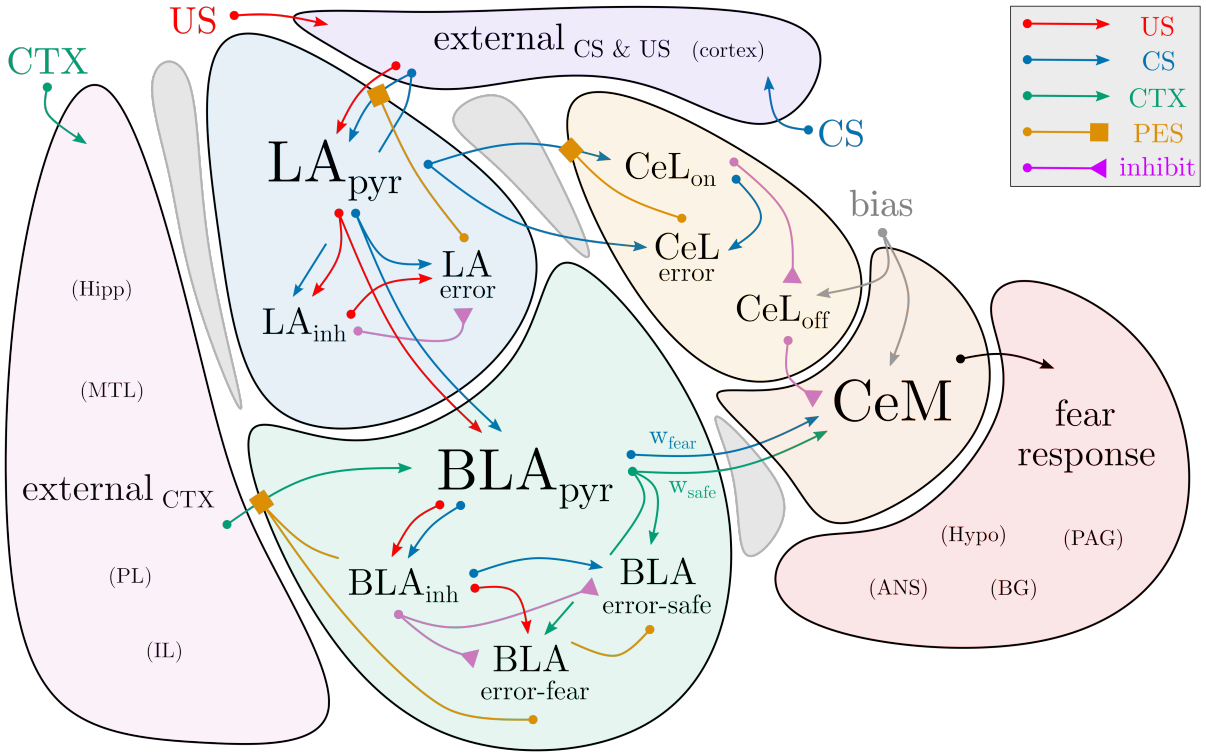

Figure 2.9: Anatomically detailed model of fear conditioning in the amygdala. Black text indicates neural populations. Colored arrows indicate the communication of information (red for US, blue for CS, green for CTX) or the application of a particular function (orange for PES learning, pink for direct inhibition). See text for details about the representations used in each population and the functions computed by each connection.

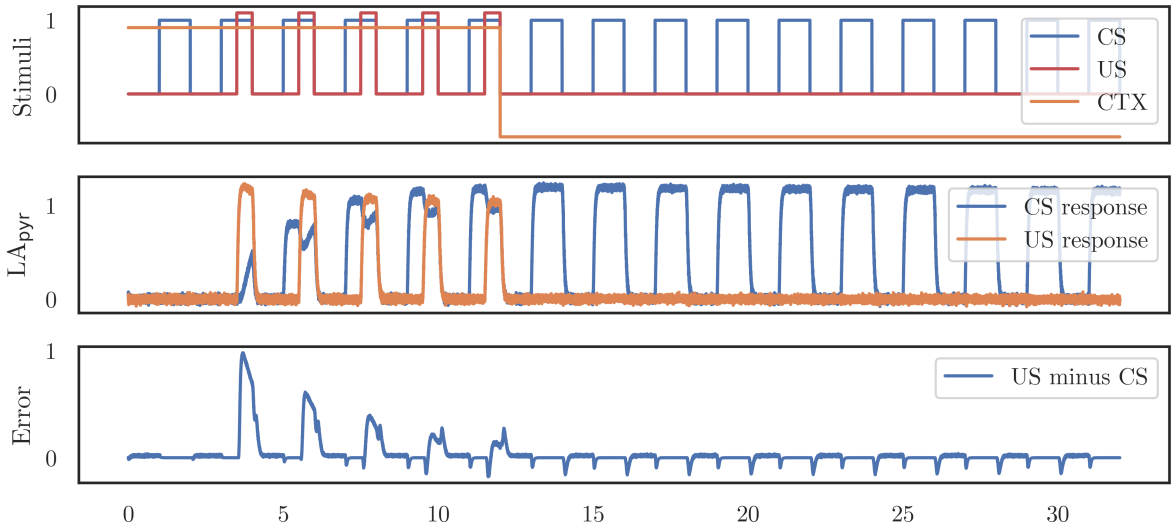

Figure 2.10: Dynamics of fear acquisition in model 2; compare to Fig. 2.3 from model 1.

leading to a heightened fear response that persists through the extinction phase.

Next, we add several populations for the neurons found in the central amygdala, CeL and CeM. Broadly speaking, CeL learns CS-US associations during fear acquisition, and transmits fear responses from LA to CeM; while CeM is the output nucleus, combining fear and safety signals from all other populations into a final fear response. In model 2, LA_{pyr} connects to a population called CeL_{on}, which inhibits another population called CeL_{off}, which in turn inhibits the final population CeM. During fear acquisition, CeL_{on} learns to mirror the CS-responsiveness of LA_{pyr}: the error population CeL_{error} calculates the difference between CeL_{on}'s response and LA_{pyr}'s CS response, which drives PES learning in the usual manner. Following training, CeL_{on} activity inhibits CeL_{off}, which disinhibits CeM and creates a fear response. Fig. 2.11 shows an example simulation of this network: CeL_{on} responses closely mirror LA_{pyr}'s CS responses after fear acquisition, and this correspondence is preserved during extinction.

The final components of model 2 are four BLA populations: BLA_{pyr}, BLA_{inh}, and two error populations. BLA_{pyr} represents learned responses to CS, US, and CTX; it receives CS and US information from LA_{pyr} and CTX information from $external_{\text{CTX}}$. The former connection is fixed (a simple communication channel), and the latter is updated with PES learning driven by $BLA_{\text{error-safe}}$ and $BLA_{\text{error-fear}}$, which govern extinction learning and context-dependent fear conditioning, respectively. BLA_{inh} gates context learning

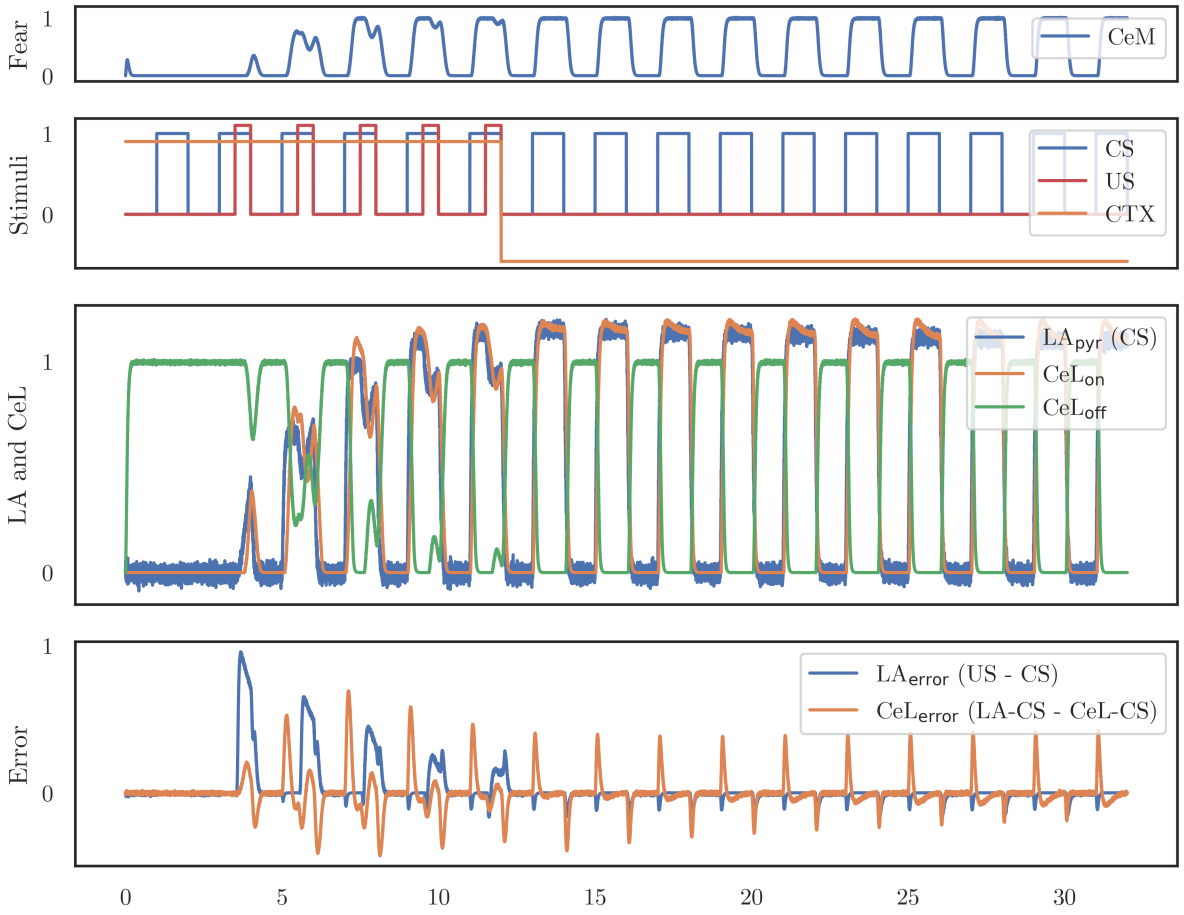

Figure 2.11: Fear conditioning produces CS-induced responses in CeL_{on} neurons, which in turn inhibits CeL_{off} neurons and disinhibits CeM, causing a fear response.

by communicating relevant signals and by inhibiting the error populations in appropriate conditions: it receives CS and US information from BLA_{pyr} and transmits these signals to the learning rules; it lightly inhibits $BLA_{\text{error-fear}}$ unless the US is present, leading to fast contextual fear acquisition but slow de-acquisition; and it inhibits $BLA_{\text{error-safe}}$ unless CS is present and no US has occurred in recent history, leading to fear extinction only during repeated presentations of CS without US. Finally, BLA_{pyr} connects to CeM, communicating both contextual fear and safety signals; the relative weights of these signals, w_{fear} and w_{safe}, are free parameters than govern the strength of of contextual fear suppression.

Fig. 2.12 shows the dynamics of neural representations in the complete model during acquisition and extinction training. BLA_{pyr}'s CS response increases during acquisition as LA_{pyr} neurons undergo synaptic plasticity. At the same time, whenever a CS and US are simultaneously present, BLA_{pyr} acquires a fear association between CTX and US. In CTX+, this CTX-induced fear response combines with the CS-induced fear responses in BLA_{pyr} (and CeL_{on}) to produce maximal levels of freezing. In between CS presentations during acquisition, model 2 still exhibits moderate fear expression, due to continued fear activation driven by BLA_{pyr}'s CTX+ response. After switching to CTX-, BLA_{pyr}'s CTX response drops back to (approximately) zero: the new context shares few enough features with CTX+ that it does not activate the same $external_{\text{CTX}}$ neurons, meaning the learned CTX+ response will not be engaged. What's more, during extinction training, newly activated $external_{\text{CTX}}$ neurons (those sensitive to CTX- but not CTX+) will provide the basis for learning a new association between CTX-, CS, and the absence of US. This association drives BLA_{pyr}'s CTX response to negative values; when this is added to the CS-induced fear response in CeM, the result is suppressed fear expression.

2.3 Results

To summarize, model 2 contains four major anatomical divisions (LA, BLA, CeL, and CeM), each of which contains subpopulations of excitatory pyramidal and/or inhibitory interneurons. The internal and external connectivity between these subpopulations is constrained by the known neuroanatomy of AMY. Sensory information (CS and US) are conveyed through LA, which projects to BLA and CeL, both of which project to CeM in turn; while context information (CTX) is conveyed to BLA exclusively. Internal connectivity within LA, BLA, and CeL computes the errors between the current responses of excitatory neurons and the appropriate responses, given the presence or absence of the US. Error signals are gated by inhibitory subpopulations within each nucleus, which are laterally connected to the pyramidal neurons. Learning within the network occurs online,

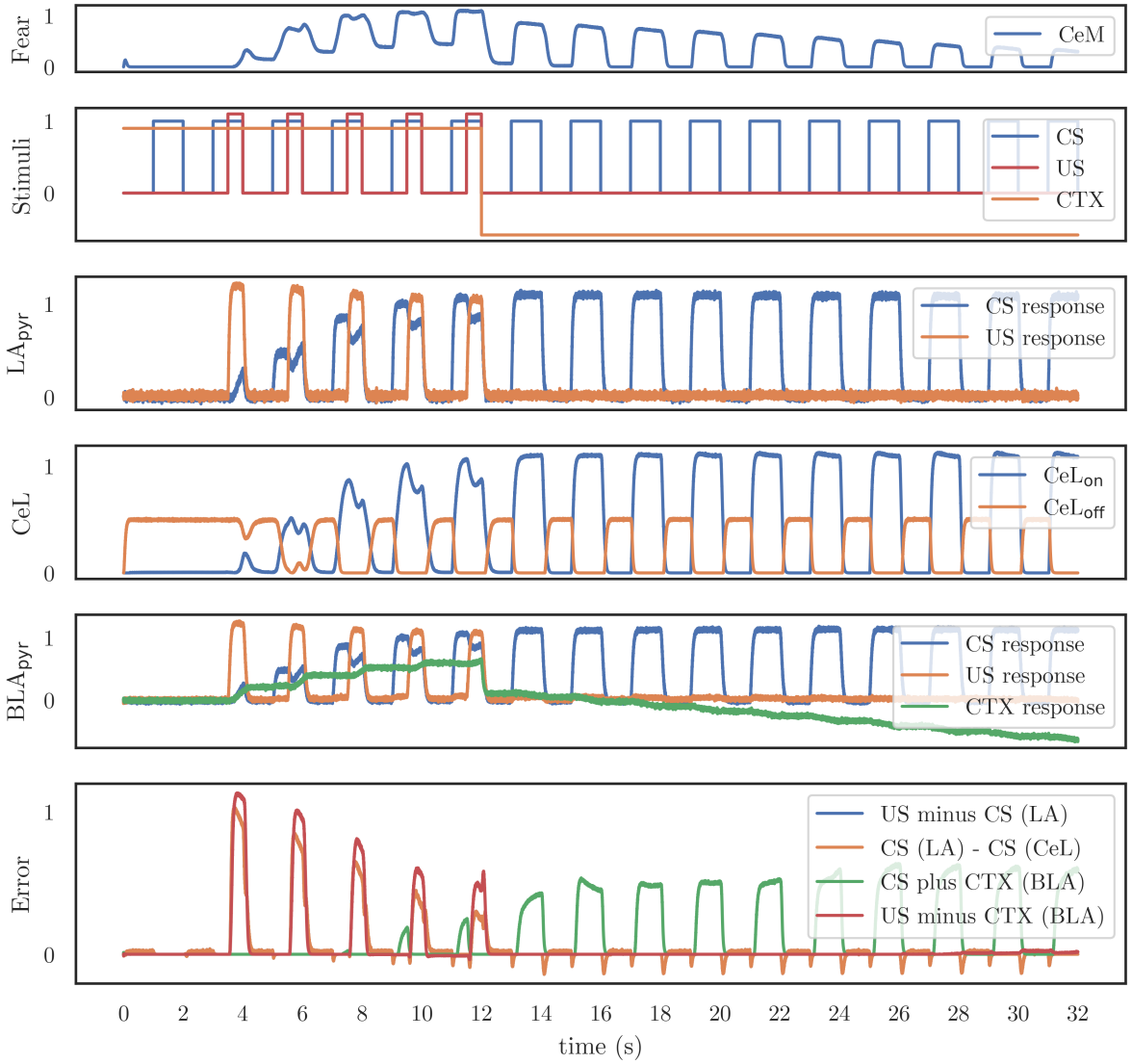

Figure 2.12: Fear conditioning and extinction in the complete version of model 2. CS-induced fear responses learned in LA_{pyr} and CeL_{on} drive fear expression in CeM, but CTX-induced safety responses learned in BLA_{pyr} during extinction training partially suppress these responses during testing.

as CS and/or US are presented to the network while in some CTX. Below, we run a series of experiments to assess the electrophysiological and behavioral realism of model 2.

2.3.1 Fear Expression

We begin with the standard ABX fear conditioning protocol. We randomly initialized 100 unique instances of model 2 with random network seeds, learning rates, w_{fear}, and w_{safe}. We trained these networks in CTX+ with five CS-US pairings, then switched to CTX- and presented ten unpaired CSs. During testing, we externally inhibited all error populations to prevent learning, then presented the CS once in either CTX+, CTX-, or a novel CTX. We recorded the decoded value from *CeM* as a proxy for model 2's overall fear response; we report the mean value of each network averaged over time, then plot the variance in this value across networks. Fig. 2.13 (right panel) shows the results: the fear response is zero before training, and is greatest when the model is presented with the CS in CTX$^+$. In CTX$^-$, the fear response is almost entirely suppressed, but in a novel CTX*, we observe a renewed fear response. These results align with behavioral data from animal studies [245, 103], and are consistent with human data [104] in other fear conditioning paradigms (Fig. 2.13, left panel).

2.3.2 Neural Responses

We collected spike data from each neuron in model 2 during the tests reported in the experiment above. As in model 1 and Fig. 2.7, we classified neurons based on differences between their mean firing rates in during each fear expression test. LA_{pyr} neurons were classified as "up" neurons if their mean activity in the CTX+ test was at least 50% larger than their mean baseline activity, and as "down" if their mean CTX+ test activity was at least 50% smaller. Similarly, BLA_{pyr} neurons were classified (a) as "fear" neurons if their CTX+ activity was at least 50% larger than baseline and if their CTX- activity was at least 50% smaller than CTX+ activity (i.e., the neurons became responsive, then were suppressed); (b) as "extinction" neurons if CTX- activity was at least 50% larger than CTX+ activity (i.e., the neurons only became responsive to the safe context); and (c) "persistent" neurons if CTX+ activity was at least 50% larger than baseline, but then did not change by more than 50% in CTX- (i.e., fear neurons, without the suppression). We also measured the mean activities of CeL_{on} and Cel_{off} neurons, which were defined as separate populations within the network (rather than classified based on mean activities after the simulation). Fig. 2.14 reports the mean activities of these categorized neurons,

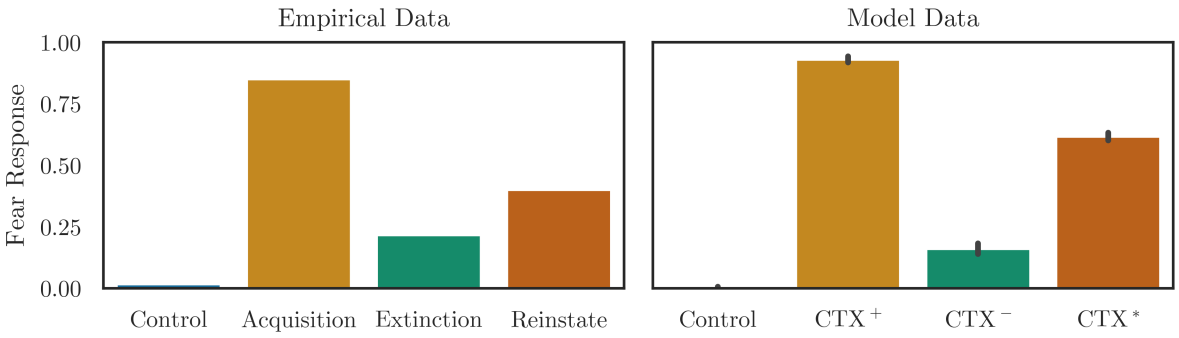

Figure 2.13: **Left**: Empirical data on fear expression. In [104], the authors paired CS^+ with US and CS^- with safety during acquisition, then removed the CS^+-US association during extinction. During a reinstatement test, they administered a few unpaired USs, then measured fear responses to unpaired CS^+. Reported values indicate the differences between participant CS^+ and CS^- fear responses, as measured by self-reporting (or US expectancy, not shown). While this procedure is different than the ABX protocol that we simulate, the measured fear expression following acquisition, extinction, and testing follows a similar pattern. Data reproduced from [104] ($n = 14$); fear response values were renormalized to fall between 0 and 1. **Right**: Fear responses across 100 unique model 2 instances. Networks were subjected to acquisition and extinction training then tested in three different contexts: the acquisition context CTX^+, the extinction context CTX^-, or a novel context CTX^*. Consistent with empirical data, fear responses are negligable before conditioning (control) and greatest in CTX^+; fear responses in CTX^- are greatly reduced due to contextual extinction, but reemerge in novel CTX^*. Error bars plot 95% bootstrapped confidence intervals across model instances.

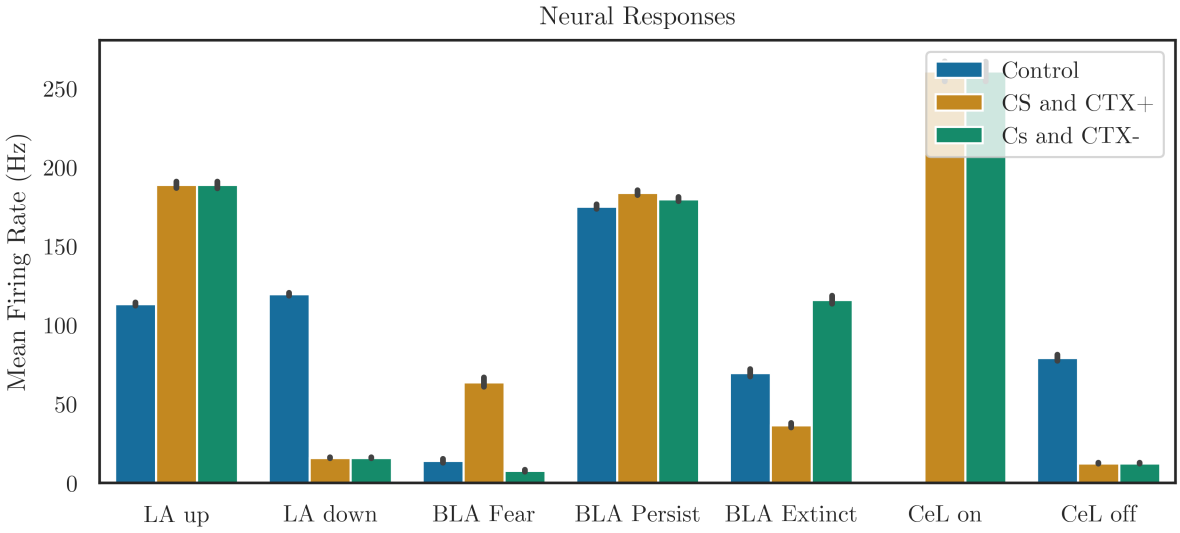

Figure 2.14: Mean firing rates of neurons with characteristic response profiles. See text for details on neuron classification. Population firing rates were measured by convolving simulated spike trains with a lowpass filter (τ=30ms) and averaging over time and neurons. Coloration indicates when, and in what context, these measurements were taken: blue bars show the baseline mean firing rate before conditioning; while orange and green bars show the mean firing rate following acquisition and extinction training, when the network is presented with the CS in either the acquisition context (orange) or the extinction context (green). For each population, differences between colored bars demonstrate characteristic response profiles: for example, BLA "fear" neurons are initially quiet, activate in response to the CS presented in CTX^+, but are inhibited when the CS is presented in CTX^-.

while Fig. 2.15 depicts how many of these neurons were identified (reported as a percent of the total number of neurons in the respective population).

2.3.3 External Activation or Inactivation

One advantage of simulating an anatomically detailed neural model is the ability to plausibly recreate animal experiments. Many researchers who study fear conditioning use pharmacology (drug application), lesioning (physically damaging a volume of brain tissue), or direct brain stimulation (through magnets or electrodes) to externally manipulate neural activities within AMY. These experiments have provided important insights into AMY's

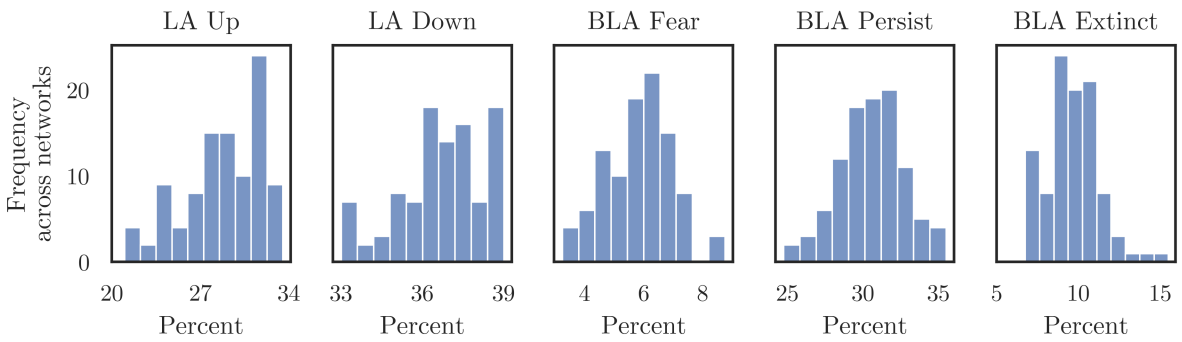

Figure 2.15: Histograms of the number of neurons from each classification across 100 unique instances of model 2. Learning induces CS-responses within each model, but the number of affected neurons depends on the network seed and learning rates. The observed number of fear, extinction, and persistent neurons in model 2 agrees with empirical estimates, which typically range from 20%-30% [27, 193]

abstract functional neuroanatomy, and have been critical in the development in theoretical and computational models of fear conditioning. In this section, we attempt to recreate some of these experiments by directly injecting current into our model neurons and observing the effects on acquisition, extinction, or expression. In each experiment, a constant current is delivered to all neurons within one nuclei (LA_{pyr}, LA_{inh}, BLA_{pyr}, BLA_{inh}, CeL, or CeM) during one phase of the standard ABX experiment (control, during acquisition, during extinction, during the CTX+ test, or during the CTX- test).

Fig. 2.16 reports the results of the inhibition experiments, and Fig. 2.17 reports the results of the activation experiments. In the majority of cases, externally activating or inhibiting a population has no effect: for instance, perturbing CeM during training has no effect on fear responses, and perturbing interneurons in LA or BLA during testing has no effect on fear expression. For experiments where external inhibition or activation had a significant effect of fear responses, we have indicated whether these results align with the empirical data: green circles indicate agreement, red circles indicate disagreement, and blue circles are unclear. We discuss the specifics below.

Inhibiting pyramidal neurons in LA prevents the learning of CS-US associations during acquisition and reduces fear expression during testing [173]. We observed both these effects in our model (row 2, columns 2,4,5). We also observed that inhibiting LA_{pyr} during extinction impaired contextual learning (row 2, column 3): although such learning occurs entirely in BLA, our model predicts that healthy LA activity is required to relay stimulus

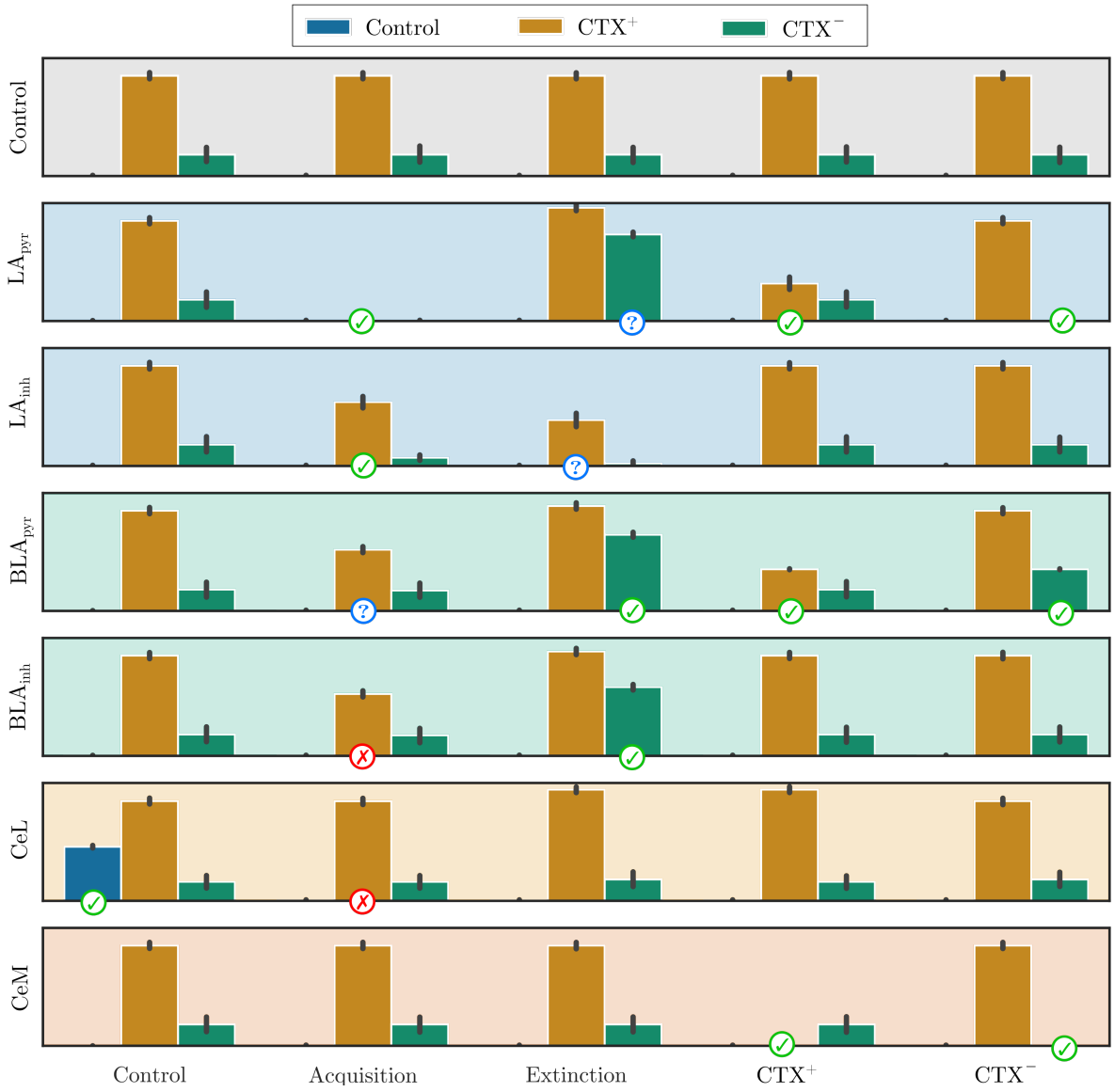

Figure 2.16: Fear responses when one AMY nucleus is externally inhibited during one phase of the ABX experiment. Data is collected from 10 unique model 2 networks, and mean fear response is plotted on the y-axis. Rows indicate which AMY nucleus was inhibited, columns indicate when the inhibition was applied, and bar coloration indicates when the fear response was measured. Circled marks indicate whether the result agrees with empirical evidence (green), disagrees (red), or is ambiguous or untested (blue). Unmarked data are not significantly different from the control condition.

information to BLA.

Inhibiting interneurons in LA prevents the learning of CS-US associations. Empirically, the precise timing of inhibitory spikes is essential for LTP in pyramidal neurons, and complex networks of inhibition and disinhibition may facilitate pattern separation in LA and BLA [131]. Although PES is not an spike-timing dependent plasticity rule and does not explicitly perform pattern separation (but see [22, 19]), we found that inhibiting $\mathrm{LA_{inh}}$ impaired fear acquisition but preserved fear expression (row 3, columns 2,4,5). We also found that inhibiting $\mathrm{LA_{inh}}$ during extinction led to reduced fear expression (row 3, column 3).

Inhibiting pyramidal neurons in BLA prevents extinction and suppresses fear responses [213]. We observed these effects in our model (row 4, columns 3,4,5). We also found that inhibiting $\mathrm{BLA_{pyr}}$ during acquisition impaired fear responses (row 4, column 2), presumably because contextual fear associations were not properly learned. We predict that, in experiments where contextual cues associate with the US, inhibiting pyramidal neurons in BLA will impair acquisition.

Inhibiting interneurons in BLA has varied effects on fear conditioning. Optogenetically inactivating BLA neurons during acquisition appears to facilitate fear learning [131]. This result disagrees with our simulations, in which inactivating $\mathrm{BLA_{inh}}$ impaired fear conditioning (row 5, column 2). One explanation for this discrepancy is that [131] targeted only a single interneuron cell type (PV neurons); inhibiting other interneuron subtypes (such as SOM and CCK) might impair fear acquisition, as we observed. In contrast, disrupting BLA interneurons during extinction, or ablating axoaxonic inhibitory synapses in BLA [207], appears to impair contextual safety learning [131]; our results support this conclusion (row 5, column 3).

Inhibiting neurons in CeL removes its default inhibition of CeM, causing unconditioned fear responses [40]. We also observed this effect (row 6, column 1). Inhibiting CeL during acquisition has also been shown to impair fear learning, but in our simulations, inactivating CeL during acquisition had no effect (row 6, column 2). This is surprising, given that external inhibition should prevent learning in CeL during acquisition. It seems that the connection between $\mathrm{LA_{pyr}}$ and $\mathrm{CeL_{on}}$ is instead updated during other phases of the experiment. This can occur because CeL_{error} tries to make $\mathrm{CeL_{on}}$ behave like $\mathrm{LA_{pyr}}$, independent of the US, allowing learning during the extinction phase. This is an oversight in our model that should be addressed in future work.

Finally, inhibiting CeM impairs fear expression, both empirically [40] and in our simulation (row 7, columns 4,5).

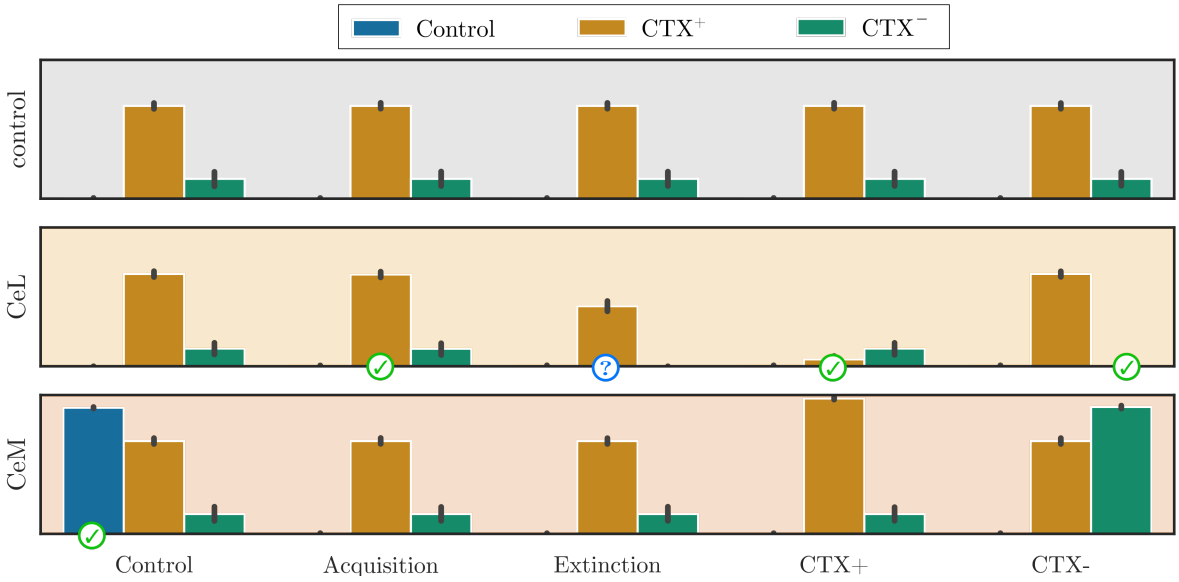

Figure 2.17: Fear responses when external activation is applied to model 2. See Fig. 2.16 for details.

We also experimented with externally activating CeA by directly injecting positive current. In [40], the authors showed that applying oxytocin, an excitatory neurotransmitter, to CeL during acquisition preserves fear learning, but applying it during testing impairs fear expression. Similarly, in our simulations, we found that exciting CeL impaired expression but did not affect acquisition (row 2, columns 2,4,5). We also found that activating CeM produces a spontaneous fear response (row 3, column 1), another result consistent with empirical data [40].

2.3.4 Fear Generalization

To investigate fear generalization, we trained our model by presenting ten pairs of CS^+-US and ten unpaired CS^- (order randomized), turned off learning, and presented a series of novel stimuli (CS^*). Recall that we represent CS inputs using high-dimensional vectors that encode complex sensory information; to generate CS^- and CS^*, we simply created more vectors using our original sampling procedure. The similarity between these vectors and CS^+ was calculated using cosine similarity. Fig. 2.18 (right panel) shows how the model's fear responses to CS^* changes as a function of this similarity. As expected, fear responses decrease as CS^* becomes more dissimilar to CS^+.

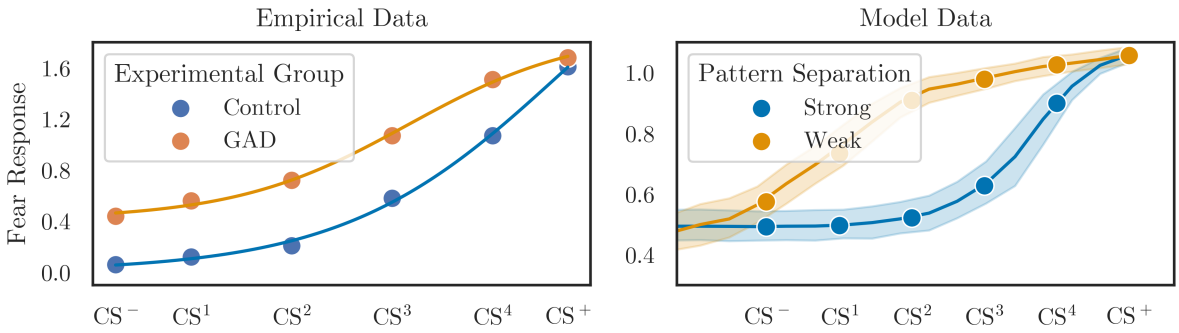

Figure 2.18: **Left**: Empirical data on fear generalization. In [145], the authors paired the presentation of a large circle (CS^+) with a mild electric shock, and paired the presentation of a small circle (CS^-) with no shock. They then presented circles of intermediary sizes (CS^1-CS^4) and measured participant fear responses using a self-reported risk rating (or startle EMG, not shown). They performed this experiment in both healthy controls ($n = 26$) and patients with generalized anxiety disorder (GAD, $n = 22$). They found that generalization gradients were more gradual in GAD patients. Data reproduced from [145], and sigmoids were fit to the data using Scipy's *curve_fit* function [247]. **Right**: Fear generalization as a function of similarity between a novel CS^* and the CS^+. To manipulate the degree of pattern separation in our model, we adjusted the tuning curves in EXT_{cs} to make neurons more (or less) sensitive to a broad range of input stimuli. We found that models with stronger pattern separation exhibited sharper fear generalization gradients, while networks with weaker pattern separation exhibited more gradual gradients. This trend is consistent with empirically-observed differences between healthy humans and individuals with anxiety disorders. Shaded regions indicate 95% bootstrapped confidence intervals across 10 unique model instances.

Several empirical experiments have characterized how fear responses to CS* decreases with similarity to CS$^+$. Fig. 2.18 (left panel) reproduces the data from an influential experiment, which studied fear generalization in both healthy control subjects and participants with generalized anxiety disorder (GAD) [145]. The authors found that the fear responses of control subjects declined sharply as similarity to CS$^+$ decreased, but that the gradient for GAD subjects was more gradual. These results are consistent with the hypothesis that GAD patients overgeneralize fear responses to novel stimuli, prompting a fear response when CS* shares even a few features with the CS$^+$ [56]. Similar trends have been reported in individuals with panic disorders [146] and post-traumatic stress disorders [147].

One possible explanation for overgeneralized fear responses is that neural tuning curves in AMY, sensory cortex, and/or hippocampus are insufficiently selective for the CS$^+$: they continue firing even when presented with dissimilar stimuli [145, 60]. To test the idea that poor pattern separation smooths the gradient of fear generalization, we modified the tuning properties of our neurons and repeated the above experiment. In our model, external stimuli are first represented in EXT_{cs} and EXT_{ctx}, which correspond to sensory cortex and hippocampus. When initializing these populations, we use encoders that evenly tile the input space, ensuring that all possible stimuli engage a subset of neurons. We also specify the sparsity of neural representation in EXT_{cs} by setting α_i and β_i, which effectively determines how similar a CS input must be to the encoder $\mathbf{e}_i$ before the neuron begins firing. We hypothesized that, by varying the sparsity of neurons in EXT_{cs}, we could control the degree of pattern separation in our model and reproduce the differences between healthy controls and anxious individuals.

Fig. 2.18 shows generalization gradients for two sparsity values. For each sparsity, we trained ten unique model instances, and tested each with a set of 100 CS*s. We observed that models with sparser neural representations (stronger pattern separation) displayed sharper generalization gradients. In sparse networks, fear responses declined rapidly as soon as CS* became distinguishable from CS$^+$. In contrast, in networks with weak pattern separation, fear responses remained high for CS*s that were similar to CS$^+$, and only declined as CS* began to resemble CS$^-$. These trends are also present in the empirical data. In healthy controls, fear responses declined rapidly between CS$^+$ and CS^2 then plateaued for CS^1 and CS$^-$. In contrast, in GAD participants, fear responses remained high between CS$^+$ and CS^4, then declined gradually until CS$^-$. Similar gradients are apparent in patients with panic disorders [146] and PTSD [147]. Both the empirical and simulated gradients seem well-characterized by sigmoid curves, shifted up-down or left-right by the degree of pattern separation (or patient pathology).

To confirm this characterization, and to demonstrate the scalability of our model to more complex stimuli, we repeated this generalization experiment for multiple values of the

CS dimensionality (D) and network sparsity. In order to visualize the entire generalization gradient, we trained the model with a CS^- that was maximally dissimilar to CS^+, then tested 100 CS^*s with intermediate similarities. Fig. 2.19 shows that generalization gradients appear sigmoidal regardless of stimulus complexity, and that weakening pattern separation in the model consistently shifts this sigmoidal gradient to the left. Unfortunately, the empirical data on fear generalization are too sparse to validate our findings: the current data all use the same experimental paradigm, do not report error bars in the plots, and use only simple stimuli. We predict that future experiments on fear generalization will observe the following trends: (1) the relationship between fear response and stimulus similarity will be well-characterized by a sigmoid curve; (2) the generalization curves of anxious individuals will be shifted left (with centers towards more dissimilar stimuli) relative to healthy individuals; and (3) experiments that use simpler stimuli (those that include few discernible features) will produce more extreme differences between healthy and anxious individuals (larger shifts to the left).

2.4 Discussion

Now that we have introduced our model, demonstrated its capacity for fear conditioning, and run several experiments, we discuss how our model relates to other empirical and computational studies of AMY. We begin by evaluating the biological realism of our model, then summarize its functional capabilities. We then compare our model to other computational models of AMY, thinking in particular about neurons, representations, learning rules, and empirical validation; these comparisons are summarized in Table 2.1. We conclude by proposing several directions for future research, and integrating our model into larger cognitive systems.

2.4.1 Biological Realism

One of the major goals of this chapter was to build a biologically detailed model of AMY with enough functional capacities to simulate associative learning and fear conditioning. To facilitate this, we first built a simpler model that was capable of the desired functionality: conditioning, extinction, and renewal of fear associations. By beginning with this model, we gained insights into which components (in particular, which representations and learning rules) were necessary, without the added complications of anatomical constraints. We then deployed these insights into a more complicated model, which captured many biological features of AMY. We discuss the realistic and unrealistic aspects of our model below.

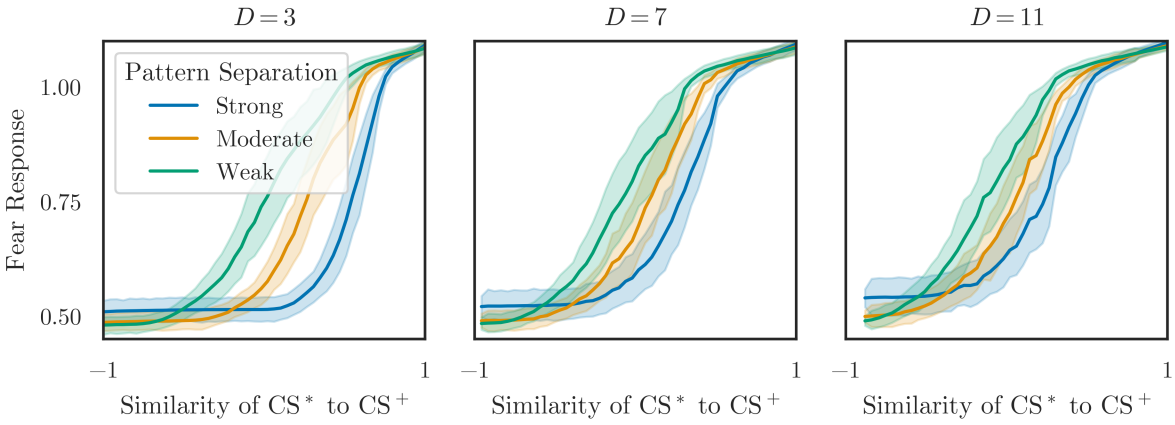

Figure 2.19: Simulated fear generalization as a function of stimulus complexity and network sparsity. As the dimensionality D of the input stimulus increases, fear generalization curves maintain their sigmoidal shape, regardless of the degree of pattern separation in the model. However, differences between the observed generalization gradients in networks with strong versus weak pattern separation are most obvious for low-complexity stimuli ($D = 3$), and are less apparent for high-complexity stimuli ($D = 11$). In conjunction with the results from Fig. 2.18, these findings allow us to make three predictions: fear generalization curves are generally sigmoidal; weaker pattern separation shifts generalization curves to the left, such that greater fear responses are observed for more dissimilar CS*s; and experiments that test simpler stimuli better reveal the differences between individuals with strong and weak pattern separation.

The main biological feature of our model was the simulation of multiple AMY nuclei with constrained connectivity: we simulated the lateral amygdala (LA), basolateral amygdala (BLA), central lateral amygdala (CeL), and central lateral amygdala (CeM). While other subdivisions of AMY are possible (for instance, further subdivision of LA and BLA by cell type or response properties [161]), these four populations are the most frequently mentioned in theoretical accounts of fear conditioning. We also simulated populations of neurons that correspond to GABAergic interneurons in LA and BLA. The connectivity between AMY nuclei was anatomically constrained, with the result that the flow of information within AMY was highly structured. This posed an interesting modelling challenge, but ultimately led to well-defined functional roles for each nucleus.

The learning rules in our model produce neurons whose activities change in response to the CS following conditioning and/or extinction, including: CS-up and CS-down neurons in LA; fear neurons, extinction neurons, and persistent neurons in BLA; and CS-up and CS-down neurons in CeL. Only the latter two were directly implied by the structural divisions in our model. In Sec. 2.3.2, we showed the differences between mean firing rates of these neurons as a function of test context, and provided estimates of the relative frequency of these neurons within their respective populations. While these figures seem broadly consistent with measurements made in a few empirical experiments [105], direct comparison is difficult for two reasons. First, we found that the number of neurons classified as, for example, fear neurons versus extinction neurons, depended on several model parameters that were arbitrarily chosen. These include the dimensionality of the input signals, the encoder distribution of AMY neurons, and learning rates. These theoretical quantities are difficult to define or measure empirically. Second, while measurement of neural spike trains is easy in simulation, it is difficult in practice. AMY is located deep in the temporal lobe, making imaging with external tools like fMRI or EEG problematic. Neuroscientists have traditionally used more invasive techniques to measure neural activity in LA and BLA, supplemented by modern tools like optogenetics. Unfortunately, even with these tools, it is difficult to gather a large dataset of neurons whose activities change during fear conditioning, and to ensure that neural activity is properly measured during baseline, throughout training, and during testing. Better electrophysiological data are needed to properly validate the response properties of our simulated neurons.

The third instance of biological realism in our model comes from our "ablation studies" in Sec. 2.3.3, in which we externally activated or inhibited various nuclei, and observed the effects on fear conditioning, extinction, and expression. We found that, in almost all cases, these perturbations produced behavioral effects that aligned with empirical experiments, in which chemicals like muscimol, or neurotransmitters like oxytocin, were used to inhibit or excite neurons within specific nuclei. These results support our theoretical account of

how biology and anatomy relate to function.

Despite these successes, there are many aspects of our model that are clearly unrealistic, or do not align with the known neuroanatomy of AMY. We simulate leaky-integrate-and-fire (LIF) spiking neurons; while these neurons provide a better correspondence to biology than mean field approximations or rate-based rectified linear neurons, they are much less realistic than neuron models with detailed physiology, such as the Hodgkin-Huxley model. In particular, LIF neurons (as simulated here) do not reproduce many of the advanced electrophysiological properties observed in AMY neurons, including transient bursting (most LA neurons fire phasic bursts of spikes when CS is initially presented, but return to low tonic firing rates even before CS is removed [183]), habituation (neurons in LA and BLA become exhausted if US or CS is presented too many times), and expectation (neurons fire in anticipation of a stimuli arriving, rather than in response [169]). Furthermore, we do not model any physiological differences between pyramidal neurons and inhibitory interneurons: both are instances of the same LIF model class, and use synapses with identical time constants. A more realistic model would use different neuron models for pyramidal and inhibitory neurons, and would enforce Dale's principle when connecting these populations to one another, ideally using synapse models with sufficient complexity to account for neuromodulation. We explore many of these ideas in Ch. 5; future work should apply the insights gained there back to the AMY model presented here.

Many recent accounts have highlighted the importance of inhibitory interneurons for fear conditioning [153, 131]. Inhibitory neurons may (a) provide a "brake" that prevents runaway excitation in pyramidal neurons and controls the magnitude of learning, (b) be a site of synaptic plasticity that contributes to fear or extinction memories, (c) facilitate consolidation of extinction memories during experience replay, or (d) encourage pattern separation within AMY nuclei. In our model, inhibitory neurons merely gate learning based on co-occurance of the CS and US, inhibiting the error populations (and preventing synaptic plasticity) in order to prevent inappropriate learning (e.g., unlearning fear associations or learning contextual safety while a US is present). Clearly, our inhibitory neurons lack the rich functional capabilities that others have attributed to them. While we acknowledge the importance of these inhibitory circuits for normal AMY function, we were unable to devise a computationally- or functionally-specific theory of how they might realize these capabilities. While some other computational models include interneuron populations [140, 160], these models tend to focus on how anatomical connectivity produces empirically-observed patterns of neural activity, and place less emphasis on how this connectivity drives learning and behavior; as such, they do not provide functional justification for these neurons. We believe that specifying and modelling the functional role of inhibitory circuits is an important topic for future research.

Given this lack of functionality, it is not surprising that externally activating or inactivating inhibitory neurons in our model does not always have the expected effect. We believe there are three reasons for this. First, our interneurons do not control the default activity of pyramidal neurons, so inactivating them does not release inhibition and produce stronger learning [131]. While we could have implemented this lateral inhibition in our model, doing so would not have served any purpose: an inhibitory brake is only necessary if (a) excitatory neurons are recurrently connected and inhibition would prevent runaway excitation, or (b) external systems need to control the learning rate by pulling the brake. Second, we do not model inhibitory microcircuits in LA or BLA; interneurons of different types (those expressing various proteins and receptors, including parvalbumin, somatostain, and or cholecystokinin) appear to inhibit one another, and pyramidal neurons, in convoluted ways. Different interneuron types likely play specific roles in encouraging pattern separation and coordinating memory consolidation. Externally inhibiting different cell types consequently produces different effects; our model, which ignores these microcircuits, cannot capture these effects. Third, we do not model learning within inhibitory neurons; some synaptic plasticity between BLA pyramidal and interneurons has been observed empirically, but its mechanisms and functional roles are unclear. Future work that seeks to simultaneously realize fear conditioning, extinction, pattern separation, and long-term consolidation (in an anatomical circuit where excitatory and inhibitory neurons are laterally connected) will have to contend with the difficulty of managing many learning rules acting in parallel.

Our model also does not simulate recurrent connectivity between AMY and other brain structures, such as BG, sensory cortex, MTL, or hippocampus. Such connections have been featured in other models [160, 183, 114], whose focus is either higher-level (how AMY assigns salience in a larger cognitive circuit) or lower-level (reproducing electrophysiology). These connections appear to be important for contextual representation and memory consolidation, especially with regards to network oscillations in particular frequency bands [27, 248]. Recent NEF models have used hippocampal representations and network oscillations to perform navigation tasks guided by contextual cues [58, 57, 223]; we hope to integrate these mechanisms into future versions of the model, with the goal of connecting the complex contextual representations in MTL with the associative learning in AMY.

Finally, our learning rules are implemented online based on errors computed elsewhere in the network. While there is abundant evidence for such error signals throughout the brain [21] and within AMY specifically [142, 163, 171], many researchers believe that fear conditioning is driven (at least partly) by the precise timing of spikes incident on the dendrites [131, 27, 193]. While decoded US and CS signals must be concurrently present for learning to occur in our model, the precise timing of spikes representing these quantities

is not essential for learning. Future empirical work should continue to investigate whether spiking timing is a key component of associative learning in AMY.

2.4.2 Functional Capacity

The other major goal of our AMY model was to reproduce a variety of behavioral data from fear conditioning experiments. We represented external signals as high-dimensional vectors: one was labelled as the CS and relayed to LA via cortex; while the other was labelled as the CTX and related to BLA via MTL. In addition, a low-dimensional signal representing an aversive stimulus (the US) was periodically sent to LA. The task of the AMY network was to learn associations between the CS, CTX, and US, and to generate the fear response only when CS and CTX together predicted the onset of the US. Through the action of three error-driven learning rules acting in LA, BLA, and CeL, our model successfully learned these associations, exhibiting behavior that reproduced many effects observed in empirical animal experiments.

In our experiments, CS-US and CTX-US associations were presented during acquisition, but during testing only the CS or CTX was presented. As expected, our model produced the fear response during these tests. Next, we performed extinction training: we presented the CS without the US in a new CTX, and observed our model gradually suppress its fear response, as it learned that the new CTX- was associated with safety from the US. Though we did not report these results, we tested several extinction protocols: the classic ABX experiment, as well as "AAX", in which extinction occurs in the same context as acquisition and testing occurs in a new context X. We also investigated CTX-only acquisition and extinction using both these protocols: in these experiments, no CS was presented, and the model learned that CTX itself predicted danger or safety. In all these paradigms, we observed appropriate test behavior in our models: fear responses in the extinction context (with or without CS) were suppressed, fear responses in the acquisition context remained high, and fear responses in novel context were intermediate. These results align with the relative magnitude of fear responses in animal experiments that include acquisition, extinction, and renewal/test phases [245, 103].

We also showed that our model generalized its learned fear associations to similar stimuli and contexts. One major advantage of building the model in the NEF is that external information with rich semantic content can be represented and processed by the neural network. We arbitrarily chose our CS information to be three-dimensional, and our CTX information to be five-dimensional. We used normalized vectors within these spaces to represent various neural stimuli and contexts; we presented different combinations of these

vectors to the model during training and testing. After showing that the model produced a fear response to one particular vector, we investigated how this association generalized to similar vectors in the input space. We found that, when neural representations in our model were sufficiently sparse (i.e., realized strong pattern separation), that our generalization gradients agreed with empirical gradients that were observed in fear conditioning experiments [144, 59]: both simulated and empirical fear responses decline sigmoidally as similarity to CS+ decreases. This result is broadly consistent with generalization gradients across animal groups, behavioral contexts, sensory modalities, and learning styles [82]. Interestingly, we found that when we decreased the sparsity of neural representations (i.e., realized weak pattern separation), our generalization curves shifted left, extending the fear response to more dissimilar stimuli. These results resemble the generalization cures observed in patients with fear-related disorders, including generalized anxiety disorders, post-traumatic stress disorder, and panic disorders [60, 56]. Unfortunately, the empirical data on fear generalization is relatively sparse, being limited to a few research groups and task designs. Our computational model thus provides initial support for the theory that these disorders are mediated by weaker pattern separation in areas such as MTL, HIPP, and AMY, and provides quantitative predictions about the shape and magnitude of these curves that can be tested in future empirical studies.

While generalization would ideally emerge from any AMY model that learns fear associations between complex external stimuli, it is by no means assured, and (in fact) has rarely been demonstrated. Firstly, computational models that use simple representations cannot smoothly "compare" a novel CS to CS+. If a model represents CS inputs as a one-dimensional "on/off" signal, it cannot represent novel stimuli at all; and if a model represents CS inputs as "one-hot" vectors, every novel CS will be maximally dissimilar from CS+. Secondly, the simulated connection that learns fear associations may not support generalization: depending on the response curves of neurons in the model, and the mechanics of the learning rules, learning may not extend to novel stimuli not seen in the training dataset. In our model, the connection from the external populations representing CS/CTX, which are high-dimensional, to the LA/BLA, which are low-dimensional, necessarily requires dimensionality reduction. Before running our generalization experiments, we did not know whether this reduction would discard the information required for generalization. We were pleasantly surprised to discover that the NEF and PES learning rules, which account for the rich information encoded in spike trains originating from a diversely-tuned population of neurons, preserve the relevant information. We discuss this point further in Sec. 2.4.3

2.4.3 Comparison to Other Computational Models

The study of fear conditioning in the AMY has a long history of theoretical and computational work. Here, we discuss how these models compare to our model with respect to particular features, such as neural representation and empirical validation. These comparisons show that our model shares many features with existing models, but recombines and extends them in interesting ways.

Neural Detail

One of the key differences between brain models is how neurons are represented and simulated. In [35], the authors use the DANA library for neuronal representation to simulate acquisition, extinction, and renewal in a simplified AMY circuit; each neuron in the population implements a dynamical equation governed by the mean-field formalism, and outputs a real-valued firing rate. Each population (in LA, BA, or CeL) contains either one or ten neurons. Similarity, [158] simulates second-order conditioning in the interactions between LA, BLA, and CeA; each of these populations contain four neurons whose dynamics are governed by weighted input, a leak term, and a *tanh* nonlinearity. In [171], the authors simulate nodes in HIPP, vmPFC, BLA, ITC, and CeA with weighted inputs and logistic nonlinearities, and perturb connections weights with Gaussian noise to simulate neuronal variability. These models seek to explain fear conditioning at a high-level of abstraction, but bear little resemblance to the physical substrate of AMY. In contrast, [160] present an anatomically-detailed model that simulate a highly-complex circuit of firing rate neurons within LA, BA, PL, IL, CeL, ITC, and CeM. Each population in the model contains subpopulations of paravalbumin interneurons, somatostatin interneurons, and pyramidal neurons, whose membrane time constants and firing rates are calibrated against empirical data, and whose connectivity is tightly constrained by the known neuroanatomy. The authors show that synaptic changes induced by two forms of learning produce activity patterns that reproduce empirical response curves.

Several spiking neural networks have also simulated learning and salience detection in AMY. A model by [248] simulated both a mean-field and spiking version of a BLA network, which includes excitatory and inhibitory populations of spiking LIF neurons that compete through lateral inhibition. Similarly, [114] simulated both a firing-rate and spiking model of salience detection, attentional control, and planning that involves the AMY, thalamus, and cortex. Their AMY model includes LA, BA, and a small microcircuit of excitatory and inhibitory cells, and their spiking neurons were simulated using the Izhikevhich equations. Two recent models of AMY have also been developed using the NEF. In [75], the author

simulated an AMY circuit that contains LA, BLA and CeA along with a visual system, PFC, and BG; and in [258], the author developed a family of AMY models that simulate connections between LA, BA, CeA, and external systems for working memory and decision making. Both of these NEF models used LIF neurons, but do not model learning.

A third class of models simulate AMY neurons in greater biological detail. These models tend to place greater emphasis on recreating electrophysiological data than on reproducing behavioral results. In a model of AMY-PFC interaction, [183] used conductance-based Hodgkin-Huxley cells to simulate pyramidal neurons and interneurons in LA, BLA, CeA, PL, and IL; the authors showed that dopamine and norepinephrine release in BLA (through interaction with PL) helps transform transient LA responses into sustained fear responses. They reproduced CS-induced activity in PL, with and without pharmacological perturbation. In a model of fear acquisition and extinction in LA, [141] built a small microcircuit consisting of eight pyramidal cells and two interneurons: the cell and synapse models were highly realisitic, and the application of a Hebbian learning rule produced both conditioning and extinction, as evidenced by changes in neural activity. Finally, in a model of fear acquisition in LA, [126] built a network containing detailed pyramidal cells and interneurons; they showed that synaptic plasticy within LA (from connections originating in cortex and thalamus) was primarily responsible for fear learning, with synapses between LA neurons playing a minor role.

In our model, we simulate a total of fourteen neural populations across LA, BLA, CeL, CeM, Cortex, and HIPP. Each population contains 100-500 spiking LIF neurons that are sensitive to different combinations of the CS, CTX, and US. Neurons are connected by current-based exponential synapses, and synaptic weights are updated using an error-driven learning rule. We simulate the behavioral changes associated with fear conditioning, extinction, and generalization, measure the impact of externally inactivating neural populations, and observe changes in our neural activities that resemble the response profiles of neurons in-vivo. Our model therefore captures both the symbolic and functional properties of simpler models, and many of the anatomical and electrophysiological properties of the detailed models.

Representation

One feature that sets our model apart from most other models of AMY fear conditioning are the complex representations that we support. In many models, inputs to the network are one-dimensional signals delivered as external current to model neurons [141, 126, 183, 160, 248]. These inputs obviously cannot contain much information: they simply relay whether a CS/US is present or absent. Of these models, only [248] attempts to model

contextual inputs; they do so by sending a one-dimensional on/off signal for CTX-A into one population, and another on/off signal for CTX-B into another population.

Other models use "one-hot" vectors to represent sensory inputs. The model by [158] contains four input neurons, which code for the presence or absence of the four one-dimensional inputs (light, tone, food seen, and food taste). LA and BLA in this model also contain four neurons each, such that internal representations are rigidly specified by the connectivity of the model. In a slightly more interesting approach, the authors of [35] generate input vectors representing inputs from cortex, HIPP, and IL: these vectors are ten-dimensional vectors, with one dimension set to the maximum value (1 or 1.5), and the others set to a minimum value (0 or 0.1). These signals are conveyed to LA and BLA, which have 10 neurons; it is unclear whether the dimensionality and number of neurons are set identically to support one-hot representation internally (strong pattern separation), or whether the Rescorla-Wagner learning rule implemented in the model produces mixed representations internally (weak pattern separation). In a much larger-scale model, [114] uses a one-dimensional feature map to represent sensory inputs: each feature has a corresponding neuron in the map, and topographical connections continue to support these one-hot representations throughout the model (although mixing does occur). Finally, the model of [171] uses a 32-bit input stimulus, which contains information about one CS, two CTXs, and one US (each stimulus is an 8-bit vector). However, in the input layer of the network, the representation of different stimuli are orthogonal, such that different stimuli are encoded by different nodes.

The only models we encountered that support feature-rich representations (internally and externally) are the NEF models. The model in [75] uses a dedicated visual system to extract a seven-dimensional emotional state from an input image; these visual neurons pass information to LA, collapsing the representation down to a two-dimensional emotional representation. This representation is used elsewhere in the model, eventually eliciting a behavioral response based on winner-take-all competition. The model in [258] also uses high-dimensional input representations to drive a salience-detection and motivational circuit, which uses a number of brain areas, including AMY. Both these models support real-world applications, including emotional analysis from images and robot navigation, demonstrating that high-dimensional representations have practical utility.

Our models represent CS and CTX as high-dimensional vectors that drive spiking activity in neural populations outside amygdala according to Eq. 1.2. Input vectors, as well as the encoders in the external populations, are drawn by sampling vectors from an N-dimensional hypersphere: doing so ensures that vectors are normalized to unit length, and increases the probability that at least one neuron will respond to any given input. Within AMY, activities in LA and BLA represent the learned response to CS and CTX inputs;

rather than represent these inputs directly, our AMY nuclei encode signals that indicate the "salience" of the current input. By framing AMY nuclei representations in this salience-space, we were able to formulate error-driven learning rules: if the salience estimate is low when it should be high (or visa versa), the connection between the external population and the AMY nucleus must be updated accordingly. The advantage of the PES learning rule, as opposed to a Hebbian learning rule, is that it operates in the high-dimensional space of the input signal, rather than the low-dimensional space of neural activity and weights. PES leverages knowledge about the existing representational properties of AMY neurons ($\mathbf{e}_i$) and the input space ($\mathbf{d}_i$) to update entries in the weight matrix based on how much they contribute to the error. This helps preserve important information about the input feature space when external inputs are collapsed into AMY salience representations. One natural consequence is that AMY's salience response generalizes to similar inputs: as CS and CTX change, the salience response changes proportionally, leading to a smooth generalization gradient. This is why our model can be applied to fear generalization experiments without any additional mechanisms.

Another notable feature of our model is that it naturally produces the neural responses seen in LA and BLA. Many computational models treat BLA fear and extinction neurons as separate populations that are connected via an inhibitory microcircuit [35, 62, 160]: their CS-induced responses are typically a direct result of this connectivity and the unique external signals they receive. In our model, all pyramidal neurons in LA and BLA receive the same information from the external populations (although the connection weights are initialized to zero before training), and do not receive direct projections from BLA inhibitory neurons. We found that the CS-responses of our LA and BLA neurons naturally diverge as learning proceeds: some neurons become more responsive to CS or CTX, and some become less responsive. This simplified picture of AMY connectivity, which does not require interneurons as intermediary controllers between BLA interneurons, may provide a more parsimonious picture of AMY representation; more comparative computational work and validating empirical experiments are needed to resolve this question.

Learning Rules

Computational models of AMY use a variety of learning rules for associative learning and salience detection. In biophysical models, Hebbian learning is common: these learning rules depend on local calcium concentrations, which vary based on pre- and postsynaptic activity [183, 141, 126]. Rate-based Hebbian learning, where weight updates explicitly depend on pre- and postsynaptic firing rates, are also common in both biological and mathematical models [160, 171, 158]. Other models extend rate-based Hebbian learning

to include an error term, such that the learning rules resemble the Rescorla-Wagner rule [35] or reinforcement learning [114]. In other models, synaptic update is governed by coincident arrival of CS and CTX US inputs [248]. Finally, several computational models do not simulate learning at all, but are instead focused on salience detection within larger networks [75, 258].

Our model uses a learning rule that differs from all of these mechanisms, but has been used in many other cognitive models [64, 189, 249]. PES learning (Eq.1.13) is not Hebbian, because it does not explicitly depend on postsynaptic activity. Instead, it is driven by an error signal and by presynaptic activity, and because it updates weights in a factorized manner (Eq. 1.5). We discussed above how PES facilitates feature-rich associative learning that produces conditioned fear responses, context-dependent extinction, and graded fear responses. These functional results validate the use of PES in models of fear conditioning. What's more, while Hebbian learning and STDP are often touted for their biological plausibility, PES may also be feasibly implemented in a biological networks. One could even make the argument that PES learning in our model resembles Hebbian learning, insofar as it uses pre- and postsynaptic activities to determine weight updates. Suppose that our error populations, rather than being separate neural populations, were physically located within the dendritic tree of pyramidal neurons in LA and BLA. These units would receive feedback from the pyramidal cells themselves (postsynaptic activity conveying CS or CTX responses), feedforward signals from interneurons (conveying US information and gating learning), and feedforward signals from external populations (presynaptic activity conveying CS and CTX information). In this way, the dendrites would have all the information needed to realize PES learning, and would only need to perform simple computations (subtraction and multiplication) to update weights. While an exploration of this perspective is outside the scope of this chapter, it does suggest that learning in our model may not be so different from the plasticity mechanisms used in the models above.

2.4.4 Empirical Validation

The final dimension for comparing models of AMY is validation with empirical data. Of course, different models are built at different scales and seek to explain different phenomenon, so we should not expect any model to reproduce the full spectrum of empirical data available in fear conditioning experiments. Still, noting whether a model seeks to validate its findings through empirical comparison, and identifying which classes of data it can and cannot explain, is an important task for any theoretical or computational model.

The AMY models discussed above fall into two categories: those that estimate salience and coordinate other cognitive systems; and those that learn fear and safety associations

between stimuli. Models in the former class [114, 75, 258] have traditionally focused less on empirical data, and more on describing AMY's functionality within a larger cognitive system. This form of validation is critically important, as it demonstrates how AMY contributes to embodied cognition, but it cannot really be labelled "empirical", since it does not attempt to recreate specific neural or behavioral data from biological systems.

The most common form of validation in AMY models is the reproduction of the neural activities and behaviors associated with fear conditioning. Many of the above models showed that learning within AMY leads to changes in the CS-responsiveness of individual neurons [141, 126, 183, 248, 160, 171]. Specifically, these models identified simulated neurons whose activities increased following conditioning, or decreased following extinction. In these models, high-level validation compared mean CS-induced activities throughout the experiment, while low-level validation compared simulated spike trains to empirical data. Other computational models decoded neural activities to show a change in behavioral fear responses following training [141, 35]. Here, high-level validation compared mean fear responses throughout the experiment, and low-level validation compared learning trajectories sampled after each CS presentation. All of the above fear conditioning models also examined extinction, but only a subset investigated the relationship between context and extinction [35, 171]; the other models either do not have the representational power to consider multiple contexts, or permit extinction to unlearn the original association.

Another common means of empirical validation involves simulated experiments in which specific neurons or populations are externally inhibited. Several of the models described above [171, 248, 183] externally inactivated parts of the model (via simulated GABAergic inhibition or physical lesioning), and examined the effects on either neural activity, or on behavioral responses. These comparisons were usually qualitative, but the biophysical models directly compared simulated spike data to electrophysiological data [183]. Finally, some models were not concerned with empirical validation at all [158]; these models were either too biologically simple to compare to empirical data, or simply did not make an effort at direct comparison.

Our model is validated against a wide variety of empirical data: we (a) identified neurons whose CS-induced neural activity changes following conditioning and extinction, comparing their mean firing rates and counting the number of neurons in each category; (b) decoded neural activities to estimate fear responses, and showed how they differed in various test contexts; and (c) simulated external excitation and inhibition, and observed the effects on fear responses. In all these cases, we provided citations for empirical studies that qualitatively support our results. Finally, we (d) simulated fear generalizations experiments, observing how the presentation of novel stimuli and contexts produced gradients of fear responses. We also varied neuron parameters to induce different degrees of

pattern separation in our neurons, and compared their generalization curves. We quantitatively compared these data to recent empirical experiments involving fear generalization in healthy and anxious patients. Among the models of fear conditioning in AMY that we studied, ours is the only model that investigates the origins and effects of fear generalization (although a few models of generalization in HIPP and cortex do exist, see [8, 132]).

2.4.5 Summary

To summarize, our AMY model extends previous computational models of fear conditioning in several respects. First, in contrast to AMY models that investigate the interaction of two or three neural populations, we recreate the detailed anatomy of multiple neural populations in AMY, mPFC, cortex, and basal ganglia: the connections between our excitatory populations are anatomically realistic, and we simulate inhibitory populations that control the dynamics of fear conditioning. Second, rather than using a Hebbian learning rule, our model uses an online error-driven learning rule: this mechanism realizes associative learning at a symbolic level of abstraction, and its dynamics naturally produce neural responses that must be hard-coded in other models. Third, we used sophisticated representation of fear stimuli and environmental contexts: while other models use low-dimensional inputs to represent external information, we use high-dimensional inputs to encode multiple features of the external world. These representations, when realized in an anatomically-realistic network, allow us to simulate a variety of fear conditioning paradigms, ranging from traditional acquisition and extinction experiments, to lesion studies, to fear generalization experiments, with a single model. Finally, while most computational models compare their results to only a single class of empirical data, we validate our model using both neural and behavioral data; we also identify gaps in our understanding of fear conditioning and make predictions about the outcome of future experiments.

2.5 Conclusion

In this chapter, we presented a computational model of the amygdala (AMY) that learns to associate neural stimuli (CS) and environmental contexts (CTX) with the presence or absence of a negative event (US). Our model recreates many anatomical features of AMY, notably the divisions and connections between its nuclei, and the response properties of its pyramidal and inhibitory neurons. We trained the model using online, error-driven learning rules that update the connection weights between brain areas like cortex and hippocampus, which represent input stimuli, and the lateral or basolateral amygdala. These nuclei excited

neurons in the central amygdala, generating or suppressing fear responses as appropriate. We ran numerous experiments on the model, identifying neurons with different emergent response profiles, measuring fear expression in various test contexts, externally inactivating parts of the model, and observing how fear responses varied as we presented the model with novel stimuli. We found that many of our results were consistent with empirical data, and discussed the similarities and differences between our model and other computational models of AMY.

Future work can extend this model in many directions. In our simulations, we presented the US concurrently with the CS; the simultaneous presence of these signals was required for learning in our model. However, in many empirical experiments, the US is presented for a short duration following the cessation of the CS. Our model cannot currently handle this experimental protocol. Luckily, recent work using the NEF has shown that associative learning between neutral and aversive stimuli is possible when the US is delayed by a fixed duration, for instance in eyeblink conditioning experiments [226]. Our model also does not exhibit effects such as secondary conditioning (pairing US with CS1, then pairing CS2 with CS1, will produce a fear response to CS2), spontaneous recovery (forgetting contextual safety associations are slowly forgotten over time), reinstatement (presenting the US in CTX- abolishes contextual safety associations), or rapid reacquisition (replaying a previously-paired CS-US stimuli leads to faster learning than a novel CS-US pair).

Beyond these specific effects, there are a number of high-level functional shortcomings of our model. Notably, the contextual fear associations learned in our model do not support complex interactions between the cue and the context. Our model learns that, in CTX-, when the CS is present but the US is not, that CTX- is safe. While this learning rule uses the presence of the CS to guide learning, the subsequent association depends only on the CTX, not on joint information about the CTX and the CS. A more interesting and versatile association might learn that the combined presence of CTX- and CS+ predicts safety, but that CTX- alone does not imply safety. A related shortcoming is that representations within AMY nuclei are quite simple, indicating only whether various combinations of external stimuli are salient or not. Some evidence suggests that AMY itself is involved in pattern separation, and it seems likely that the brain uses AMY to judge salience in a multi-dimensional space (i.e., with respect to several independent goals). Significant functional changes would be required to support these capacities.

In the opposite direction, the biological realism of our AMY model could be expanded in two ways. First, we could replace our LIF neurons with detailed neuron models whose electrophysiology better matched the idiosyncrasies of pyramidal neurons and GABAergic interneurons. This would allow biophysical manipulations, such as the direct injection of dopamine, norephinerphine, or oxytocin to the model; currently, these perturbations can

only be simulated at an abstract level, for instance by injecting current or changing the synaptic gains. Second, we could begin to model inhibitory microcircuits in LA and BLA with greater fidelity, which would open avenues for investigating the role of interneurons in pattern separation and memory consolidation. In particular, it would be interesting to see whether the NEF learning rule proposed by [29], which changes neural encoders to facilitate associative memory and pattern separation, could be realized within an anatomically-detailed network.

Finally, our AMY model should be embedded in larger cognitive networks. Many NEF models combine several dedicated cognitive modules into larger networks in order to perform complex cognitive tasks and reproduce empirical data [64, 66, 38]. Other AMY models have shown how this region may help assign salience within a larger task-performing network [75, 258, 114]. In the remainder of this thesis, we explore how our AMY model might interface with other cognitive systems, particularly those related to social cognition. For instance, in Ch. 3, we discuss how emotional systems like AMY may be used to module decision making systems, accelerating their dynamics to produce faster decisions at the expense of accuracy; and in Ch. 5, we discuss how AMY is part of a cognitive network that uses associative learning to estimate the trustworthiness of other individuals based on prior experience. Within the fear conditioning paradigm, it would be interesting to investigate how cortical areas differentiate a world full of stimuli into isolated stimuli and broader contexts, or to study how higher-level reasoning about causal relationships and object categories may influence the fear conditioning process. Building a model of how fear associations and salience detection modulate attention and sensory discrimination, two functions that are frequently attributed to AMY, could be particularly important for understanding how AMY contributes to emotional processing in general. We hope to investigate many of these topics in future work.

Table 2.1: Comparison between computational models of AMY. See Sec. 2.4.3 for details.

Reference	**Neurons**	**Representation**	**Learning Rules**	**Validation**
Carre [35]	mean-field	one-hot	Hebbian (error)	behavior context
Mannella [158]	rate-based	one-hot	Hebbian (rate)	none
Moustafa [171]	rate-based	one-hot	Hebbian (rate)	activity context ablation
Materra [160]	rate-based	current-based	Hebbian (rate)	activity
Vlachos [248]	spiking	current-based	signal timing	activity ablation
John [114]	spiking	one-hot	Hebbian (error)	none
Fischl [75]	spiking	high-dimensional	none	none
Zelgen [258]	spiking	high-dimensional	none	none
Pendyam [183]	biophysical	current-based	Hebbian (calcium)	activity ablation
Li [141]	biophysical	current-based	Hebbian (calcium)	activity behavior
Kim [126]	biophysical	current-based	Hebbian (calcium)	activity
Duggins	spiking	high-dimensional	error	activity ablation behavior context generaliza- tion

Chapter 3

Biologically Detailed Models and Working Memory

Author's Note: the majority of content in this chapter was previously published as a journal article in PLOS Computational Biology [54]. Code is available on GitHub.

3.1 Introduction

Biological detail is an important concern for neural models that seek to bridge the gap between neural and cognitive processes. By including synaptic and cellular mechanisms in cognitive models, researchers may investigate aspects of psychology, such as biologically-grounded mental disorders and their pharmacological treatment, that often cannot be studied with simpler models. While the benefits of modelling biological detail have been widely recognized within the computational neuroscience community, doing so makes model construction and analysis significantly more difficult. As a consequence, most researchers compromise on either biological realism or cognitive capacity when building cognitive models. For example, cognitive architectures like ACT-R [6] have produced numerous models of cognition that reproduce behavioral data, but these models use production rules and activation functions that abstract away from the brain's neural substrate, limiting their ability to investigate many neurological phenomenon. At the opposite extreme, the Human Brain Project [159] has produced models that recreate the anatomy and spiking behavior of entire cortical microcircuits, but these models do not perform recognizable neural computations or produce behavior, limiting their utility to investigate many cognitive phenomenon.

To study the biological foundations of the human brain and design biologically-inspired cognitive algorithms, we need models that unify biophysical detail and cognitive capacity.

Two aspects of biological detail may be especially important in understanding learning and decision making: the interplay between excitatory and inhibitory neurons, and the role of modulatory neurotransmitters, or "neuromodulators". In many social cognitive areas, GABAergic interneurons play an important role in learning, consolidation, and control [109]. To fully understand how they accomplish these functional capabilities, we need to build models that specifically simulate the biological details of these distinctive neurons. In Ch. 2, we saw that a minimal functional model of the amygdala produced several behavioral results from the fear conditioning literature, but an anatomically-detailed model that simulated the interplay of pyramidal cells and interneurons reproduced a wider class of empirical data [153]. In other brain systems, the rapid dynamics of the neurotransmitter GABA strongly influences network dynamics through inhibitory control and gating: in both Ch. 4 and 5, we build models that require fast and precise inhibition to control the output of individual populations. Neuromodulators also play an important role in social cognition. Many emotional systems use hormonal signals to broadly modulate cognition in the cortex, for instance by using oxytocin to manipulate prosocial evaluations made in AMY [148]. Furthermore, many learning systems rely on modulators like dopamine (DA) to facilitate synaptic plasticity [50]. The cellular action of these neuromodulators goes beyond simple excitation and inhibition, altering cellular states in complex ways that change how networks process information. While the exact mechanisms and functional implications of these changes remain unclear, it is likely that modulators like DA, seratonin (5HT), and norepinephrine (NE) play an important but functionally (and biophysically) subtle role in social cognition [13, 238, 234].

Several existing frameworks leverage biological neural networks to perform cognitive tasks. Of course, the NEF and SPA use large-scale, biologically-constrained spiking neural networks to study the functional aspects of cognition. Similarly, the Leabra cognitive architecture [179] uses biologically-plausible learning rules to capture the functionality of major cognitive systems. Other approaches emphasize dynamics within neural networks, which, when properly controlled, can be used to perform mathematical transformations of represented information and issue behavioral commands. The efficient coding hypothesis [48] uses cortical connectivity to produce controlled spiking networks, while FORCE [229] uses online learning rules in recurrent networks to implement dynamical systems. While each of these frameworks emphasizes the role of certain biological features for cognition, little attention has been paid to the complexity of individual neurons, the fundamental unit in most intelligent systems. For the most part, these architectures assume simple models for neurons, ranging from rate-mode neural assemblies to LIF point neurons.

In this chapter, we propose a method for training and simulating networks that contain biologically-detailed neuron models and perform useful cognitive operations. We label this method "oracle-supervised Neural Engineering Framework" (osNEF): it is an extension of core NEF principles that uses an "oracle", a parallel network that is used during training to supervise the learning process. One major advantage of osNEF is that it treats the neuron model as a "black box", relying on learning rules that only consider the spiking inputs and outputs to each cell. Because it does not rely on detailed knowledge of cellular dynamics, osNEF can be applied to a wide range of neuron models without major changes to the algorithm. To facilitate easy adoption of osNEF, we develop an interface where modellers may plug in existing neuron models, written in Python or NEURON, into Nengo models. This approach lets modellers specify detailed low-level mechanisms like conductance-based synapses, voltage-gated ion channels, and dendritic geometry, then train the network to realize high-level dynamics or computations. When properly applied, this allows researchers to investigate questions that relate low-level biological details to high-level cognitive capacity.

Our goal is to show that osNEF can be used to construct a variety of functional neural networks from various biologically-detailed components. To demonstrate these capabilities, we apply osNEF to produce two classes of cognitive functionality. First, we highlight the broad applicability of osNEF by simulating networks of biologically-detailed neurons and training them to realize dynamical systems. We construct and train four functional networks, which compute a variety of useful cognitive operations, and simulate them with four different neuron models, which range in complexity from a LIF point neuron to a spatially-extended pyramidal cell. Second, to demonstrate a concrete cognitive application, we construct a biologically-detailed model of working memory in PFC that performs an idealized memory task. The network is composed of anatomically-detailed reconstructions of pyramidal cells and interneurons, both containing numerous geometric compartments and ionic currents, and complex synapse models, including conductance-based NMDA, AMPA, and GABA synapses. We show that the mnemonic performance of the model is consistent with empirical data from a standard test of working memory, the delayed match-to-sample task (DMTST). We conclude with a discussion of the strengths and limitations of osNEF, including its biological plausibility and cognitive generality, and by comparing it to similar approaches.

3.2 Oracle-Supervised NEF

As the name implies, osNEF extends the Neural Engineering Framework with the goal of simulating more complex neuron and synapse models. To introduce the goals of osNEF, we first discuss the notion of target tuning curves. Afterwards we define the two central methods of osNEF: an online learning rule for encoders and decoders, and an offline optimization for synaptic time constants. We finish by defining the neuron models used to test these methods.

3.2.1 Target Tuning Curves

State space representation in the NEF requires that a population of neurons be sensitive to the whole range of $\mathbf{x}$, otherwise parts of the state space will have low signal-to-noise ratios and the accuracy of the computation will suffer. Previous work has extensively studied how different distributions may effectively represent a state space and dynamically compute functions [65, 250]. One effective distribution is shown in Fig. 3.1: these tuning curves have x-intercepts that vary uniformly across the state space, meaning they are inactive for some values of $\mathbf{x}$ and otherwise have state-dependent activity; and they have y-intercepts that vary uniformly over some range, meaning their spike rates change by different amounts per unit change in $\mathbf{x}$. The combination of these constraints makes it easy to accurately decode an estimate $\hat{\mathbf{x}}$ from the collection of neural activities.

When building NEF networks, the modeller usually specifies a target range of x- and y-intercepts for a population of n neurons, which defines a distribution of tuning curves like the one shown in Fig. 3.1. Nengo then uses an algorithm to find gains α and biases β (Eq. 1.2) that will realize this distribution. Unfortunately, the algorithm relies on several assumptions that become problematic when simulating biologically-detailed neuron models. In the algorithm, Nengo generates a series of input currents, delivers these currents to one instance of the neuron model, and measures the resulting firing rates. Next, it uses an interpolation method to estimate how firing rates relate to input currents. From this, it calculates the input currents required to (a) make the neuron start firing, and (b) bring the neuron to the specified maximum firing rate. Finally, for each neuron i in the population of n neurons, the algorithm solves for the gain and bias needed to map between the input currents and the specified and x- and y-intercept.

There are two problems with this algorithm for biologically-detailed neurons. First, the algorithm operates in current-space, and has access to a direct bias current β when controlling the neuron's activity. In biological networks, neurons do not have an arbitrary

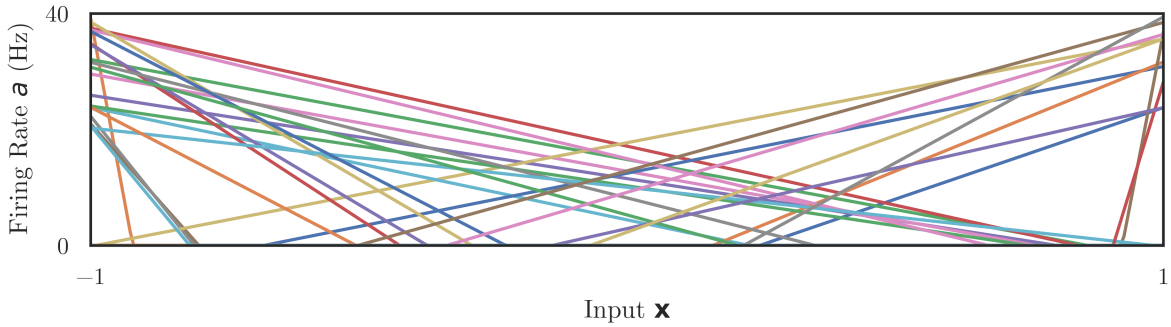

Figure 3.1: Target tuning curves. We use the standard Nengo method to generate a population of 30 ReLU neurons with x-intercepts in -[1, 1] and y-intercepts in [20, 40].

source of external current driving their activity: β can only be realized by the internal physiology of the neuron itself, or through external synaptic input. While decoding an effective β from synaptic input is possible, the broader issue is that this algorithm reduces synaptic inputs into a one-dimensional value for input current. This is a valid model when spikes are delivered through "current-based" synapses onto point neurons, but it cannot adequately describe the consequences of either (a) delivering spikes through "conductance-based" synapses, or (b) delivering spikes to cells with extended geometries.

The second problem with the algorithm is that it assumes neurons have a well-defined firing rate for every given input current (or associated value of $\mathbf{x}$). Real neurons are highly adaptive: even when driven with a constant input stimuli or current, their firing rates fluctuate in complicated ways. A complete identification of how firing rates depend on inputs must thus account for the passage of time, which cannot be easily handled by this algorithm: temporal nonlinearities also become an issue when solving for decoders, as we discuss below. Although some methods have successfully identified the "temporal kernels" realized by complex adaptive neurons [209, 110], these methods are quite involved, and often rely on expert knowledge about the specific neuron model being identified.

Rather than adapt Nengo's algorithm to accommodate detailed neurons, we devise a new method with the same goal: to find neural parameters that produce neural activities resembling a target tuning curve. Target tuning curves can be specified in many ways, but regardless of how we specify them, the distribution should have three features: it should qualitatively match in-vivo elecrophysiology; it should be achievable with our detailed neuron models; and it should support state space representation given NEF-style encoding and decoding. Fortunately, the tuning curve distributions in most NEF models meet both

the first and third criteria, so we use Nengo's default algorithm to generate our target tuning curves, making sure to choose parameters that lead to plausible electrophysiology.

Fig. 3.1 shows a typical distribution of target tuning curves for osNEF. These curves were generated using spiking rectified linear neurons (ReLUs), which are parameterized by an x-intercept (the value of the input current below which a neuron stops firing) and a y-intercept (the neural activity when the input is at its extrema). We use ReLUs to generate target tuning curves for three reasons. First, ReLUs are (arguably) the simplest spiking neurons that still capture an essential neural nonlinearity: the transition from a region (in state space) where the neuron remains (mostly) inactive, to a region where activity increases monotonically (as state inputs change). Second, networks of ReLUs are cheap to simulate but functionally powerful: deep neural networks populated with ReLUs can be trained to perform a wide variety of complex tasks [1]. Third, ReLU parameters are intuitively aligned with the parameters α and β from Eq. 1.2: α directly determines the slope of the linear tuning curve, while β determines the input value for which these neurons begin firing. In contrast to these benefits, the ReLU response function is a poor fit for most electrophysiological data. Notably, as inputs increase, ReLU firing rates increase linearly without bounds, while the activity of biological neurons are constrained to some maximum value, such that firing rates will plateau as neural activity approaches this value. In Sec. 3.3.1, we explore the degree to which biologically-implausible target tuning curves affect osNEF's ability to train networks of detailed neurons.

3.2.2 Online Learning Rule for Encoders and Decoders

Given a set of target tuning curves, our first objective is to train synaptic parameters such that, when a population of complex neuron models is simulated within a network, the observed spike rates resemble the rates given by the target tuning curves. The network and procedure used to train these parameters are depicted in Fig. 3.2 and summarized in Table 3.1. We begin with an input signal $\mathbf{x}(t)$ that is fed into two streams. The top stream of Fig. 3.2 is the *oracle*, where the desired transformations of $\mathbf{x}(t)$ are computed analytically (i.e., Eq. 1.8) and used to generate the target spikes; and the bottom stream of Fig. 3.2 is the neural network, where $\mathbf{x}(t)$ is represented by neural spikes and the desired transformations are realized through weighted connections between neural populations. In the oracle stream, $\mathbf{x}(t)$ passes through filters (rectangles), which convolve $\mathbf{x}(t)$ with a filter $h(t)$, and nodes (diamonds), where state space transformations are applied. Finally, the oracle stream feeds this (filtered, transformed) signal into a population of neurons *tar* that realize the target tuning curves; this generates the neural activities $a^{\text{tar}}(t)$ that osNEF will use to train the neural network proper.

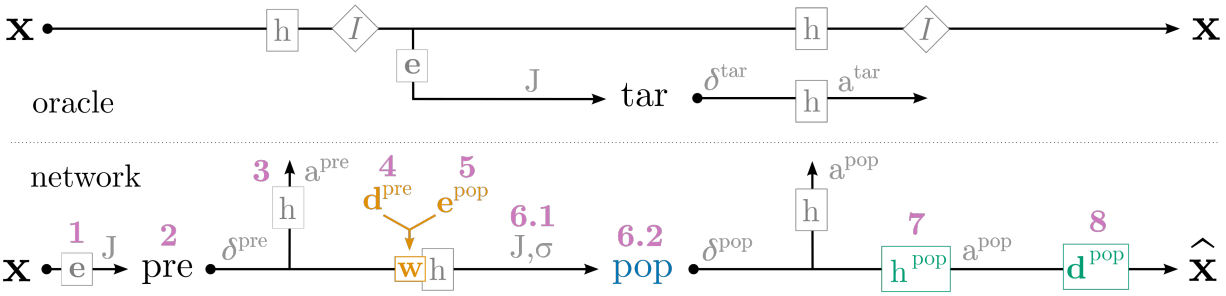

Figure 3.2: Network used during osNEF training. The top half of the figure is the oracle stream, where the desired filters and transformations are applied analytically, and where the target activities are generated. The bottom half of the figure is the network stream, where synaptic connections realize the desired filters and transformations, and where osNEF training is applied to update the relevant synaptic parameters. Both streams are driven by an input $\mathbf{x}(t)$. Arrows represent the signal travelling through each stream. Boxes letters (filters h, weights w, transformations I, and decoders $\mathbf{d}$) indicate mathematical operations being applied to the signal. The resulting quantities (spikes δ, synaptic currents J, synaptic conductances σ, and states $\mathbf{x}$) are shown above the arrow. The pink numbers reference Table 3.1, which lists the operations that are applied at each step. *pre* and *post* are neural populations, which receive synaptic inputs and produce spikes. Coloration indicates ReLU neurons (black) or detailed neurons (blue), measured signals (gray), parameters updated by osNEF's online learning rules (orange) or offline synaptic optimization (green), references (pink), and NEF operations (gray).

In the bottom stream of Fig. 3.2, the input signal drives a population of preliminary neurons *pre* as per standard NEF encoding: $\mathbf{x}(t)$ is converted to a synaptic current that drives dynamics in the neuron model. We arbitrarily chose *pre* to contain ReLU neurons, but any neuron type compatible with standard NEF techniques will do. Neurons in *pre* generate spikes δ^{pre} over time, which are smoothed into a real-valued neural activity signal $a^{\text{pre}}(t)$ by convolution with the synaptic filter h according to Eq. 1.6. These activities are then weighted by synaptic weights w and delivered to the population of interest, *pop*, which contains detailed neurons. These weights must be trained by osNEF such that the neural activies from *pop*, $a^{\text{pop}}(t)$, have the desired tuning curves. Fortunately, the oracle stream provides neural activities $a^{\text{tar}}(t)$ that are guaranteed to realize these desired tuning curves, because *tar* is driven by the same signal that drives *pop*. Therefore, if $a^{\text{pop}}(t)$ closely matches $a^{\text{tar}}(t)$, we can say that *pop* implements the desired tuning curves.

In order to achieve this, we must train the weights w such that, as spikes from *pre* are

Table 3.1: Summary of equations used during osNEF training, as depicted in Fig. 3.2

Label	Reference	Notes
1	Eq. 1.1	converts state space to input current via encoder
2	Eq. 1.2	G is a spiking ReLU neuron
3	Eq. 1.6	converts discrete spikes to smoothed neural activity
4	Eq. 1.13	online update of decoders, co-determines synaptic weight
5	Eq. 3.2	online update of encoders, co-determines synaptic weight
6.1	Eq. 3.3	dynamics for synapse (current- or conductance-based)
6.2	Sec 3.2.4	dynamics for neuron model (may be coupled with synapse)
7	Sec 3.2.3	offline update of synaptic time constants and decoders
8	Eq. 1.4	converts neural spikes to state space estimate

weighted, passed through the synapse h, and run through the neuron model G, $a^{\text{pop}}(t)$ approaches $a^{\text{tar}}(t)$. This is the first objective of osNEF training, and is accomplished using online learning. First, we decompose w into encoders and decoders; for each presynaptic neuron i and postsynaptic neuron j, the corresponding entry in the weight matrix is

$$w_{ij}^{\text{pre-pop}} = \mathbf{d}_i^{\text{pre}} \cdot \mathbf{e}_{ij}^{\text{pop}}. \tag{3.1}$$

where $(\cdot)$ is the dot product, $\mathbf{d}_i^{\text{pre}}$ is the D-dimensional decoder vector for neuron i, and $\mathbf{e}_{ij}^{\text{pop}}$ is the D-dimensional vector for the (i, j) neuron pair. osNEF uses two online learning rules, one to update encoders, and another to update decoders. To update presynaptic decoders $\mathbf{d}^{\text{pre}}$, we use the PES learning rule without any modifications.

For encoders, we introduce a new learning rule:

$$\Delta \mathbf{e}_{ij}^{\text{pop}} = \alpha^{\text{e}} \ \text{sign}(\mathbf{d}_i^{\text{pre}}) \ a_i^{\text{pre}} \ (a_j^{\text{pop}} - a_k^{\text{tar}}) \tag{3.2}$$

where α^{e} is the encoder learning rate, $\text{sign}(\mathbf{d}^{\text{pre}})$ is the elementwise sign of the presynaptic decoder, and (i, j, k) are neuron indices for the presynaptic, postsynaptic, and target neuron populations, respectively. The rule is supervised in the sense that the target activities are provided in real time as the oracle drives *tar*, and the difference between the current and target activities drives the update. The rule also uses information about both presyaptic and postsynaptic activity, making it Hebbian. Note that Eqs. 3.1-3.2 redefine encoders as a tensor, indexed over i, j, and D, rather than a matrix over j and D, as is standard in the NEF (Eq. 1.1). Note that, although we decompose weights into encoders and decoders, only

the combined weights are actually used to transform signals within the network: encoders and decoders are theoretical tools used to analyse the relationship between spike space and state space and to facilitate training, but synaptic transmission between populations of detailed neurons is governed only by weight.

3.2.3 Optimizing Synaptic Time Constants

Continuing on the bottom stream of Fig. 3.2, the spikes generated by *pre* must pass through synapses that (a) convert $\delta^{\text{pre}}(t)$ to the state variables used in the neuron model, and (b) apply the weights w that realize the target transformations. For our simpler neuron models, synapses deliver current to the cells, which directly affects the cell's voltage. For our complex neuron model, synapses update the conductance parameters in the relevant sections of the cell, which then influence transmembrane currents that govern voltage change. In both cases, we assume that the synapse is a second-order lowpass, or double-exponential, filter, whose transfer function is

$$h(s) = \frac{1}{(\tau_{\text{rise}}\ s + 1)\ (\tau_{\text{fall}}\ s + 1)} \tag{3.3}$$

where τ_{rise} and τ_{fall} are time constants and s is in the Laplace domain. Whenever a synapse receives a spike, it updates the postsynaptic cell's input current (or conductance) by an amount proportional to the synapse's dynamical state and its weight. To ensure that our decoded estimates align with the signals being transmitted in the network, we also use the double exponential filter to estimate neural activities from neuron spiking outputs (Eq. 1.6).

This leaves the question of how to choose τ_{rise} and τ_{fall}. When smoothing spikes for the purpose of encoder learning (Eq. 3.2) or synapsing from *pre* onto *pop*, the choice of time constant makes little difference, so long as it sufficiently smooths spike noise (e.g., $\tau_{\text{fall}} > 10$ms). However, as we show in the Sec 3.3.1, the choice of time constants makes a significant difference when (a) decoding $\hat{\mathbf{x}}(t)$ from *pop*, or (b) connecting one population of detailed neurons to another. We could choose τ values based on the effective time constants of biological neurotransmitters, but it is unclear whether one set of parameters would be appropriate for the variety of neuron types and networks that we investigate. To resolve this problem, osNEF uses a novel offline optimization procedure that finds appropriate time constants given spiking data from the simulated network itself. The optimization procedure is as follows:

1. Simulate the network with input $\mathbf{x}(t)$, record neural spikes $\delta^{\text{pop}}(t)$, and specify the target function $f(\mathbf{x}(t))$.

2. Choose a random τ_{rise} and τ_{fall}, filter the spikes to calculate $a^{\text{pop}}(t)$, and use least-squares to compute decoders $\mathbf{d}^{\text{pop}}$ for estimating the function.

3. Calculate the error between this estimate and the ground truth by computing the RMSE between $f(\mathbf{x}(t))$ and $\hat{f}(\mathbf{x}(t))$.

4. Repeat Steps 2 and 3, using the optimization package Hyperopt [25] to search the space of possible time constants with the objective of minimizing the error.

Returning to Fig. 3.2, this procedure is used to train the filters and decoders for *pop*, h^{pop} and $\mathbf{d}^{\text{pop}}$. Filtering spikes from *pop* produces neural activities $a^{\text{pop}}(t)$, from which we can decode an estimate $\hat{\mathbf{x}}(t)$ according to Eq. 1.4. This completes the bottom stream, which shows how a state space input may be translated into neural spikes, transformed to realize particular mathematical operations via synaptic connections, then translated back to a state space estimate.

3.2.4 Neuron Models

To test whether osNEF is capable of producing functional networks that contain neuron models of varying complexity, we investigate four neuron models: the LIF neuron, the Izhikevich neuron, the Wilson neuron, and a Layer V Pyramidal Cell compartmental model.

The LIF neuron model is a point neuron that approximates the membrane dynamics preceding and following an action potential. Although the resulting voltage traces do not quantitatively align with electrophysiological recordings, the LIF neuron does capture key features of neural behavior, namely integration of inputs, leak towards a resting potential, reset following a spike, and a refractory period. It is also extremely cheap to simulate, as voltage dynamics are governed by a single equation. As such, LIF neurons are widely used in simulations that seek to balance biological realism, computational scalability, and analytical tractability (see [65]).

Although LIF neurons are fast and functional, they do not quantitatively capture the dynamics of membrane potential. The Izhikevich neuron model [113] is another simple neuron model that captures a wider variety of spiking behavior characteristic of biological neurons. The model has only four free parameters and two state variables, but certain configurations of these parameters may produce regular spiking, intrinsic bursting, fast

spiking, chattering, and many more interesting dynamics. As such, this neuron model is useful in networks where both scallability and electrophysiological realism are important.

While the LIF and Izhikevich neurons are useful and computationally cheap neuron models, they do not simulate the action potential in detail, instead using hand-crafted reset mechanisms when the cell's voltage crosses a fixed spike threshold. For our third neuron model, we chose an intermediate-complexity neuron developed by Wilson [256] that extends the FitzHugo-Nagumo equations [76, 175] to incorporate electrophysiological detail, including Ohm's Law and equilibrium potentials of four ionic currents in neocortical neurons ($I_{\mathrm{k}}, I_{\mathrm{Na}}, I_{\mathrm{T}}, I_{\mathrm{AHP}}$). The resulting model consists of three coupled ODEs representing voltage, conductance, and recovery, can generate realistic action potentials, and naturally produces adaptation, bursting, and other neocortical behaviors [256]. Due to the lower number and cubic dynamics of the underlying equations, simulation is still relatively fast, but a smaller timestep is required to avoid numerical errors.

When describing electrophysiology in detail, the most widely-used formalism is the Hodgkin-Huxley model, which we use for our final neuron model. Reproduced from Durstewitz, Seamans, and Sejnowski [61], this model is an anatomically-detailed reconstruction of pyramidal neurons that includes four compartments (soma, proximal-, distal-, and basal-dendrites) and six ionic currents (two for sodium, three for potassium, and one for calcium). The Durstewitz reconstruction accurately reproduces the neural activities of layer-V pyramidal neurons in rat PFC, cells that are known to be active during the delay period of working memory tasks. This neuron model is implemented in NEURON and uses conductance-based synapses, distributed randomly on the three dendritic compartments.

3.3 Results

To demonstrate that osNEF is capable of training cognitively-useful neural networks built from a variety of neuron models, we divide our results into three sections: representation, computation, and application. First, we show that osNEF produces populations of neurons with the desired tuning properties, and demonstrate that their spiking responses *represent* the target signal. These results indicate that an input signal may be encoded and decoded effectively by a single population of biologically-detailed neurons using weights and filters trained using osNEF. Next, we simulate networks containing multiple populations of biologically-detailed neurons, and show that the synaptic connections between them *compute* specific functions and exhibit the target dynamics. These results show that osNEF combines online learning and offline optimization to realize encoding, decoding, and dynamics in a single synaptic process that occurs between populations of detailed neurons.

Finally, we *apply* these techniques to train a biologically-detailed model that performs a working memory task. These results suggest that osNEF is a useful tool for researchers who wish to build, manipulate, and validate biologically-detailed simulations of cognitive systems.

3.3.1 Representation

First, we show that encoding and decoding are possible with a single population of detailed neurons. We simulate four networks using the architecture shown in Fig. 3.2; each network has one LIF, Izhikevich, Wilson, or Pyramidal neuron in the *pop* population, and one ReLU neuron in the "target" population. The networks are trained using the online learning rules described in Sec 3.2.2, then presented with a novel input during testing. The top and middle panels of Fig. 3.3 show the input signal $\mathbf{x}(t)$ (smoothed, 1Hz band-limited white-noise) and neural activities over time, respectively. In response to this input, all four neurons dynamically exhibit spiking activity that closely aligns with the spiking activity of the target ReLU neuron. The bottom panel of Fig. 3.3 shows the observed tuning curves calculated from these data: for each timestep in the simulation, we record the state space value of the input signal $\mathbf{x}(t')$ and the smoothed neural activity $a(t')$ at that time. We divide the state space into 21 equally-sized regions (bins), then associate each $a(t')$ with the appropriate bin. Finally, we plot the mean neural activity for each bin.

Examining the observed tuning curves in Fig. 3.3, we see that all four trained neurons have x- and y-intercepts that closely align with the intercepts of the target ReLU neuron. However, the y-intercept of the ReLU neuron is higher than the trained neurons. This result is expected: ReLU activities ramp without bounds, while the activities of realistic neuron plateau. Although this is, generally speaking, a danger of using ReLUs (or other unrealistic neuron models) as spike space targets, it is not problematic in our simulations for two reasons. First, the differences between the ReLU and trained tuning curves are minimal: in Fig. 3.3, ReLU activities only exceed the trained activities for extreme inputs ($\mathbf{x} > 0.8$), and these differences fall within the observed confidence intervals. Second, because osNEF training minimizes the differences between the spiking activity of a trained neuron and a target neuron, trained neurons will often match the "physiologically plausible" features of a target tuning curve but fail to match any "physiologically implausible" features. This tendency can also be observed in Fig. 3.3: for most inputs ($\mathbf{x} < 0.8$), our trained neurons qualitatively match the inactive and linear segments of the target ReLU curve, but for extreme inputs ($\mathbf{x} > 0.8$), our trained neurons begin to plateau, failing to match the target tuning curve *but* retaining realistic response properties.

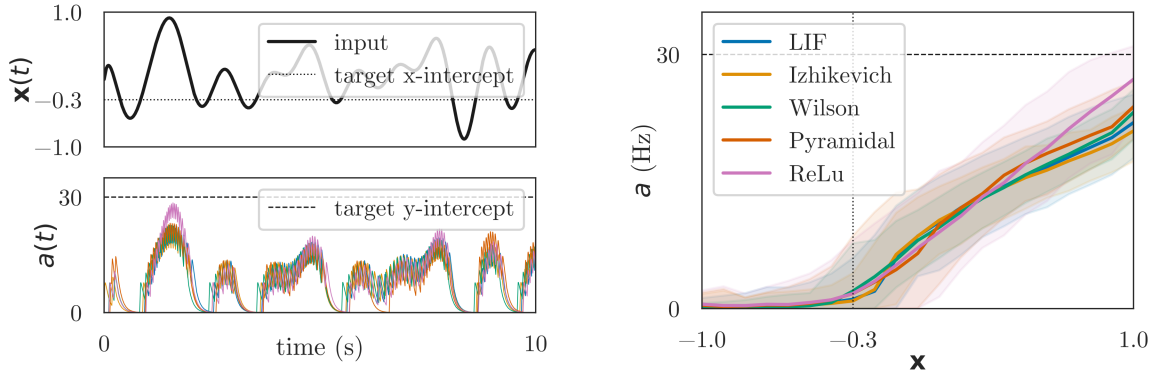

Figure 3.3: Encoding and tuning curves. The top left panel shows the input signal and the target x-intercept, the state space value at which our neurons should begin spiking. The bottom left panel shows neural activities and the target y-intercept, the desired neural activity when our neurons are driven with maximum input. The right panel shows the tuning curves derived from these data. All four neurons exhibit minimal spiking activity when the input is below the target x-intercept, and their activities increase with $\mathbf{x}$ up to the target y-intercept. Shaded error regions are 95% confidence intervals for neural activity.

Ultimately, any target tuning curve will respond differently than the neuron model being trained. By choosing a target tuning curve that is manifestly *im*plausible in some respects, then showing that our trained neurons still exhibit reasonable tuning curves, we demonstrate that osNEF does not require perfect selection of target tuning curves. More importantly, Fig. 3.3 demonstrates that osNEF trains encoders effectively: our learning rules produce dynamic activities that closely resemble the target tuning curves for all four neuron models, despite significant differences in neural complexity and cellular dynamics.

To investigate decoding, we simulated four populations that contained 100 LIF, Izhikevich, Wilson, or Pyramidal neurons, and used osNEF to train encoders, decoders, and readout filters for each population. An interesting problem arose when we used Eq. 1.4 and Eq. 1.6 to decode the neural activity and estimate the state space representation in *pop*. When we used a default filter to smooth the spike trains ($\tau_{\text{rise}} = 1\text{ms}$ and $\tau_{\text{fall}} = 100\text{ms}$), our decoded estimates $\hat{\mathbf{x}}(t)$ were often phase-shifted to the left of the target $\mathbf{x}(t)$, leading to systemic error, as shown in Fig. 3.4. The phase shift is more pronounced in neurons with greater observed spike adaptation or variance in interspike intervals, most notably the Wilson neuron.

To account for this phase shift and decode a better estimate from the neural spikes, we used our synaptic optimization to find better readout filters for each of the four networks.

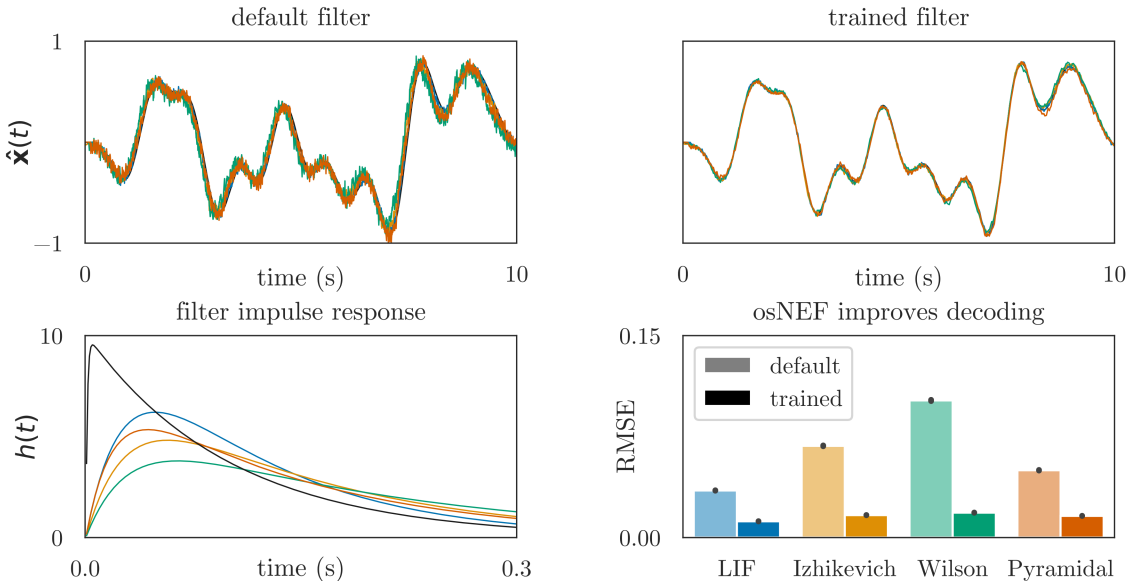

Figure 3.4: Decoding and readout filters. Nonlinear dynamics within complex neuron models leads to systematic decoding error if a default filter is used to smooth the spikes. Optimizing a filter's time constants accounts for this problem and reduces spike noise, leading to highly accurate estimates across all neuron models. The top panels show the target values and the state space estimates, which are decoded from the activities of 100 LIF, Izhikevich, Wilson, or Pyramidal neurons in *pop*; spikes are smoothed using either the default filter (left panel) or the osNEF-trained filter (right panel). The bottom right panel confirms that the RMSE between state space targets and decoded estimates (averaged across 10 simulations with unique inputs) are significantly lower when using trained filters.

The optimization produced filters with longer time constants, which effectively (a) delays the signal and negates the leftward phase shift, and (b) smooths noisy spike trains to recover a more accurate estimate of the input signal. Panel one and two of Fig. 3.4 show the decoded estimate when calculating $\hat{\mathbf{x}}(t)$ using the default filter versus the trained filter. Panels three shows the impulse responses of the trained filters, while panel four compares the RMSE between $\hat{\mathbf{x}}(t)$ and $\mathbf{x}(t)$, when filtering with the default versus trained filters, across 10 input signals. The gains in accuracy with the trailed filter are substantial, and demonstrate that osNEF is capable of accurately decoding state space signals from the activities of nonlinear, adaptive neurons. In the Sec 3.3.2, we report results from networks whose synaptic filters have been trained using the above methods.

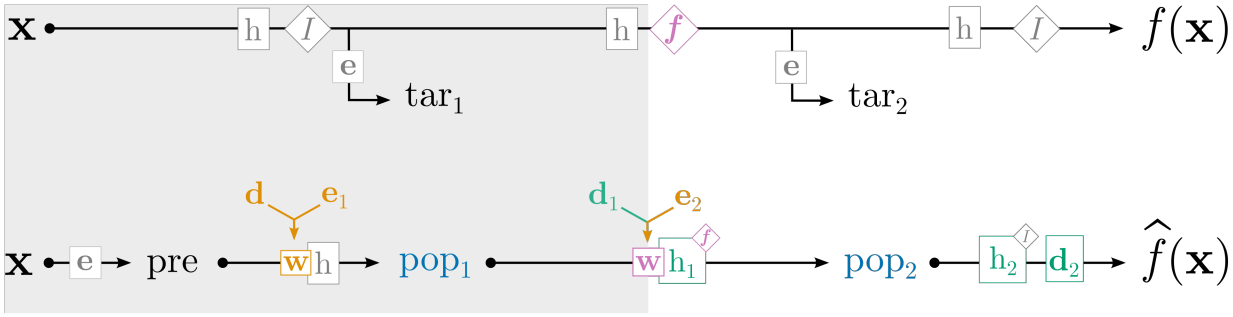

Figure 3.5: Network architecture for computing feedforward functions, including the identity function and multiplication of two inputs. This network extends Fig. 3.2, represented by components with the gray background, by including an additional detailed neuron population pop_2 and the corresponding oracle components. With this architecture, we can compute the feedforward function $f(\mathbf{x})$ on the connection between pop_1 and pop_2 by using osNEF to train the synaptic parameters $\mathbf{d}_1$, $\mathbf{e}_2$, and h_1. As before, coloration indicates ReLU neurons (black) or detailed neurons (blue), synaptic parameters trained by online learning (orange) or offline optimization (green), NEF computations (gray), and finally the new components involved in the calculation of $f(\mathbf{x})$ (pink).

3.3.2 Computation

Having established that encoders, decoders, and filters may be used to translate between the spike space and the state space in a single neural population, we now apply osNEF to train neural networks that compute cognitively-useful functions using the weighted synaptic connections between two (or more) populations of detailed neurons. The simplest network is a communication channel, which simply computes the identity function. The network architecture is shown in Fig. 3.5; the target function is computed between detailed neuron populations pop_1 and pop_2. The target function is

$$f(\mathbf{u}, \mathbf{x}) = \mathbf{u}, \tag{3.4}$$

where $\mathbf{u}(t)$ is the input signal and $\mathbf{x}(t)$ is the state space representation. A functioning communication channel ensures that information can be relayed between components of a cognitive system without significant loss, or be decoded by muscle effectors to implement behavior. Fig. 3.6 shows that osNEF successfully trains encoders, decoders, and time constants that preserve the input signal: the target signal can be reliably decoded from pop_2 with very low error for all four neuron models. While Fig. 3.3 shows that encoder

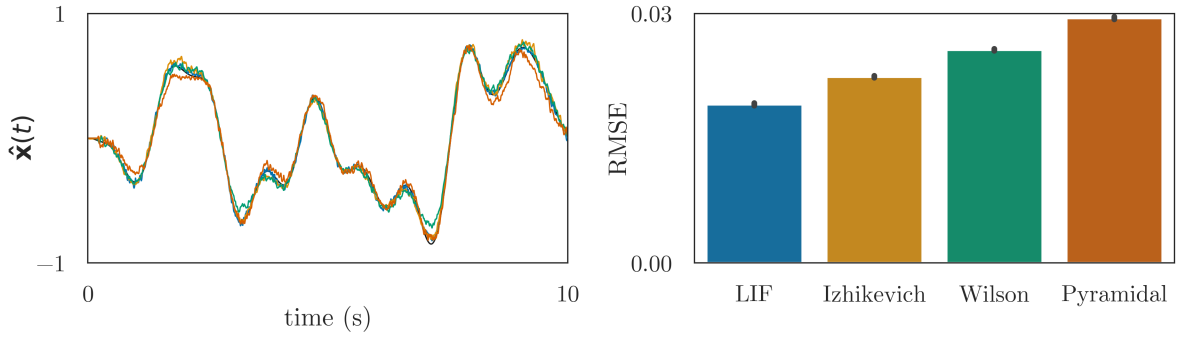

Figure 3.6: Computing the identity function, Eq. 3.4. Using the network architecture in Fig. 3.5, we initialize neural populations pop_1 and pop_2 with 100 detailed neurons, then use osNEF to train encoders, decoders, and synaptic filters. The connection between pop_1 and pop_2 is trained to compute the identity function, such that pop_2 represents the same information as pop_1. The left plot shows the state space target and the decoded estimates from pop_2, and the right plot shows the error (RMSE) between this estimate and the target across 10 simulations with unique input signals.

learning leads to representative spiking activity, and Fig. 3.4 shows that this activity may be decoded to retrieve the signal, Fig. 3.6 shows that encoding and decoding may be combined into a single step via neural connection weights.

We also constructed a network that multiplies two input signals,

$$f(\mathbf{u}, \mathbf{x}) = \mathbf{u}_1 \odot \mathbf{u}_2, \tag{3.5}$$

where $\odot$ is the element-wise product. Multiplication is a computational primitive that may be used in a wide variety of cognitive systems to transform simpler representations into more complex ones. For example, the decision making network we use in Ch. 4 uses multiplication to modulate the rate of evidence accumulation for faster action selection, and the learning network we develop in Ch. 5 uses multiplication to calculate the expected value of an action given a finite time horizon. Fig. 3.7 shows the state estimate decoded from the final population for each of our neuron models. Errors are higher than in Fig. 3.6, as expected given the increased difficulty of the computation. Still, all four neuron models perform well overall.

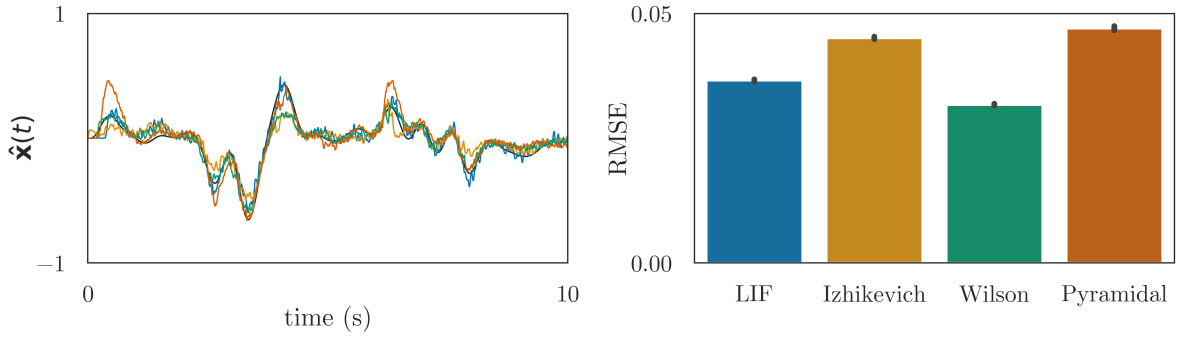

Figure 3.7: Computing the product of two unique input signals, Eq. 3.5, using the network architecture in Fig. 3.2. The connection between pop_1 and pop_2 is trained to multiply two values: pop_1 represents both values, and pop_2 should represent their product. The left plot shows the state space target and the decoded estimates from pop_2, and the right plot shows the error across 10 simulations with unique input signals.

Many biological systems include recurrent connections, allowing the currently represented state to directly affect future states and permitting new classes of network dynamics, including rhythmic oscillations and working memory. Indeed, a network that includes feedforward and feedback components may implement any dynamical system described by Eq. 1.8. The network architecture we use for simulating recurrent networks is shown in Fig. 3.8: feedforward computation of $\dot{\mathbf{x}} = B\mathbf{u}$ occurs on the connection between *pre* and *pop*, while feedback computation of $\dot{\mathbf{x}} = A\mathbf{x}$ occurs on the recurrent connection on *pop*. To train the network, we reuse the architecture in Fig. 3.2, but set pop_1 and pop_2 to be identical populations of detailed neurons computing the target recurrent function $\dot{\mathbf{x}} = A\mathbf{x}$. In doing so, we effectively "unroll" the recurrence, but still use osNEF to train network parameters given a dynamic input signal.

The first recurrent system we investigate is a simple harmonic oscillator, that is, a two-dimensional oscillator with frequency ω.

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \mathbf{x} \tag{3.6}$$

This oscillator is a classic example of central pattern generation: after a brief kick, the system should maintain oscillatory dynamics without external input, which may be used to drive rhythmic behavior in the body, or provide carrier signals that may be modulated by downstream cognitive systems. We arbitrarily chose $\omega = 2\pi$ as our target frequency.

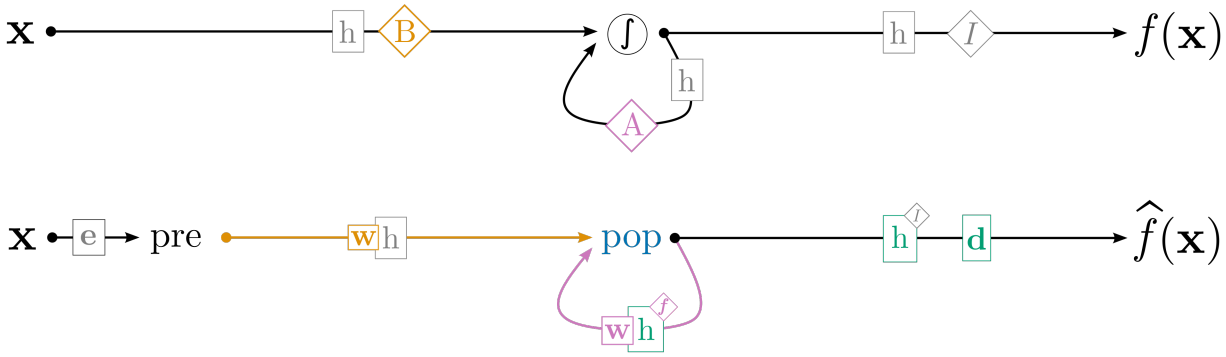

Figure 3.8: Network architecture for recurrent networks. Orange components indicate the feedforward computation of $\dot{\mathbf{x}} = B\mathbf{u}$, and pink components indicate the recurrent computation of $\dot{\mathbf{x}} = A\mathbf{x}$; together they implement Eq. 1.8. While this network is used at test-time, an "unrolled" version similar to Fig. 3.2 is used during training. As such, we remove reference to *tar*, and to the decoders and encoders composing w, in this figure.

Fig. 3.9 shows the network dynamics after a square-wave pulse (0.1s) is used to kick the system. All four neuron types quickly settle into stable harmonic oscillation with frequency approximating the target ω, and these oscillations persist for 100 seconds. Because the networks invariably oscillate at a frequency that differs slightly from ω, a naive calculation of RMSE between $\mathbf{x}$ and the decoded $\hat{\mathbf{x}}$ is a poor metric of the system's stability. To account for this, we report two error values: we first fit a sinusoid of the form $y = a\sin(bt + c) + d$ to $\hat{\mathbf{x}}$, then report (a) the RMSE between this sinusoid and the neural estimate, and (b) the normalized frequency error $(b - \omega)/\omega$.

The second recurrent system we investigate is an integrator, a system that continually adds a feedforward input to a remembered representation of its current value:

$$\dot{\mathbf{x}} = \mathbf{u}. \tag{3.7}$$

This dynamical system requires a neural network to continuously combine feedforward and recurrent signals, an important operation for working memory. We use the network in Fig. 3.10 to realize such a memory, which we refer to as a "gated working memory" [66, 38]. This network was previously discussed in Sec. 1.4.3, and our implementation here, shown in Fig. 3.10, is functionally equivalent to Fig. 1.2. To summarize, the purpose of this network is to (a) read an input value and represent that value using neural activities in a recurrent

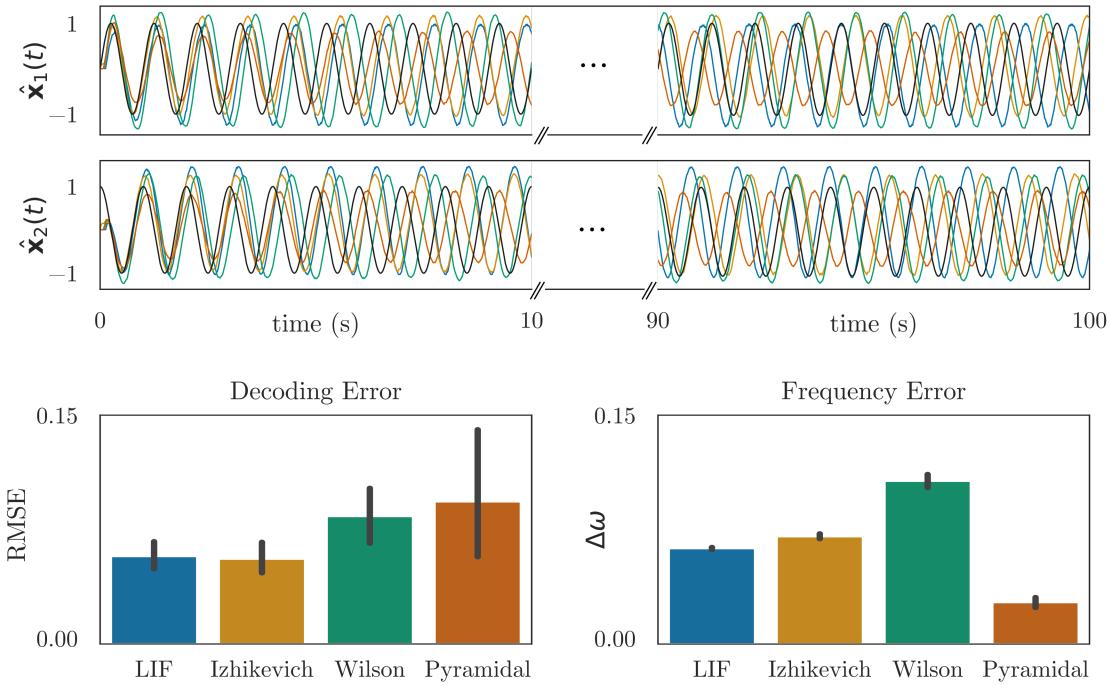

Figure 3.9: Implementing a simple harmonic oscillator (Eq. 3.6) using the recurrent network architecture in Fig. 3.2. The connection between pop_1 and pop_2 is trained to compute Eq. 3.6, then the trained weights and synapses are substituted into a testing network, shown in Fig. 3.8. The top panel shows the targets and estimates from pop_2, with a break in the x-axis to show that oscillations remain stable over 100 seconds. The bottom panel show the mean error between this estimate and a best-fit sinusoid, and the frequency error between this sinusoid and the target frequency.

population, and (b) continue to represent that value once the input has been removed. In our network, the feedforward connection from *pre* to *pop* passes to the memory a two-dimensional value (representing, for example, a visual cue), while the recurrent connection on *pop* maintains the represented cue location once the input has been removed. This is the core "memory" component. The second component is the "difference" component, which ensures that the value represented in *pop* approaches the cue value represented in *pre*. Because the recurrent connection continuously computes the identity function $f(\mathbf{x}) = \mathbf{x}$, maintaining whatever 2D value is currently represented, and the feedforward connection continuously adds the 2D value of the perceived cue to the representation in *pop*, a naive integrator will overshoot the target value if the cue is presented for an extended

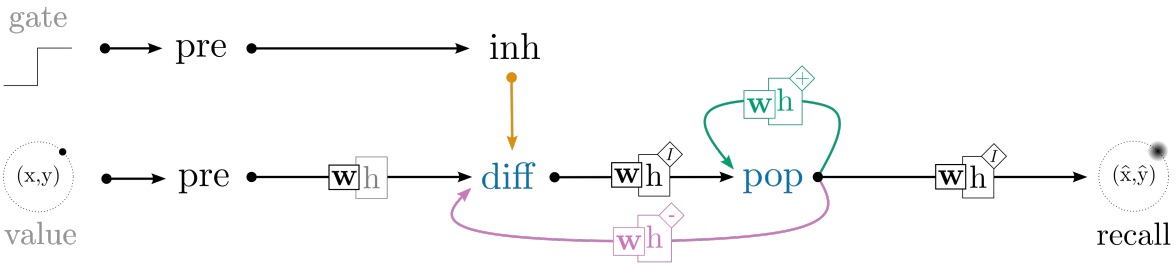

Figure 3.10: Network architecture for the gated working memory. This system loads and stores a two-dimensional value in memory; when the *gate* is closed, the system maintains its current value through recurrent activity, and when the *gate* is open, the system replaces its current representation with the input value. Black text indicates LIF populations while blue text indicates detailed neuron populations. The green connection is trained by osNEF to compute $f(\mathbf{x}) = \mathbf{x}$, while the pink connection is trained to compute $f(\mathbf{x}) = -\mathbf{x}$. The orange connection directly inhibits neurons in *diff* using fixed negative weights.

duration. To prevent this, we add an intermediary population *diff* between *pre* and *pop*. This population receives the feedforward signal from *pre* and transmits feedforward to *pop*, acting as a simple passthrough. However, it also receives a feedback connection from *pop* that computes the negative of the identity function $f(\mathbf{x}) = -\mathbf{x}$. As a result, the value represented in *diff* is equal to the cue's value minus the integrator's estimate; when this estimate becomes equal to the cue's value, *diff* should represent zero, and the representation in *pop* should stabilize at the target value. Finally, the *gate* allows the network to ignore the input and simply retain its current representation. When the visual cue is removed, we treat its absence as a secondary input to the system, which activates a population of neurons *inh* that inhibits *diff*, preventing any further update of the representation in *pop*. To recall an estimate of the remembered cue's location, we simply decode the neural activities in *pop* with the identity function.

Fig. 3.11 shows the DRT performance of our network for each neuron type. In each trial, we present a cue to the network for 1s, then close the gate and record the decoded estimate from *pop*, computing the RMSE between this value and the original input over a 10s delay period. The cues are distributed evenly around the unit circle across our 10 trials, and we report the RMSE averaged across the 10s delay period. Once the input is removed, the value represented in *pop* must be maintained through the recurrent connection; over time, noise inevitably causes this system to drift away from the target value, leading to imperfect recall. However, for most of the presented cues, the network settles on an attractor that is proximate to the target value, leading to reasonable error rates across all neuron types.

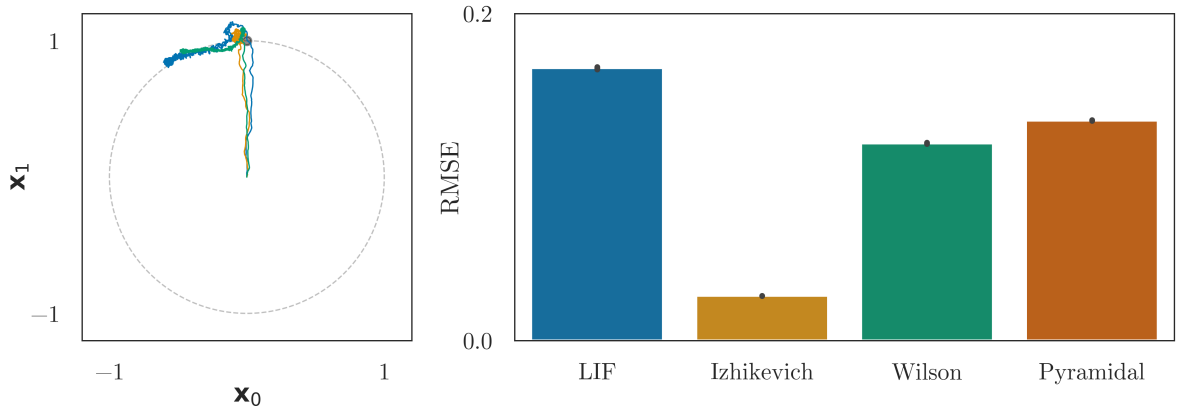

Figure 3.11: Implementing a gated working memory using a combination of feedforward, recurrent, feedback, and inhibitory connections. The left panel shows the estimate decoded from *pop* as a trajectory in (x, y) space: as the cue is presented, the estimate travels from the origin ($t = 0$) to the cue's location, which lies somewhere on the unit circle. At $t = 1$ the cue is removed, and the system must rely on its recurrent dynamics to maintain an estimate of the cue's location. We observe minimal drift in the decoded trajectories for most cues, indicating that our memories are fairly stable over time. The right panel shows the RMSE between the decoded estimate and the cue's true location, averaged over a 10s delay period and over 10 cue locations, for each neuron model.

3.3.3 Application

We now demonstrate that osNEF may be used to build a simple cognitive model out of biologically-detailed components. As above, we use a gated working memory to model a DRT, in which an animal must read information from an external signal, remember that information for a period of time once the signal has been removed, then recall that information. Numerous researchers use DRTs to investigate the neural basis for working memory [45, 208, 61, 86], and neural integrators have been used to model working memory in larger cognitive models that reproduce human behavioral data [23, 67, 55].

We extend the gated working memory network described above by enforcing additional biological constraints and adding an associative memory with winner-take-all (WTA) dynamics to select actions, as shown in Fig. 3.12. As above, we use the Pyramidal cell model proposed by Durstewitz et al., as this cell reconstruction was explicitly designed to simulate pyramidal neurons with delay-period activity in working memory tasks [61]. We also

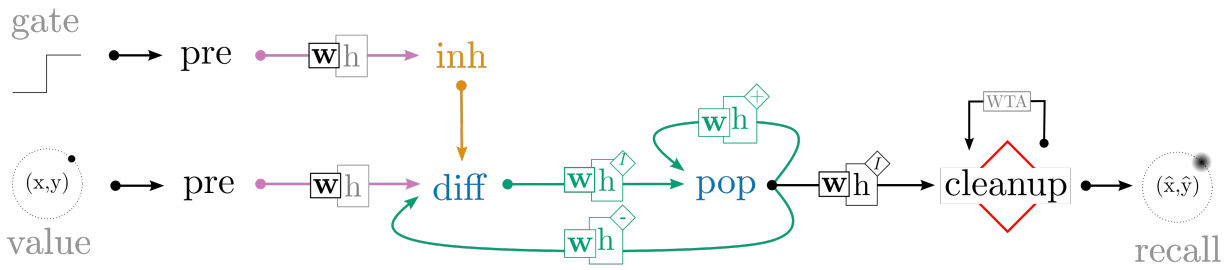

Figure 3.12: Architecture for the biologically-constrained DRT neural network. This network extends Fig. 3.10 by (a) replacing *inh* with a detailed inhibitory interneurons, (b) adding a *cleanup* network that uses WTA competition to find the cue that best resembles the recalled location from *pop* (see [118]), and (c) replacing all connections to/from detailed neurons with conductance-based AMPA, GABA, or (voltage-gated) NMDA synapses. Black populations contain LIF neurons, orange populations are interneurons, and blue populations are pyramidal cells. Pink connections use AMPA synapses, orange connections use GABA synapses, and green connections use NMDA synapses.

use Durstewitz's (a) cellular reconstruction of inhibitory interneurons; (b) conductance-based synapse models for GABA and AMPA; (c) conductance-based, voltage-gated synapse model for NMDA; and (d) biophysical simulation of Dopamine (DA). In these models, DA affects the activation threshold for the persistent Na^+ current, the conductance of the slowly inactivating K^+ current and high-voltage-activated Ca^{2+} current, and the magnitude of NMDA, AMPA, and GABA synaptic conductances. The parameter values were taken directly from the Durstewitz's original source code and were not modified to improve the performance of our model. As above, we use osNEF to train synaptic weights, resulting in dense connectivity that includes both excitatory and inhibitory synapses between pyramidal cells and interneurons. Unlike in the networks above, however, we do *not* use the osNEF to optimize synaptic time constants, since the time constants of AMPA, NMDA, and GABA synapses are fixed to their biological values.

Our cognitive neural network performs a two-dimensional DRT. The network is first presented with an input cue, which represents the (x, y)-location of a target point on a visual screen, for 1s. The cue is removed, followed by a delay period, during which the network continuously reports its remembered estimate of the cue's location. This estimate is sent to an associative memory, which stores the possible true locations of the target; the associative memory compares the current estimate of the cue's location to these targets and outputs the target vector with the greatest similarity, effectively acting as a cleanup operation for the remembered location [220]. We classify a response as "correct" if the

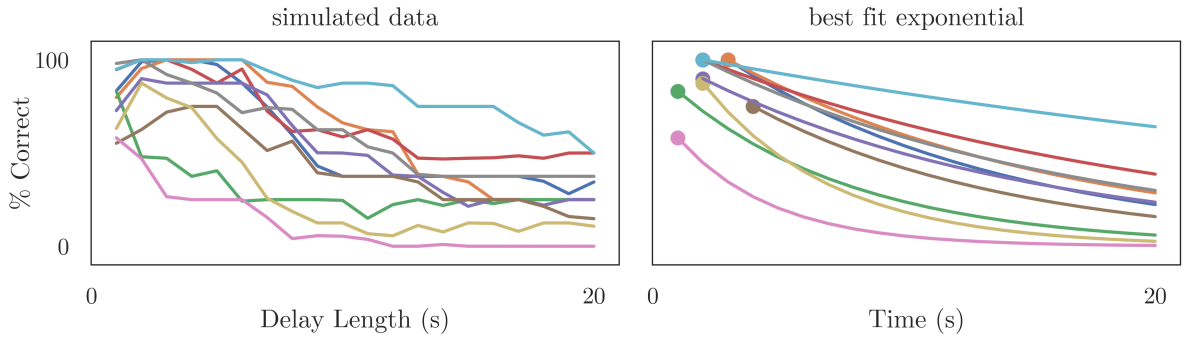

Figure 3.13: Mnemonic performance of the biologically-detailed cognitive network and the best fit exponential forgetting curve. We trained the network show in Fig. 3.12 using osNEF to produce a gated working memory, then plotted error (percentage of correct responses over 8 cue locations) as a function of time. We repeated this training and testing procedure for 10 unique networks, treating each as an individual "participant" performing this task, then fit an exponential function to each network's forgetting curve. From these fitted curves, we obtained parameters for baseline performance and performance half-life, which we compared with the empirical data shown in Fig. 3.14.

output of the cleanup memory falls within a certain (Euclidean) distance of the target cue, and measure the percentage of correct responses as a function of the delay period length. To ensure robust results, we train 10 unique networks, and test each network's recall accuracy for 8 cues with (x, y) locations distributed evenly around the unit circle.

Fig. 3.13 reports the forgetting curves for the biologically-detailed network, plotting the percentage of correct responses given by the model as a function of the delay length. We use use Scipy's `curve_fit` function to estimate parameters for an exponential forgetting curve ($y(t) = B\exp(-t/\tau)$) that fits the simulated data from each network. When preprocessing the data, we observed that the maximum accuracy achieved by the network often occurred 500-2000ms after the cue was removed, presumably as a result of the long NMDA time constants and recurrent dynamics within the network. We set B equal to this accuracy (for each network) and assumed an exponential rate of forgetting beyond this value; we also transformed τ into a "half-life" by multiplying with $\ln 2$. This resulted in best fit parameters B ranging from 58% to 100% (mean 89%, median 95%) and τ_{half} ranging from $2.7 - 28.0$s (mean 9.6s, median 8.6s). The exponential curve is a good fit for the simulated data, a result consistent with numerous working memory studies in animal studies [12, 255].

Our best-fit parameters are consistent with the forgetting parameters reported in a

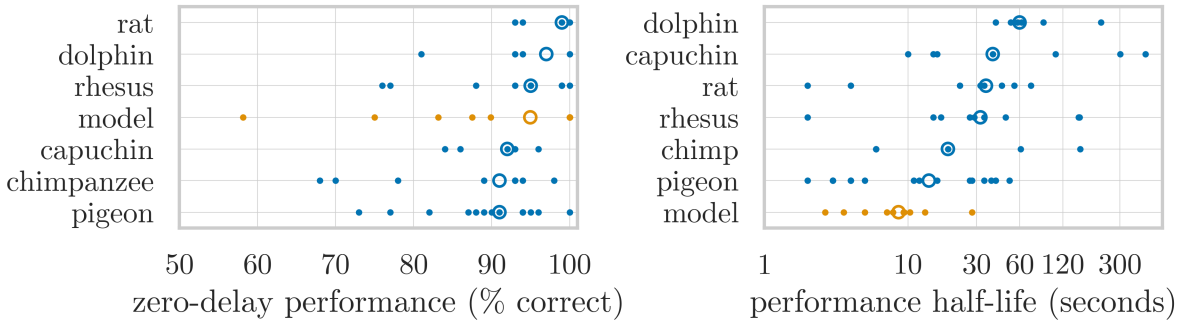

Figure 3.14: Estimated zero-delay performance (left) and performance half-life (right) in DMTST across species. Performance half-life represents the time it takes for an animal's performance to drop halfway between its zero-delay value and chance performance. Open circles are species medians. Empirical data are taken from [143]. There is significant variance in forgetting rates across networks and across individual animals (or experiments) within the empirical data. While the median mnemonic performance of our networks is lower than the median performance of most species in [143], our high-performing networks still outperform a significant number of individual monkeys, rodents, and birds, suggesting that our cognitive networks operate in a biologically-plausible WM regime.

recent meta-analysis of animal mnemonic performance in the delayed match-to-sample (DMTST) task [143], a DRT that assesses numerous aspects of working memory capacity. The resulting dataset includes behavioral data from over 90 experiments, 25 species, and multiple delay intervals. For each species in the dataset, the authors used an exponential curve to quantify the relationship between DMTST performance and delay intervals. Unsurprisingly, the fitted parameters varied significantly both across species and across experiments with the same species, as shown in Fig. 3.14. The baseline performance B (characterized as "zero-delay performance") was consistently high for all species, with median estimates varying between 58% (chickadees) and 99.5% correct (rats) with a grand median of 93% correct. The forgetting time constant τ_{half} differed significantly across species, with median estimates varying between 2.4s (bees) and 71s (dogs), with a grand median of 27s. As shown in the bottom of Fig. 3.14, the performance of our cognitive network falls within these ranges, with a median baseline performance of $B = 95\%$ and a median forgetting time constant of $\tau_{\text{half}} = 8.6$s (comparable to pigeons' $\tau_{\text{half}} \simeq 10$s). These correspondences speak to the cognitive plausibility of our biologically-detailed model, showing that it produces behaviorally-plausible results despite numerous low-level constraints.

3.4 Discussion

Our goal in this chapter has been to develop the "oracle supervised Neural Engineering Framework" (osNEF), a method for training biologically-detailed spiking neural networks to realize dynamical systems that are relevant to cognition. We presented a novel online learning rule for training encoders and decoders to realize a set of target tuning curves in the context of a larger spiking neural network. We also presented an offline optimization procedure for training the time constants of synaptic connections that helped account for nonlinear dynamics within the cell when realizing network-level dynamics. After presenting four neuron models that ranged from the simple and computationally-inexpensive to the complex and biologically-detailed, we showed that osNEF could be used to train neural networks that implement several cognitively-relevant dynamical systems. Specifically, our networks were populated by LIF neurons, Izhikevich neurons, Wilson neurons, or 4-compartment, 6-ion-channel Pyramidal cell reconstructions, and we trained them to compute the identity function, to multiply two inputs, to exhibit simple harmonic oscillation in two dimensions, and to save and load information with a working memory. Finally, we applied these methods to build a simple cognitive system that performs a DRT using biologically-detailed components, including Pyramidal cells, inhibitory interneurons, conductance-based AMPA and GABA synapses, and voltage-gated NMDA synapses. We tested our network's mnemonic performance by measuring the number of correct responses as a function of delay length, then showed that this performance is comparable to animal performance in the DMTST [143]. In this section, we discuss our methods and results in terms of biological plausiblity, cognitive capacity, and usability, compare our methods to similar approaches, and present avenues for future research.

3.4.1 Biological Plausibility

The central motivation of this chapter was to introduce more biological realism into NEF networks without compromising on cognitive capability. Because osNEF operates exclusively in the spike space of neural activity and the state space of dynamical systems, our methods are agnostic about the internal structure of neuron and synapse models. This means osNEF can be applied to many neuron models, ranging from univariate point neurons to multi-compartment models with complex state dynamics, and can use features of neuron synapses to manage intracellular nonlinearities and achieve the desired network-level dynamics. Here, we simulated neuron models with electrophysiologically-plausible internal dynamics, connected them with higher-order synapses (including some whose dynamics couple with intracellular dynamics directly), and showed that they can perform

a variety of cognitive operations. As we discuss below, these biological and functional capabilities are a significant extension of existing methods in computational neuroscience.

Despite these biological extensions, our networks still depart from biological realism in several respects. While our neuron and synapse models themselves conform to biology, the connectivity within the network is less constrained. For instance, neurons in our networks connect to post-synaptic cells with both excitatory and inhibitory synapses, but Dale's Principle suggests that biological neurons exclusively release one type of neurotransmitter [63] (although some neuroscientists have questioned this principle, see [227]). More importantly, the networks presented here do not attempt to reproduce the statistics of neural connectivity between populations, to diversify neuron morphology (beyond small variations in compartmental geometry) or cell type (beyond Pyramidal cells and inhibitory interneurons), or to match other network-level anatomical details. These are important biological features that have been the focus of projects that anatomically reconstruct the brain with high fidelity [159]. Functionally, these features may affect a network's ability to compute specific functions or perform particular cognitive operations, either by adding subcellular information processing, or by constraining network-level dynamics. Fortunately, osNEF does not prohibit the inclusion of such feature; future work should investigate whether imposing these constraints poses a problem for our methods.

Another questionably-realistic aspect of our method is the online learning rule for updating synaptic weights, Eq. 3.2. Most terms in this equation require only local information, such as presynaptic activity, postsynaptic activity, and a state space error signal, and so are plausibly-accessible by individual neurons for learning. However, one critical component in osNEF is the "oracle", a population of neurons built using standard NEF tools that generates the desired neural activities. While it is possible that the brain contains "teacher" populations that supervise the weight updates within "student" populations, we are not aware of any empirical evidence that directly supports this mechanism. We must therefore treat osNEF's encoder learning as biologically unfounded: it is a theoretical tool for constructing networks, not a biological hypothesis.

Finally, the osNEF procedure for optimizing synaptic time constants is not intended to mimic a biological mechanism, but simply to find synapses that realize particular network-level dynamics given complex neuron-level dynamics. It is possible that future neuroscience research will reveal some optimization process in the brain that effectively selects which neurotransmitters, receptors, or network structures should be used to control system dynamics. Studies of highly-structured neural circuits, such as the granule-Golgi cells in the cerebellum [121], indicate that plasticity may interact with synaptic and intrinsic cellular responses (such as rebound firing [206] and inhibitory inputs [201]) to alter the temporal properties of the network. While the osNEF optimization via Hyperopt would not repro-

duce the processes of such an optimization, it might reproduce the results, with respect to the balance of time constants observed in the final network.

3.4.2 Cognitive Capacity

To create cognitive models out of biologically-detailed neural networks, we trained our networks to implement linear dynamical systems described by control theory (Eq. 1.8). We first showed that osNEF could produce neurons whose tuning curves were an effective function basis despite neural adaptation, spike noise, and the like. We then trained networks to perform a variety of operations common to cognitive systems, and finally combined several such operations together into a larger cognitive network. This network successfully performed a DRT, exhibiting an exponential forgetting curve that resembles animal forgetting curves in the DMTST [143]. These mnemonic capacities are impressive given that (a) our network uses only hundreds of neurons, which is orders of magnitude fewer neurons than are active in behaving animals, and (b) our DRT task is significantly harder than the DMTST task used in the meta-analysis (eight cues versus two). In conjunction with the broader successes of the NEF and SPA, we expect that osNEF can be used to train more sophisticated cognitive networks built from biologically-detailed components.

However, osNEF tools are limited in several respects. Networks trained using the standard NEF methods typically have less error than the results we reported in Sec. 3.3.2. This is due, in large part, to the lack of biological constraints imposed by the default parameters implementing NEF networks in Nengo: such networks use point neurons and lowpass current-based synapses, leading to fewer cellular nonlinearities, and have firing rates approximately ten times higher than our simulations. Normal NEF methods also realize tuning curve distributions that provide a superior function basis. In standard NEF networks, neurons are biased by directly injecting a current into the "soma", allowing for precise control over the conditions under which a neuron will begin firing. Before the network is simulated, Nengo optimizes these bias currents and the encoding vectors such that neurons exhibit a wide range of x- and y-intercepts. osNEF does not use current injection to bias neurons: instead, it trains synaptic weights to achieve the target tuning curve distribution, making postsynaptic activities more dependent on noisy presynaptic spikes. In recurrent networks where feedforward input was selectively removed (the oscillator and the gated working memory), we needed to introduce and train a dedicated *bias* population to stabilize recurrent activity. Finally, not all target tuning curves can be realized by a given neuron type: osNEF may fail if the targets have physiologically-implausible response curves or if a detailed neuron is tightly constrained by its morphology. Because osNEF is a

black-box approach, the only way to know whether a neuron may be trained to reproduce a target tuning curve is through trial and error.

3.4.3 Usability

In developing osNEF, we strove to make our methods both broadly applicable and easy to learn, with the hopes that other researchers will apply them in their own projects. We used the Nengo ecosystem [20] because it is a scalable and flexible neuron simulator that has been used to build and train numerous neural and cognitive models, including some of the world's largest functional brain models [66, 38]. To support a wide range of biological mechanisms, users may simulate neural and synaptic dynamics in either Python or NEURON. Python is appropriate for simulating simple models: this language has intuitive syntax and can be used alongside NumPy for efficient matrix operations [96]. NEURON is appropriate for simulating complex neuron models: this language is designed to efficiently simulate detailed cellular and synaptic mechanisms. Libraries of neural reconstructions written in NEURON are widely available from online repositories like the Allen Cell Types Database [228] and the Neocortical Microcircuit Collaboration Portal [186], and our interface allows researchers to connect these models into the Nengo ecosystem for functional modelling, as we did with the Durstewitz model [61].

That said, training neural networks with osNEF requires significantly more effort and simulation time than standard Nengo networks. Users must choose appropriate training signals, build a parallel oracle network to generate the target activities, and simulate the network over time to engage online learning. The challenges posed by different neuron types and target dynamics makes automating this process difficult, so the user must complete these steps manually. However, we believe that the broad applicability of osNEF justify these challenges for the motivated researcher.

3.4.4 Comparison to Other Methods

The NEF is the theoretical core of osNEF: from it, we borrow (a) the distinction between spike space and state space, (b) the decomposition of weights into encoders and decoders, and (c) the use of control theory to specify target dynamics. We also use a number of standard NEF tools, including (d) the PES learning rule, and the specification of target networks via (e) least-squares optimization of decoders and (f) distribution of encoders, gains, and biases. osNEF extends the NEF by (1) redefining encoders as a tensor over presynapic neurons, postsynaptic neurons, and state space dimensions, (2) introducing an

online learning rule to update encoders and decoders based on state space error, spike space supervision, and Hebbian activity, and (3) optimizing the time constants of synapses to realize network-level dynamics while accounting for adaptive neuron-level dynamics. Many of these techniques bear a resemblance to other research in computational neuroscience. For instance, a central theme of methods like FORCE learning [229] and efficient balanced networks (EBN, [28]) is to describe cognitive algorithms in terms of the dynamics of a latent state variable $\mathbf{x}(t)$ represented by neural activity, then train neural connection weights such that the network behaves like a target dynamical system. However, the techniques for constructing and training networks varies significantly between these paradigms: osNEF borrowed several of these techniques in extending the NEF.

In the full-FORCE method [49], the recurrent activities of a neural network are trained with the aid of a parallel target-generating network that is driven by the desired output of the system. full-FORCE target networks have random internal connectivity: when driven by the target dynamics, such networks produce activities that include both a chaotic component (from the random recurrent connectivity) and a desired component (from the driving input). Such activities, the authors hypothesize, is a suitable basis for realizing non-trivial dynamics, especially when combined with an optimized readout filter. In their paper, the authors show that a recursive least squares optimization process, which compares the target activities with the activities of the task-performing network, may be used to train recurrent weights in the later and reproduce a wide variety of dynamics. Our oracle populations are also driven to exhibit the target dynamics and used to learn recurrent weights. However, our oracle does not rely on random connectivity: we use the NEF to specify weights that guarantee that *tar* will exhibit spike variability and the target dynamics. Our approach also simplifies the relationship between the task-performing and target networks: they are both driven by the same external inputs, rely solely on recurrent activity to generate the target dynamics, and should exhibit the same activities. This eliminates a great deal of parameter tuning required by full-FORCE, and provides a clear conceptual pictures of how the target network supervises the task network. Furthermore, osNEF uses a local, online, error driven learning rule, while the RLS approach used in full-FORCE is a global, iterative update that is very unlikely to be implemented by brains.

A recent extension of EBNs to nonlinear adaptive control theory [3] also bears many similarities to osNEF. In this paper, the authors realize nonlinear dynamics in a recurrently-connected population of spiking LIF neurons using a state space teacher and an online learning rule. Many mathematical similarities exist between this approach and the NEF, especially with regards to the PES rule. The authors convincingly demonstrate their ability to learn nonlinear dynamics, including a bistable attractor and rhythmic walking motions derived from motion capture data, and their networks are similarly constrained to low

firing rates, small numbers of neurons, and irregular spiking. However, where their work (and EBNs in general) focuses on using fast inhibitory connections to create an efficient, balanced coding scheme, our work focuses on implementing all of the required components in biological detail. While the authors describe how biological components could be used to implement their methods (AMPA/GABA-A synapses for fast connections, NMDA/GABA-B synapses for slow connections, and nonlinear dendrites for the basis functions), dealing with the nonlinear, non-instantaneous dynamics imposed by compartmental neurons and conductance-based synapses often requires significant theoretical extensions, even when the underlying framework can already model nonlinear dynamical systems.

Recent papers using both FORCE [176] and EBN [209] have implemented linear (and sometimes nonlinear) dynamics in neural networks populated with biologically-detailed neurons. If simulation of functional networks with complex neurons is already possible with other techniques, why use osNEF? Broadly speaking, there are important theoretical differences between the underlying frameworks, FORCE, EBN, and NEF. Even if we assume that all three methods solve similar problems with similar performance, there is significant value in developing and presenting an NEF-based method that is comparable to FORCE- and EBN-based methods. The NEF modeling community is quite large, so validating a method for training biologically-detailed functional models within this framework is important, independent of similar successes with FORCE and EBN. Furthermore, given the complexity of training biologically-detailed networks, we believe that the research community will benefit from the existence of multiple methods that tackle this problem in different ways. Future work should compare these methods, identify their relative strengths and weaknesses, and develop new methods that build upon their successes.

There are also significant differences between these three methods with respect to biological plausibility and computational capacity. In the FORCE paper [176], the authors simulate spiking LIF, Theta, and Izhikevich neurons, showing that they can produce a number of dynamical systems, including oscillators, chaotic systems, songbird calls, and episodic memories. These methods use a variety of current-based synapse models (exponential, double exponential, and alpha) and respect Dale's principle. In the EBN paper [209], the authors simulate point neurons that include Hodgkin-Huxley-type ionic currents, showing that they can produce several types of one-dimensional integrators that statistically reproduce empirical patterns of neural activity. Their methods also use double-exponential synapses, but do not respect Dale's principle. In this chapter, we simulate three point neurons and one pyramidal cell reconstruction with four compartments and six ionic currents. We show that our trained networks reproduce four linear dynamical systems (feedforward and recurrent networks) and perform a cognitive task at performance levels comparable to simple animals. Our methods use both current-based double-exponential synapses and

conductance-based, voltage-gated NMDA synapses; we do not respect Dale's principle. We feel confident in claiming that we have (a) simulated neurons with significantly more biological realism than the FORCE and EBN papers, except with respect to Dale's principle (but see below), and (b) demonstrated that our trained networks perform a wider variety of computations than [209] and a comparable number to [176] (excepting nonlinear systems).

Other researchers have extended the NEF with the goal of increasing the framework's biological plausibility. Stöckel [223] has developed numerous methods for training neural networks to realize computationally-useful dynamical systems. One such method simulates networks of multi-compartment LIF neurons in which synaptic connections are decomposed into excitatory and inhibitory components, each characterized by an appropriate equilibrium potential. By selectively placing these synapses on different compartments and optimizing excitatory and inhibitory weights, Stöckel shows that these dendrites can effectively compute functions like four-quadrant multiplication [225]. Other work applies this E-I optimizer to a functional model of eyeblink conditioning in the cerebellum's granule–Golgi microcircuit, demonstrating that anatomically-detailed spatial connectivity can be profitably incorporated into NEF models while respecting Dale's principle [226]. These methods nicely complement the osNEF: where Stöckel focuses on adding biological details to the connections between groups of neurons (managing E-I balance, targeting dendrites, and reproducing spatial structure), we focus on adding biological details to the internal dynamics of the underlying components (the neuron and synapse models). Future work that combines these techniques would greatly enhance the biological plausibility of NEF networks, which have previously been criticized for lacking particular biological features.

The use of high-order synapses to control network-level dynamics has also been explored using the NEF and EBN. Voelker has derived a method for computing the parameters of high-order synapses that, when simulated in networks of simple neurons, preserves state space dynamics in the network. These NEF-style techniques can be used to construct models that encode rolling windows of input history, and the resulting neural activities closely resemble the mnemonic responses of time cells in the cortex [249]. In contrast to osNEF, these analytical techniques guarantee a solution, but rely on certain assumptions that are violated once sufficient biological detail is included: specifically, if the dynamics internal to the neuron model dominate the synaptic dynamics, or if the synaptic dynamics are coupled to the cellular dynamics. EBN has also used synaptic dynamics to account for the nonlinear dynamics of complex neurons, but relies on a significantly different optimization processes for discovering those synaptic dynamics. In [209], the authors use a form of system identification that (a) drives the neuron model with a specific random process, (b) analyses the resulting voltage traces when the neuron spikes, (c) calculates the average action potential waveform, (d) takes its temporal derivative, and (e) convolves it with an

exponential function. In our networks, where cellular adaptivity is a major force, where voltage-gated synapses depend on intracellular activity, and where the dynamics of distinct geometric compartments are coupled, applying these types of analytical techniques become difficult or impossible. Still, these methods demonstrate the utility of optimizing synaptic time constants for network-level dynamics.

3.5 Conclusion

Functional capacity and biological plausibility are two important criteria for neural network models that seek to clarify the relationship between low-level mechanisms and high-level behavior. In this chapter, we developed osNEF, a method for training biologically-detailed spiking neural networks to realize cognitively-relevant dynamics. osNEF uses an online learning rule that (1) combines insights from several theoretical frameworks, (2) includes error-driven, supervised, and Hebbian components, and (3) can be applied to a wide variety of neuron and synapse models. We demonstrated the utility of osNEF by (a) showing that the neural activities of a trained network form an appropriate function basis for dynamic computation, (b) building several functional networks that perform cognitively-useful operations with high accuracy, and (c) combining these operations into a larger cognitive network that performs a simple working memory task. This cognitive network is built from numerous biologically-detailed components, including Pyramidal cells, inhibitory interneurons, conductance-based GABA synapses, and voltage-gated NMDA synapses, and performs a delayed response task with mnemonic performance comparable to simple animals. We concluded by discussing the biological realism of osNEF, assessing its cognitive capacity and usability, and comparing it to similar methods in the literature.

The methods presented in this chapter have several applications to learning and decision making in social contexts. Functional components like working memory are essential for cognition in general, and for the social tasks that we investigate in Chs. 4-5 in particular. Here, we showed how to construct a biologically-realistic working memory network that can, in conjunction with rudimentary networks for perception and action selection, perform a simple mnemonic task. Moreover, future work may apply osNEF to study many other aspects of social cognition. For example, we could extend the biological realism and cognitive flexibility of our AMY model from Ch. 2 by simulating (a) pyramidal neurons and interneurons, (b) microcircuits with exclusively excitatory or inhibitory connections, and (c) neurotransmitters like DA, 5HT, and NE. Our simulations in this chapter show that these extensions are feasible: our pyramidal and interneuron networks can compute complex functions, our learning rules can be adapted to enforce Dale's principle, and our

neuron models include enough detail to simulate the electrophysiological effects of neuromodulators. Unfortunately, time constraints kept us from pursuing these research directions. Future work should use osNEF to test specific hypotheses related to pharmacology, neurological disorders, and other phenomenon where biology directly affects cognitive abilities.

Chapter 4

Value Accumulation and the Speed-Accuracy Tradeoff

Author's Note: some of the of content in this chapter was previously published as a journal article in the Proceedings of the 42nd Annual Conference of the Cognitive Science Society [55]. Code is available on GitHub.

4.1 Introduction

In previous chapters, we focused on models of associative learning and working memory, emphazising biological plausibility in the form of detailed neuroanatomy and electrophysiology. These components play an important role in many cognitive operations related to learning and decision making. In the next two chapters, we shift our focus towards higher-level cognition by modelling how these components work together to perform complex cognitive tasks. In particular, our goal is to build neural models that realize the value based framework, as laid out in Ch. 1. In this chapter, we investigate the process of value integration and decision making, and in Ch. 5, we investigate value learning based on external feedback.

In dynamical systems like the brain, evaluations and decisions are not made instantaneously: value estimates change as evidence is gathered, and the optimal choice is constantly reevaluated as the brain processes external data through various internal models. In many contexts, agents face a decision making (DM) tradeoff between speed and accuracy: agents that take more time to gather and process information will usually make

better decisions, but agents that take less time to decide will respond more quickly. The speed-accuracy tradeoff (SAT) is prevalent in both natural systems and human societies. Take, for instance, the quasi-social task of choosing a mate. An individual who spends significant time with a potential mate, interacting with them in many different situations, will get to know them well and can make an informed decision about their compatibility. On the other hand, an individual who only spends a short time with a potential mate might make a decision based on scant evidence, but can use their extra time to investigate other potential mates, increasing their chances of finding someone who is highly compatible.

Many natural and artificial DM tasks feature some form of the SAT, so it is not surprising that the brain's DM systems have evolved to accommodate this tradeoff. Neural and behavioral results indicate that individual and contextual differences, especially those related to urgency and uncertainty, strongly influence how people approach the SAT when performing cognitive tasks. Urgency influences DM if an agent is rewarded for acting quickly, which can occur if the magnitude of the reward depends upon the decision time, or if multiple actions (and associated rewards) are allowed within a fixed time window. Uncertainty influences DM when evidence is noisy or is gathered incrementally, or when prior knowledge (accurate or inaccurate) influences the evaluation process. The cognitive mechanisms that animals use to manage urgency and uncertainty are critically important for understanding DM.

In this chapter, we present a model of dynamic value accumulation and action selection. The model gathers information from the environment by sequentially sampling cues and tracks the accumulated value for two candidate actions in a working memory. It monitors the accumulated evidence and the passage of time, which jointly control a gate that inhibits action selection until the model's decision criteria are met. Once this occurs, winner-take-all (WTA) dynamics selects one action to be implemented, and this choice becomes the model's output. Our network includes many mechanisms that have been theorized to contribute to the dynamic evaluation, accumulation, and selection of values. To investigate how these mechanisms influence the SAT, we vary model parameters to create a population of unique agents, then have each agent perform a DM task with sequential sampling. We measure each agent's accuracy and decision time and analyze their relationship, both for individual agents and across the population. Finally, we compare these data with an empirical dataset from humans performing an identical experiment, discuss which cognitive mechanisms best explain the SAT in the empirical and simulated data, and mention future work that might resolve open questions from our experiments.

4.2 Background

4.2.1 Cognitive Tasks

Decision making is an extensive field in cognitive science, and DM tasks are quite diverse. Tasks used to investigate the SAT fall into two broad categories: perceptual discrimination tasks, and sequential sampling tasks. In perceptual tasks, participants are presented with a sensory input and must classify the input as quickly as possible. In random dot motion tasks, for instance, participants view a movie in which many dots move in various directions, and must identify the predominant direction of movement. In other discrimination tasks, participants view a static image for a short duration, and must classify a target item despite the presence of distractor items. Behavioral responses in perceptual discrimination tasks are usually indicated by button presses or shifting gaze to a specific location. Speed is assessed by measuring the time between stimulus presentation and choice selection, while accuracy is measured as the mean number of correct responses across many trials. Task difficulty is controlled by varying the number of distractor stimuli.

In sequential sampling tasks, participants view a sequence of stimuli that give partial information about the value of various choices. Typically, after each item in the sequence, participants may choose to make a decision, or continue sampling evidence from the sequence to acquire more information. For example, participants might view a screen which sequentially reveals information about the value of two items, and be instructed to select the item with greater overall value. In these tasks, accurate value estimates can only be achieved by taking multiple samples, but doing so requires more time. As with perceptual tasks, behavioral responses are often indicated by button presses, and accuracy is measured by averaging the number of correct responses over many trials. In sequential sampling tasks, the correct response is determined by the underlying probability distribution from which the samples are drawn: the item with the greater chance of generating positive value is the correct choice. Response times may be measured in real time, assuming a regular schedule for presenting samples from the sequence, or simply by counting the number of samples that were requested before a decision was made. Task difficulty is controlled by manipulating the underlying probability distribution, for instance by varying how frequently each item increases in value.

There are important differences between these two types of tasks with respect to understanding the cognitive mechanisms governing the SAT. Perceptual discrimination tasks have a longer history: the SAT was first measured and characterized using perceptual tasks, and participant behaviors and neural activities are readily reproducible. However, some researchers have questioned the extent to which these tasks actually probe the types

of complex, cognitive DM that we seek to understand [73]. While perceptual discrimination does involve the accumulation of evidence for choice alternatives [242, 119, 52], and may be flexibly modulated by signals indicating the quality of evidence or the need for urgency [257], DM in these tasks does not require internal deliberation, risk assessment, or the voluntary seeking of additional information, three features that define high-level DM. Furthermore, the SAT is not always an inherent feature of perceptual discrimination tasks: this tradeoff must often be highlighted by researchers when designing the task or explaining it to participants, for instance by giving explicit verbal instructions to favor speed or accuracy, by creating a payoff structure that explicitly rewards one or the other, or by imposing external deadlines for responses [102]. Finally, differences in speed and accuracy in perceptual tasks may be governed by perceptual abilities and by the allocation of attention, rather than by exerting metacognitive control over DM systems [73]. In contrast, sequential sampling tasks, such as those presented in [85, 149], naturally address many of these concerns. Even without explicit verbal instructions or payoff structures, participants in sampling tasks conceptually understand they are faced with a SAT, and attempt to behave accordingly, with varying levels of success [73]. These tasks involve a repeated, active decision to continue gathering information, or to make a choice based on the accumulated evidence: this engages deliberative DM systems, where urgency and uncertainty estimates may dynamically interact to control cognition and behavior. Finally, sequential sampling tasks are largely divorced from sensory processing, meaning that individual differences in perceptual abilities do not confound the analysis of the SAT. For these reasons, we choose to investigate DM and the SAT using sequential sampling tasks.

4.2.2 Neuroanatomy

The functional neuroanatomy of value based accumulation and DM is largely consistent with the value base framework presented in Sec. 1.3. In many DM tasks, value estimation is not necessary, as the values associated with each candidate action is either given directly (sampling tasks) or inferred by perceptual hierarchies (discrimination tasks). Once values have been represented in the brain, these values must be integrated into the current overall value estimate for each action. Extensive evidence supports the idea that this occurs via neural accumulators distributed throughout the brain; this notion is consistent with the value based framework, especially with the hypothesis that accumulators for different modalities are realized in domain-specific WM buffers throughout cortex (see Sec. 1.3.7). In perceptual tasks, evidence accumulation occurs primarily in visual areas like the medial temporal area (MT), the fusiform gyrus (for facial recognition), or parietal areas like the intraparietal sulcus (IPS) [119, 257, 91]. In sequential sampling tasks, evidence is abstract,

and is processed in the cortial areas identified in Secs. 1.3.2-1.3.3, such as vmPFC, dlPFC, OFC, and ACC [52, 85, 172, 149, 91]. Interestingly, value accumulation is also apparent in parts of motor cortex, such as the supplementary motor area (SMA) and the premotor cortex (PMC), for both perceptual and sampling tasks [119, 91], indicating that internal representations of the action plans themselves might accumulate evidence.

Various brain areas have been implicated in flexibly controlling the SAT based on urgency, uncertainty, and prior knowledge. Areas like dlPFC may influence the weighting of incoming information, based on the reliability of evidence or the urgency of acquiring more information [52]. Various areas of the BG, including the caudate nucleus, substantia nigra pars compacta (SNc), and the subthalamic nucleus (STN), are also thought to modulate urgency information, effectively shifting decision thresholds through projections to cortex [257, 91]. The involvement of these areas in thresholding decisions is further supported by data from humans with parkinson's, schizophrenia, and autism: disrupted SNc and STN activity in these disorders correlates with higher decision thresholds and slower evidence accumulation [91]. Other researchers have argued that changes in decision threshold and the rate of evidence accumulation are realized directly in cortical areas, where sensory processing and WM evaluate and integrate evidence [96]. Finally, as discussed in Sec. 1.3.6, action selection involves WTA competition, either through inhibitory connections within the cortex (particularly motor regions) or via mutual connections between cortex and BG.

4.2.3 Theoretical Models

Many mathematical and computational models have explored the neural and cognitive basis of DM. The most influential model in this field is the drift diffusion model (DDM), in which a decision variable drifts between two decision thresholds, and a choice is made when the variable crosses one threshold [190]. In the DDM, the drift of the decision variable is subject to random noise, leading to variance in reaction times (RTs) across trials, and has a slope (drift rate) that varies between individuals or is based on the context (e.g., stronger or more credible evidence leads to higher drift rates). The three key parameters of the DDM are thus the decision threshold, the drift rate, and the "non-decision time", which determines the length of time between the start of the simulation and the onset of accumulation. The DDM has been very successful in modelling animal behavior across many domains: DDM parameters can be fit to individuals such that models reproduce individual RTs and accuracy distributions; and DDM variables have been correlated with neural activity in brains areas associated with DM [191]. Many of the results discussed in the previous section, which associate parameters like threshold and evidence accumulation rate with particular brain regions, are made in reference to the DDM. However, the DDM is a purely

mathematical model and is agnostic about neural implementation, so any correspondence between DDM variables and neural activity will necessarily be a high-level comparison.

Numerous extensions of the DDM have sought to increase its cognitive and neural realism, for instance by capturing the effects of urgency and uncertainty. Some DDM models assume that decision thresholds vary over time, shrinking as the pressure to make a decision steadily increases: the existence of a ramping urgency signal is consistent with behavioral data [51] and neural data [41], though the strength of this effect has been called into question [71]. In other models, the drift rate is the target of urgency modulation: as time pressures increase, drift rates also increase, leading to faster decisions [174]. Finally, decision conflict, or the uncertainty about choice values, may also stimulate changes in DDM parameters, speeding or slowing decisions based on past performance or current feelings of confidence [138]. Several computational models have already realized the DDM in neural networks [218, 78], and have investigated the role of urgency and uncertainty in flexibly controlling the SAT. However, neural implementations are less common than pure mathematical models.

Other theoretical models provide an alternative account of DM that address some of the weaknesses of the DDM. In its original form, the DDM includes a single decision variable that may (noisily) drift towards one of two thresholds; this formalization limits DDMs to two-choice tasks (although DDM models like [202] have extended this towards multiple choices). In so called "race" models of DM, each choice alternative has a separate decision variable, and each variable independently accumulates towards its own decision threshold: whichever choice crosses threshold first wins. Race models include the linear ballistic accumulator (LBA, [31]) and the independent accumulator (IA, [89]) models. Given that these models (a) naturally apply to multiple-choice DM, (b) can be extended to model urgency and uncertainty, and (c) reproduce empirical data, we use these frameworks as the foundation for our model.

Another framework that has been successful in explaining DM in the brain is Bayesian inference. Rather than explicitly track value estimates for candidate actions or decision variables, this framework assumes that the brain maintains beliefs about the external world using probability distributions, then uses those beliefs to calculate the expected value of each candidate action. As an agent samples information from the environment, takes actions, and observes the outcome, it uses Bayesian inference (sometimes in conjunction with Reinforcement Learning) to update its beliefs. Previous work has shown that computational models of probabilistic inference and belief updating may explain many aspects of human DM under uncertainty, both in perceptual tasks [187, 108] and social tasks [124, 125].

Bayesian models of DM have several advantages over DDM and race models. First, by maintaining a belief distribution, Bayesian agents can incorporate new information and update their beliefs in an efficient and distributed manner: that is, these agents may effectively update their beliefs about the value of many candidate actions simultaneously. Second, Bayesian agents' rich representations about the external world may serve as the basis for a wider variety of cognitive operations. For instance, Bayesian models can explicitly calculate the value gained by sampling new information; this calculation lets the model learn (via external feedback) the expected value of a new piece of information, and make a decision when the expected value of choice exceeds the expected value of sampling [108]. Another example is that Bayesian models can calculate uncertainty using formulas that account for the entire history of sampling, rather than the current evidence estimate alone: previous work has shown that these uncertainty calculations may explain counter-intuitive findings in the behavioral literature, such as the well-documented tendency for people to express high perceptual confidence in situations where they actually have low accuracy [123]. In addition to their added cognitive flexibility, empirical evidence supports the idea that brains may perform Bayesian inference, and the above computational models have shown that they can be realized in neural networks (with varying degrees of biological plausibility) [187].

While Bayesian inference is a promising approach to studying DM, its assumptions and mechanics differ from those involved in traditional value based decision making, as we have laid out in this thesis. For example, Bayesian models require that the modeller specify the mathematical form of the belief distribution used by an agent, which arguably introduces an implausible degree of "expert knowledge" into the model. In contrast, DDM and race models do not assume the model has any prior knowledge about the underlying world state. Furthermore, Bayesian updates are more computationally-intensive than value updates, and require different neural architectures and learning rules. The cognitive components required for such updates are currently under development in the NEF [210, 80], whereas the cognitive components for evidence accumulation are already well-established in the Semantic Pointer Architecture [64]. For these reasons, we relegate an investigation of decision making through Bayesian inference to future work.

4.3 Model

To investigate DM and the SAT, we adopt the sequential sampling task described in [73]. In this task, participants are presented with sequential evidence about the changing financial value of two hypothetical stocks, and are asked to select the stock with greater

value. Participants view a screen which displays information about stock A on the left and stock B on the right. Every 500ms, the screen displays new evidence about one stock, indicating whether it increased or decreased in value. Samples alternate between stocks A and B. Behind the scenes, stocks A and B are each assigned a probability of increasing in value. These probabilities range from 0.1 to 0.9, and the difference in probabilities between A and B is fixed to one of three values, $\Delta P = [0.1, 0.2, 0.4]$. Larger ΔP corresponds to an easier decision task, which should lead to faster and more accurate DM. When participants are ready to make a decision, they press the space bar to stop sampling, then press the F or J key to select stock A or B. Once participants complete a single trial, the next trial begins immediately. Participants have a fixed time period to complete the task, and the amount of remaining time is displayed on the screen. Because participants are rewarded based on the number of correct stocks they chose (+1 point for a correct choice, -1 for an incorrect choice), and because the number of trials they perform is limited only by their speed (within the fixed time period of the experiment), the task rewards both speed and accuracy, effectively establishing the SAT. In their empirical study, [73] used post-experiment surveys to confirm that participants understood this tradeoff; they also performed various manipulations of the experiment (e.g., changing reward structure, instructions, or feedback), which (slightly) altered how quickly participants performed the task.

Our spiking neural model contains four core components, summarized in Fig. 4.1. The *value* population receives external inputs: an input of +1 indicates the stock increased in value, and an input of -1 indicates it decreased. These values are sent to an *accumulator* population, whose recurrent connections integrate evidence for each option. Next, a *gate* population controls action selection by inhibiting the *action* population; this inhibition is released only when specific criteria are met. The *action* population receives inputs from the *accumulator*: as discussed below, this functional connection ensures that only the highest value choice is selected. Note that the *value*, *accumulator*, and *action* populations all represent the two value estimates independently, which guarantees that neural representations of these values do not interfere with one another.

Several parameters dictate how our model responds to urgency and uncertainty, and make gated WTA dynamics possible. The connection between *value* and *accumulator* is initialized with a gain parameter w_{ramp}: this parameter corresponds to the ramp rate in DDM models, and governs how quickly external evidence is accumulated. The *accumulator* population also receives an external input S which can be used to set the initial evidence loaded into the accumulator: S thus represents the prior knowledge of the system. With regards to WTA selection, the connection between *accumulator* and *action* ensures that

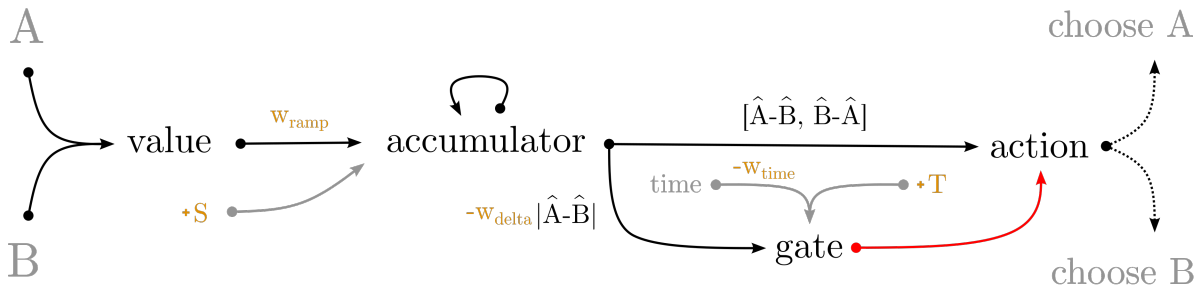

Figure 4.1: Network architecture for the value accumulation and decision making model. Grey text represents inputs and outputs to the model, while black text indicates populations of 1000 LIF neurons. Black arrows indicate connections that compute the indicated functions, while the red arrow indicates an inhibitory connection. Orange quantities are the free parameters of the model: S is an evidence bias fed into the accumulator during model initialization; w_{ramp} is the evidence accumulation rate; w_{delta} controls the strength of certainty-based threshold reduction; w_{time} controls the strength of urgency-based threshold reduction; and T is the baseline decision threshold.

only one choice has positive value. This connection computes the function

$$f(\mathbf{x}) = [\mathbf{x}_0 - \mathbf{x}_1, \mathbf{x}_1 - \mathbf{x}_0], \tag{4.1}$$

which is equivalent to the signed difference between the evidence for choice A and choice B. When choice A has larger accumulated evidence, this function will decode a value with sign $[+, -]$, whereas when choice B has more evidence, it will compute $[-, +]$. We also set the intercepts and encoders in *action* such that the population only represents positive values of $\mathbf{x}$: negative inputs will result in zero neural activity. The result is that the representation in *action* will be positive in the dimension representing the winning choice, and zero in the other dimension. We discuss the straightforward extension of this function to higher dimensions (choices between three or more options) in Sec. 4.5.

Finally, action selection must be delayed until certain decision criteria are met. Without the *gate* population, any integrated evidence in *accumulator* will lead to immediate positive values in *action* and a corresponding behavioral output from the model. In previous sections, we discussed various criteria for an "appropriate" choice, including the passage of time, the meeting of a decision threshold, or the estimation of high confidence in a choice. *gate* realizes all these criteria simultaneously via a weighted combination of internal and external signals. First, *gate* receives a constant background input T, which represents the decision threshold: this positive value must be negated by other inputs to *gate* in order to

remove inhibition from *action*. Next, *gate* receives an input based on the elapsed time t in the current trial; this linearly-ramping signal is multiplied by a constant $-w_{time}$, which represents the strength of temporal urgency. Thus, as time passes, *gate*'s representation decreases by an amount proportional to t and w_{time}. Finally, *gate* receives from *accumulator* an estimate of choice confidence, which is calculated as $f(\mathbf{x}) = |\hat{\mathbf{x}}_0 - \hat{\mathbf{x}}_1|$, the absolute difference between the two value estimates. As confidence grows, the value represented in *gate* decreases, and the effective threshold for action selection decreases.

It is worth noting that, in our model, the likelihood of making a choice increases not with the absolute value estimate of an option, but with the relative difference between that option and all other options. This property arises from two mechanisms. First, the value transmitted from *accumulator* to *action* is based on the signed difference between integrated value estimates: this value must overcome the dynamic threshold imposed by *gate* before *action* will activate and make a selection. Second, the dynamic threshold computed by *gate* is influenced by choice confidence, which depends on the absolute difference between integrated value estimates: when $w_{delta} > 0$, large differences in accumulated evidence will thus compound to produce faster decisions. These mechanisms were chosen for two reasons. First, they allow us to incorporate a (simple) calculation of choice confidence into the network, a feature which many computational models of DM lack. Second, computing the difference between accumulated evidence accommodates negative values, which are featured in this task (and many real-world situations). Many existing DM models, including Nengo's default BG network [220] and IA network [89], rely on accumulation (or comparison) of positive value to a threshold. When predominantly negative values are observed, these networks do not function properly.

4.4 Results

To investigate the DM capabilities of our model and compare its performance to human data, we divide our results into three sections. First, we investigate the dynamics of neural representation in the model, and begin to characterize the influence of model parameters related to urgency and uncertainty. Next, we examine the distributions of accuracy and speed from two human participants (one fast and inaccurate, the other slow and accurate), then attempt to tune our model parameters to reproduce their behavior. Finally, we investigate the SAT tradeoff more generally by simulating a heterogeneous population of agents and plotting the relationship between speed and accuracy. We compare these trends to the human data across multiple difficulty conditions, and analyze the relationship between individual model parameters and simulated speed and accuracy.

4.4.1 Dynamics

We begin by showing the dynamics from a simplified version of the model. In this network, we set w_{time}, w_{delta}, and S to zero, leading to a static decision threshold and an unbiased accumulator. Fig. 4.2 plots the accumulated evidence and the decision variables over time, as cues are sequentially sampled from the environment. The accumulator maintains an accurate estimate of the integrated value over time (blue and orange lines, left panel), and the connection between *accumulator* and *action* computes the signed difference between these estimates (pink line, right panel). When this difference exceeds the threshold imposed by *gate* (green line), inhibition on *action* is released, leading to nonzero values and a behavioral output (transparent blue or orange lines, right panel). We record the time when this decision occurs, and mark it with a green (red) circle, to indicate a correct (incorrect) choice. In this example, the model makes a correct decision after viewing the 7th cue. Fig. 4.3 shows a second simulation with nonzero w_{time} and w_{delta}. As time passes and evidence accumulates, the ramping urgency signal and the growing certainty about differences in accumulated value dynamically reduce the decision threshold. Relative to Fig. 4.2, this changes the point where the signal decoded from *accumulator* overcomes the dynamic threshold, leading to a decision shortly after viewing the 6th cue.

In these examples, altering the model parameters helped the model choose quickly and correctly, but in other cases, promoting speed may prompt an incorrect decision. Figs. 4.4-4.7 plot four examples of the SAT on a single trial of the sequential sampling task. In each figure, we simulate three different networks, which are identical except for a single parameter value, T, w_{ramp}, w_{time}, or w_{delta}. We evaluate the dynamics of these models on the same trial, meaning each model receives identical input values for options A and B. Our figures show that, for each parameter, choosing an extreme value leads to each a slow but accurate, or a fast but inaccurate, decision, while choosing an intermediate value leads to a fast and accurate decision. These figures demonstrate the impact of the four free model parameters on the dynamics of value accumulation and decision making, and indicate that each parameter may influence an agent's SAT.

4.4.2 Individual Behavior

We now investigate whether the behavior of our simulated agents aligns with empirical data. We use the dataset from Fiedler et al. [73], which is available online and is distributed with the Creative Commons Attribution 4.0 International license; see the original paper for experimental details. We first attempt to recreate the behavioral data from representative individuals in the dataset. We selected two individuals based on (a) the distribution of cues

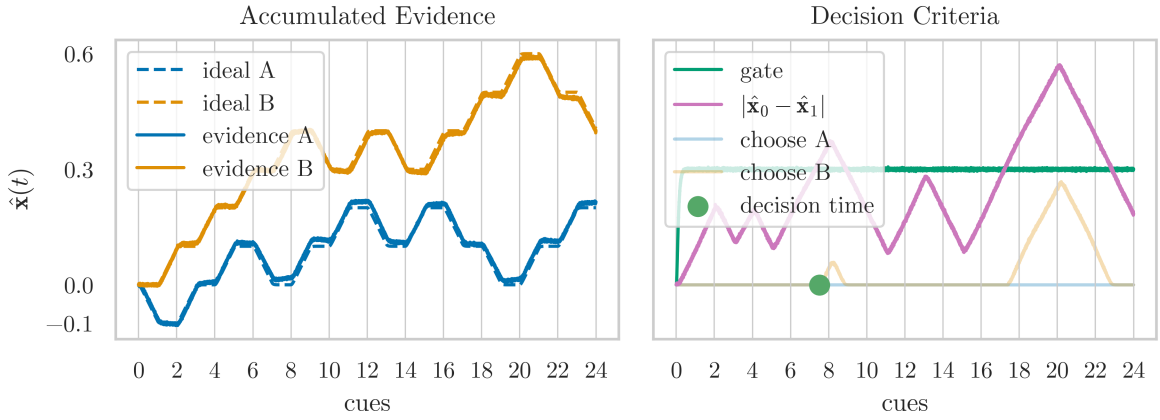

Figure 4.2: Dynamics of the DM network with a static decision threshold. We set w_{time}, w_{delta}, and S to zero, leading to a static decision threshold and an unbiased accumulator. The left panel shows the accumulated evidence for options A and B: the dotted lines are the analytically-computed summation of cue values, and the solid lines are the estimated values decoded from the spiking activity of *accumulator*. The right panel shows the output from the gate population, which must be exceeded by the output of *accumulator* ($|\hat{\mathbf{x}}_0 - \hat{\mathbf{x}}_1|$) before inhibition on *action* is released and a choice is made.

they sampled before making a decision in the hardest difficulty condition ($\Delta P = 0.1$), and (b) their mean accuracy over all trials in this condition. One individual made decisions quickly but less accurately (median number of sampled cues is 8, with accuracy 67%), while the other made decisions slowly but accurately (median cues is 16, with accuracy 84%). To see if our model could reproduce these two extremes of decision making, we ran a hyperpameter optimization on our four parameters using the Neural Network Intelligence (NNI) optimizer [165]. To compute the loss function, we simulated an agent with the same inputs (sampled values of option A and B) that the two human participants viewed during the hardest difficulty condition; if the model agent requested more cues on a trial than did the human, we generated additional data based on the trial's recorded probability that A or B would increase in value. After each trial, we recorded the number of cues sampled by the model and whether it made the correct selection. The loss was set equal to the sum of the absolute difference between the cues sampled by the agent, and the cues sampled by the participant; the total loss across all trials was the sum of these losses. Thus, the optimization attempted to find parameters that make agents choose with similar speeds as the human participant when $\Delta P = 0.1$, but did not factor in whether the agent

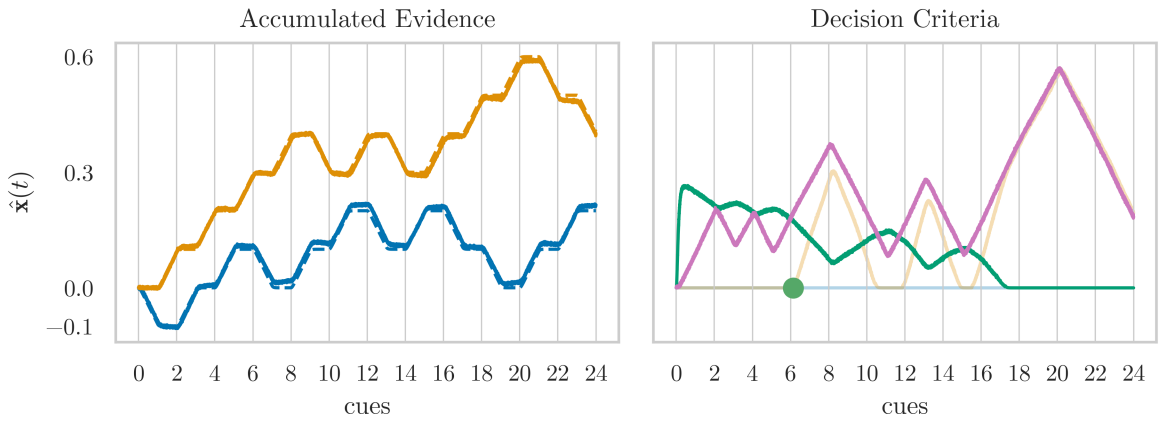

Figure 4.3: Dynamics of the DM network with a dynamic decision threshold. We set $w_{time} = 0.02$ and $w_{delta} = 0.4$ leading to a decision threshold that changes over time and with accumulating evidence.

was similarly accurate (or inaccurate) as the human, nor did it account for the easy and moderate difficulty conditions.

Fig. 4.8 shows the behavioral data from two optimized agents, one fast and one slow, compared to the two human participants, across all three difficulty conditions. We observed that the speed distributions of simulated agents resemble the human data for all three difficulty conditions. Given that these models were only "trained" on the hardest difficulty, this agreement suggests that the optimized parameters may describe core DM attributes of an individual that are invariant across contexts and tasks. We also observed that the mean accuracies of the optimized models were similar to the human accuracies, suggesting that our model may also capture the accuracy differences that accompany these decision strategies.

4.4.3 Speed Accuracy Tradeoff

Finally, we investigate whether our model reproduces the SAT in the general case, and attempt to identify which cognitive mechanisms are primarily responsible for this effect. We begin by showing how the SAT manifests in the empirical data from [73]: for each individual human in the dataset, we recorded the mean number of cues they sampled, calculated their mean accuracy, and plotted these values against each other. Fig. 4.9

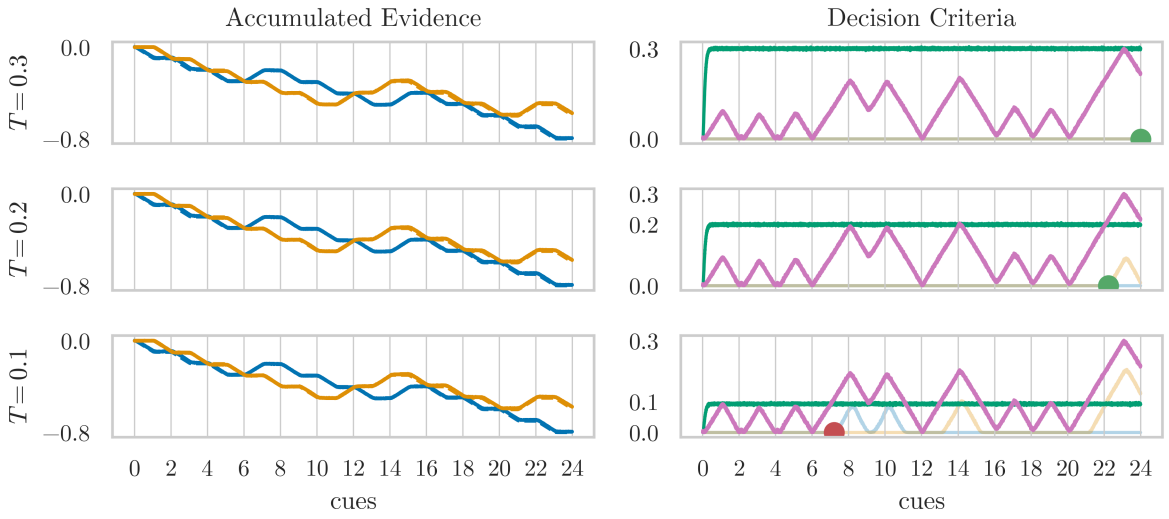

Figure 4.4: Dynamics of the DM network for different values of the decision threshold parameter T. This parameter controls the y-intercept of the dynamic threshold.

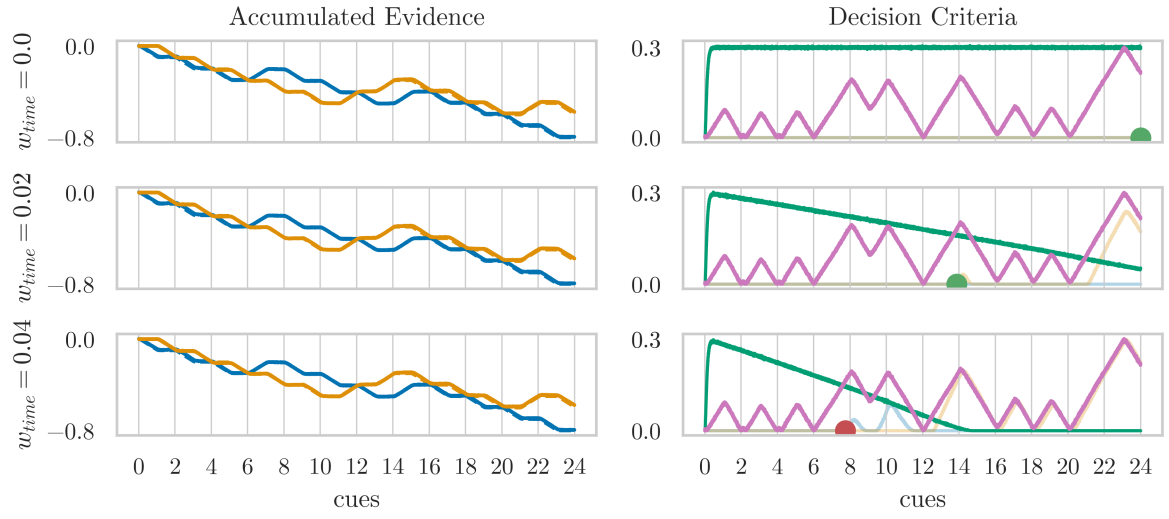

Figure 4.5: Dynamics of the DM network for different values of the urgency parameter w_{time}. This parameter controls temporal decreases in the dynamic threshold.

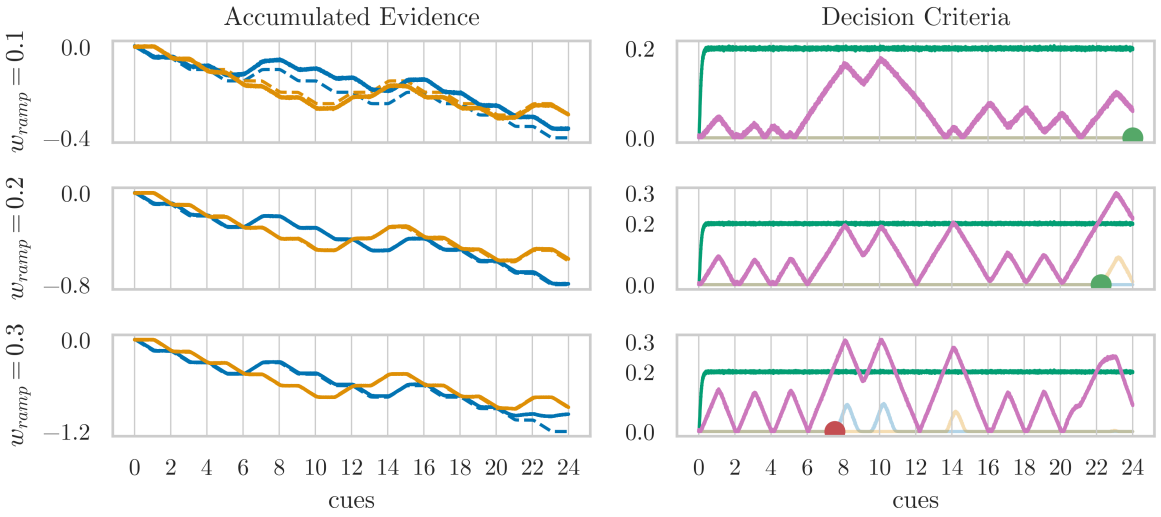

Figure 4.6: Dynamics of the DM network for different values of the ramp rate parameter w_{ramp}. This parameter controls the slope of accumulating evidence (note y-axis scale).

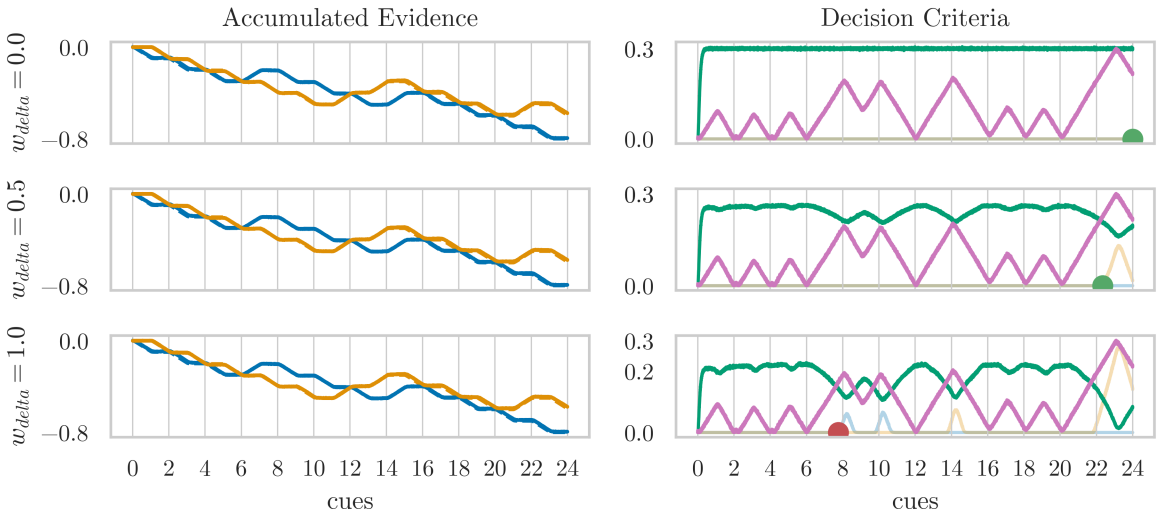

Figure 4.7: Dynamics of the DM network for different values of the confidence parameter w_{delta}. This parameter controls the relationship between differences in accumulated evidence and the dynamic threshold.

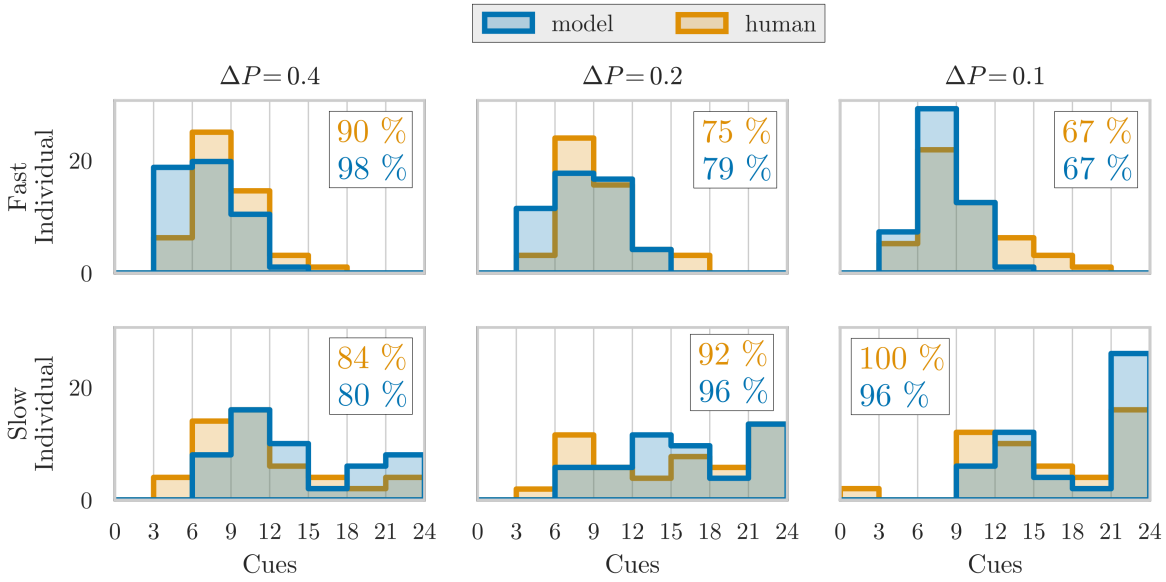

Figure 4.8: Distributions of cues sampled by simulated agents and human participants. We chose a fast and a slow decision maker from the empirical dataset, then optimized model hyperparameters to reproduce the number of cues they sampled before making a decision in the $\Delta P = 0.1$ difficulty condition. The top row plots the fast individual and the bottom row plots the slow individual; human data are in orange and model data are in blue. Optimized agents sample a similar number of cues and have similar accuracy rates across all difficulty conditions, indicating that the optimized DM parameters generalize outside the training set. The y-axis indicates percent responses for each bin of the histogram, and the mean accuracies are indicated in the plot legends.

shows the result: individuals who sampled more cues performed better on average, and the relationship between accuracy and speed is well-approximate with a linear fit. This trend is apparent across multiple difficulty conditions.

To reproduce the empirical SAT in a population of simulated agents, we repeated the optimization in Sec. 4.4.2 for each human in the empirical dataset, attempting to find parameters that caused simulated agents to sample a similar number of cues as their human counterpart on the hardest difficulty condition. As before, the optimization did not account for human accuracies or the cues sampled on the easier difficulties. After finding optimal parameters for each agent, we had them complete the sampling task on all three difficulty conditions, presenting them with the same inputs that the humans saw whenever possible. We plotted the results in Fig. 4.10. Overall, the SAT is evident in

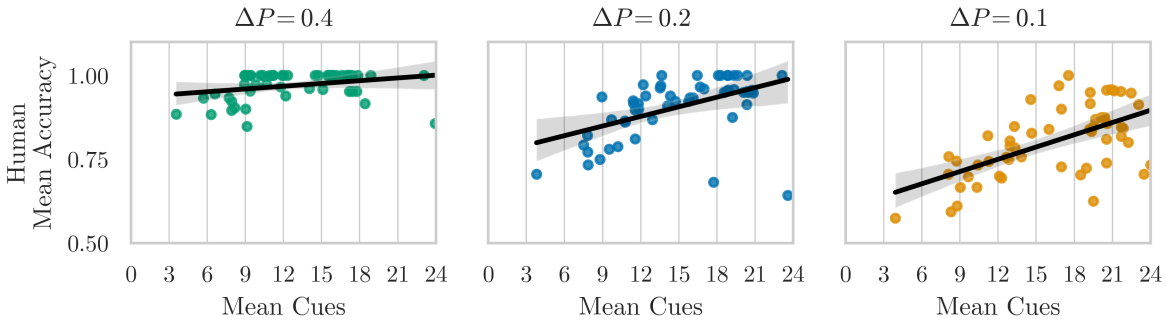

Figure 4.9: Speed-accuracy tradeoff among human participants across three difficulty conditions. Each point represents one individual performing multiple trials of the sequential sampling task. As expected, the mean accuracy of a participant reliably increases with the participant's mean number of sampled cues. More difficult task conditions (smaller ΔP) shifted the SAT curve downward but preserve this relationship. Linear regression was performed using Seaborn [254].

the model data across all difficulty conditions: the simulated variance in mean cues and accuracy captures the diversity of decision strategies present in the human data. Although the optimization only sought to find parameters that reproduced the cues distribution for $\Delta P = 0.1$, the resulting population managed to recreate SAT trends in all difficulty conditions. This result supports the tentative conclusion from previous sections, suggesting that our parameterized model captures important features of DM and the SAT, and that a model trained on one task may generalize to similar tasks and still reproduce human behavior.

We were also curious which model mechanisms were primarily responsible for driving the SAT in our simulated agents. To investigate this, we first looked at whether each model parameter could, on its own, produce the behavioral variance that characterizes the SAT. We chose a set of default parameters for our model ($T = 0.3$, $w_{ramp} = 0.2$, $w_{time} = 0$, and $w_{delta} = 0$), then created a heterogeneous population of agents by scanning over one parameter while holding the others constant. Each agent completed 30 trials of the sampling task, then we plotted agents' mean accuracies versus their mean sampled cues. Fig. 4.11 shows that varying any of the four model variables produces some form of SAT among the simulated agents: as they sample more cues, they become more accurate on average, with slope and intercepts dependent on the scanned parameter. While these trends were not good quantitative fits to the human data in Fig. 4.9, they qualitiatively

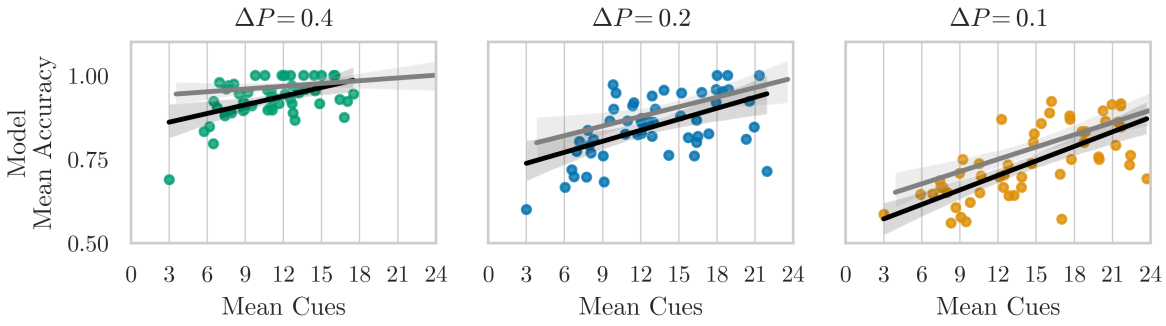

Figure 4.10: Speed-accuracy tradeoff among simulated agents across three difficulty conditions. As expected, the mean accuracy of agents reliably increases with the the mean number of cues they sample, and harder tasks induced more errors. The simulated data align closely with the human data in Fig. 4.9: our model captures the human variance in sampled cues and accuracies, and our best-fit trendlines, which quantify the SAT, are quite similar to the trendlines in the empirical data, replotted here as gray lines.

showed that any of our four model parameters may potentially drive the SAT.

Next, we analyzed the data from Fig. 4.10 to identify which model parameters best explained differences in simulated accuracies and cues sampled. We began by running an ordinary least-squares (OLS) regression on the parameters from the optimized agents; we used the software package sklearn [182] to normalize the variance of each parameters to one, then used the `LinearRegression` function to estimate the coefficients of our four model parameters. We combined the data from all three difficulty conditions for this analysis. This produced a linear fit to the simulated accuracy with score $R^2 = 0.23$ and coefficients $C_T = 0.02$, $C_{ramp} = -0.01$, $C_{time} = -0.05$, and $C_{delta=-0.02}$. This indicates that the parameter w_{time} was the strongest predictor of model accuracy, with larger parameter values leading to lower accuracies. We also ran an OLS regression with the mean number of sampled cues as the dependent variable. This produced a linear fit with score $R^2 = 0.63$ and coefficients $C_T = 1.54$, $C_{ramp} = -1.88$, $C_{time} = -2.95$, and $C_{delta} = -1.06$. Again, w_{time} was the strongest predictor of cues sampled by the simulated agents. To visually confirm these relations, we plotted mean accuracy and mean cues samples as a function of w_{time} in Fig. 4.12: this plot confirms that speed reliably increases, and accuracy reliably decreases, for agents with larger w_{time}.

Finally, we performed a principle component analysis to look for correlations between model parameters in the optimized agents. Using sklearn's `PCA` function, we ran a 4-

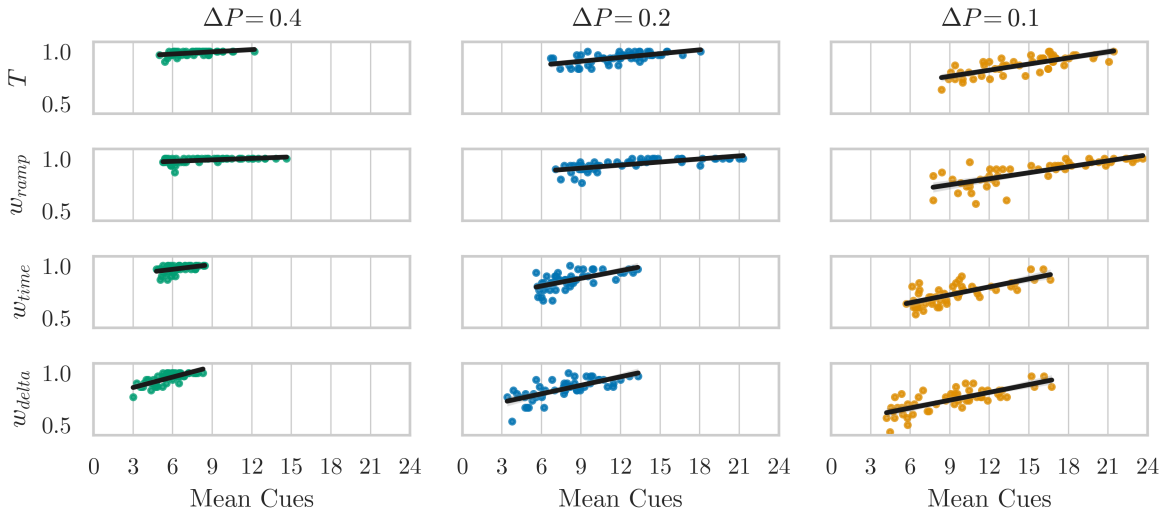

Figure 4.11: Speed-accuracy tradeoff among simulated agents that differ by only one parameter. For each row, we created a population of agents with different values of the indicated variable, then plotted their mean number of sampled cues against their mean accuracy (y-axis). We found that each parameter could qualitatively explain the SAT: accuracy increased linearly with cues sampled, with lower intercepts and greater slopes for harder tasks. While these data do not quantitatively reproduce the empirical data from Fig. 4.9, they show that a SAT may emerge from variance in any model parameter.

dimensional PCA on our parameter data, then plotted each pair of principle components against one another, coloring the data by accuracy or mean cues sampled. The results are shown in Figs. 4.13-4.14. While we did not observe any strong clustering between model parameters, the relationship between the second, third, and fourth principle component and our two dependent variables was again affirmed. This suggests that our model parameters do not require specific configurations (such as high decision thresholds paired with low temporal urgency) to recreate the human data. In the next section, we discuss how extensions of the optimization and PCA approach may be used to further identify the contribution of each model parameter.

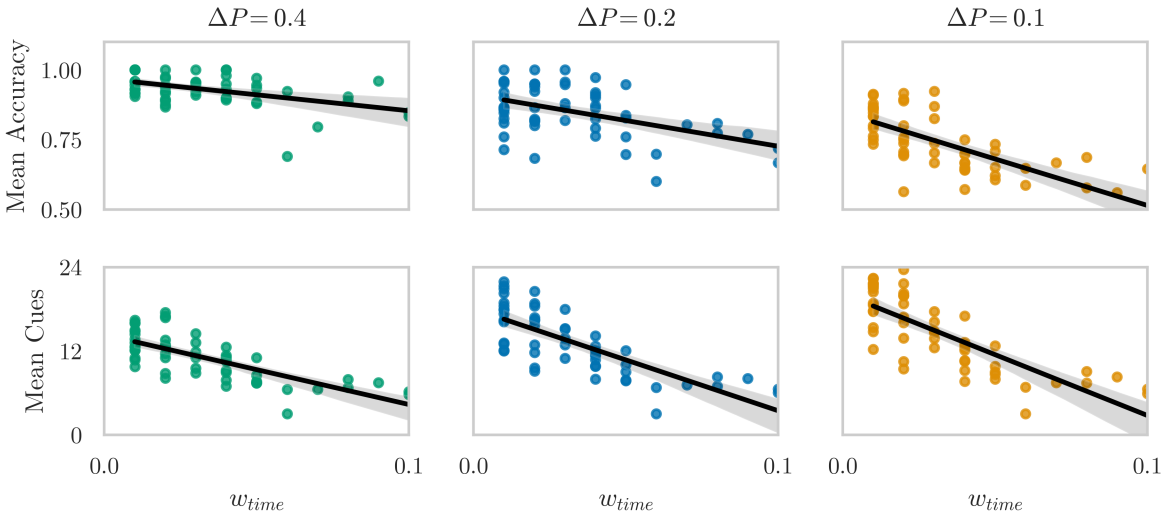

Figure 4.12: Mean speed and accuracy of simulated agents from the optimized population as a function of the urgency parameter w_{time}. We performed an OLS regression on the optimized agents depicted in Fig. 4.10 and identified the parameters w_{time} as the strongest predictor of agent speed and accuracy. We then plotted each agent's mean speed and accuracy as a function of the agent's w_{time} parameter value. This plot confirms that these two dependent variables strongly correlate with the urgency parameter. Note that the vertical striping is an artifact of parameter discretization in NNI optimization.

4.5 Discussion

In constructing our spiking neural model of value accumulation and decision making, we strove to make it biologically realistic and cognitively plausible. In this section, we discuss the extent to which we achieved these goals, and identify issues for future work.

Unlike our amygdala model in Ch. 2, the model we presented in this chapter does not have a precise neuroanatomical mapping: we labelled our neural populations based on their functional properties, rather than their anatomical identities. However, this does not mean that our model does not correspond to known DM circuits in the brain. In Sec. 4.2.2, we discussed how various areas of cortex and basal ganglia might realize a DM circuit, but pointed out that many brain areas contribute to each of these functions. We felt that explicitly giving anatomical labels to our populations would obscure this complexity, and give a false sense of certainty about the brain areas involved. It is likely

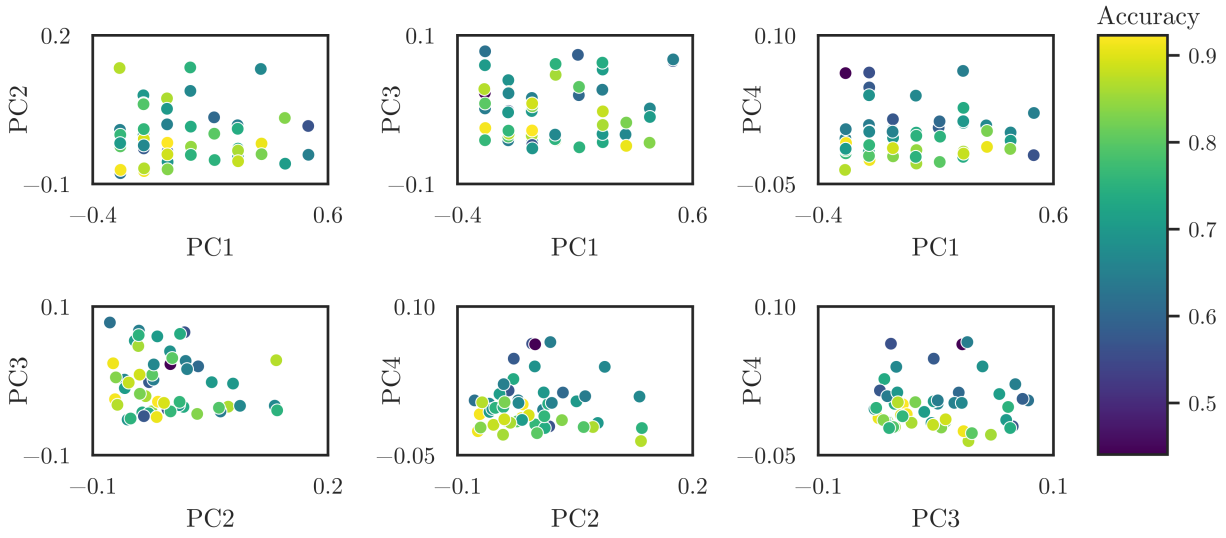

Figure 4.13: Principle component analysis for model parameters, colored by mean accuracy. We ran a 4-dimensional PCA on our parameter data and plotted each pair of PCs against one another. The lack of clustering indicates our parameters are mostly independent, while the color gradient in the bottom row indicates that accuracy correlates with PCs 2-4.

that DM circuits, such as one we presented in Fig. 4.1, occur in multiple places throughout the brain, with each specialized to a different sensory or action modality. However, if we were forced to identify the brain regions associated with our neural populations, we would say that *value* estimation occurs in modality-specific regions (e.g., V1-V4 for visual estimates, or STS/TPJ for social estimates), and value *accumulation* occurs in vmPFC and vSTR. These integrated estimates are sent to areas in motor cortex, including SMA, that help plan and execute *action*s. These motor areas are under the inhibitory control of nuclei within the BG, which receive value estimates from cortex and *gate* action selection by projecting back to cortex. Future work should gather electrophysiological data from neurons in these regions for behaving animals in order to prove or disprove this proposed functional mapping.

With regards to cognitive realism, our model included a variety of mechanisms that dynamically govern the speed and accuracy of DM. Like most DDM models, our model includes a static threshold parameter T and a ramp rate w_{ramp}. Similar to the drift rate in DDMs, our ramp rate determines how fast the decision variables ramp towards threshold; unlike DDMs, our ramp rate leads to the independent accumulation of two

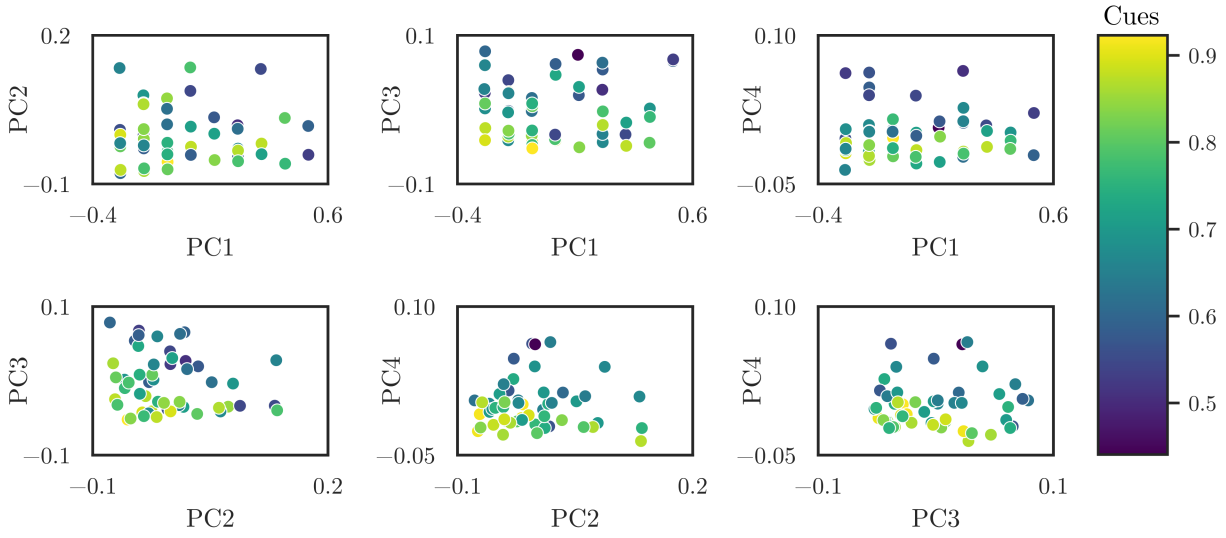

Figure 4.14: PCA for model parameters, colored by mean cues sampled. We ran a 4-dimensional PCA on our parameter data and plotted each pair of PCs against one another. The lack of clustering indicates our parameters are mostly independent, while the color gradient in the bottom row indicates that speed correlates with PCs 2-4.

decision variables (as per the IA model), and does not include a noise component. However, the spike noise inherent in neural representation, both in *value* and in *accumulator*, causes the integration of evidence to be noisy, leading to variations in decision time that are analogous to the DDM. In addition to threshold and accumulation rate, we investigated two mechanisms that might contribute to speed and accuracy: a ramping urgency signal that increases over time; and an internal estimate of choice certainty. We proposed heuristics by which these quantities might influence a dynamic threshold, and parameterized these rules with two variables, w_{time} and w_{delta}, that control the strength of these effects. The result is a model with four free parameters, any of which might vary within a human population and produce different behaviors on DM tasks. Indeed, we found that, by varying these four parameters and simulating a sequential DM task, we could produce variable agent behaviors that correspond to the SAT.

We also compared the behaviors of our agents to an empirical dataset from [73], in which hundreds of human participants completed the same sequential sampling task. We identified two individuals from this dataset whose behavior reflected a slow but accurate DM strategy, and a fast but inaccurate DM strategy. We then used an offline optimization

technique to find model parameters that reproduced the behavioral signatures of these two strategies. The resulting agents successfully recreated the empirical data from these two humans. Building on this success, we repeated this optimization for every human in the empirical dataset, then analysed simulated and empirical SATs by plotting individuals' mean accuracies against their mean number of cues sampled. We quantified the SAT by using linear regression to draw best-fit trendlines, and found that our model closely reproduced the empirical data.

In many ways, our model resembles the typical DDM model as applied to DM and the SAT: we fit model parameters like ramp rate and decision threshold to the human data, and used this to draw conclusions about the cognitive mechanisms of DM. However, our model also extends DDMs in several important ways. First, our model simulates networks of spiking neurons, and realizes all high-level model parameters within the network itself (i.e., by scaling synaptic weights on particular connections). We propose specific mechanisms by which cognitive operations like value accumulation and gating may be realized within neural networks, and map these mechanisms onto anatomical areas. Most DDMs are purely mathematical models, where the parameters and mechanisms only qualitatively map onto brain. Our model is therefore both more cognitively and biologically specific than DDMs. Second, our model is applied to a sequential sampling task, whereas most DDMs are applied to perceptual tasks; while both models can likely be adapted to perform both tasks, the organization of our model is arguably better suited to flexibly accommodate complex cognitive tasks than the DDM, which is limited by a single decision variable and a lack of internal state monitoring.

While we investigated a two-choice task in this chapter, our model may be easily extended to tasks involving three or more choice alternatives. First, the input space would be expanded to include the values of each possible stimulus, and the representations of *value* and *accumulator* would increase proportionally. Because we represent each decision variable as an independent dimension in these populations, this would require a linear increase in the number of model neurons. Next, the connections out of *accumulator* would be generalized to higher dimensions. To ensure that *action* receives an input that is positive only in the dimension of the "winning" option, the connection between *accumulator* and *action* could compute the function

$$f(\mathbf{x}_i) = D\mathbf{x}_i - \sum_{j}^{D} \mathbf{x}_j. \tag{4.2}$$

where D is the number of candidate actions, and i is the index over these actions. In other words, for each possible option $i \in D$, the connection would compute the difference

between the value of that option and each other option j, then sum them together into a single value. This equation reduces to Eq. 4.1 for $D = 2$. Finally, to compute certainty, the connection between *accumulator* and *gate* would compute some function over all accumulated values that results in a one-dimensional certainty estimate. Many functions could satisfy this criterion, and we suspect that the brain uses learned heuristics to simplify this computation. Depending on its complexity, this functional connection may require a secondary population with recurrent connections, but no further changes to the overall network structure presented in Fig. 4.1 would be required. Investigating the computational properties and behavioral signatures of such a network would be a fruitful avenue for future research.

Another topic for future work relates to parameter identification and minimal model specification. In this chapter, we repeatedly showed that each of our model parameters independently contributed to the SAT, both in the general case, and for optimized agents. Preliminary analysis with OLS and PCA suggested that the temporal urgency parameter w_{time} explained the most variance in simulated speed and accuracy. However, we did not investigate whether simpler versions of our model could adequately reproduce the trends we identified in the human data. It would be interesting to repeat the parameter optimization in Sec. 4.4.3, but with one, two, or three of the model parameters fixed to zero. If parameter optimization could not recover the empirical SAT trends with specific parameters missing, it might indicate that the associated cognitive mechanisms are essential for managing the SAT, whereas the other mechanisms provide additional means to flexibly control the SAT given task constraints. Future work might also explore different mechanisms for computing certainty, urgency, or WTA competition: we proposed simple computations for these quantities, but the exact functional form of these computations may affect the shape of the emergent SAT in a population of model agents.

As mentioned above, one advantage of simulating spiking neural networks is the ability to compare low-level model outputs to low-level biological data, for example by comparing the neural activity of model populations to neural data from behaving brains. While the design of our model was broadly informed by the representational properties of several areas in cortex and BG, we did not perform a thorough comparison between model activity and empirical tuning curves, like we did in Ch. 2. This could be a valuable area for future research that could easily validate (or invalidate) the structure and mechanisms of our model. For example, some behavioral evidence supports the notion of an urgency signal that ramps with time [51, 41], which may reduce decision threshold or increase ramp rates. However, neural evidence from motor cortex suggests that decisions are made when neural activity reaches a fixed threshold, which is invariant across urgency, task difficulty, and other contextual variables [71]. Comparing simulated and empirical neural activities,

given a model that reproduces the behavioral results of animals performing a task, may help to resolve this theoretical ambiguity, and inform the design of future empirical studies that seek to differentiate the neural mechanisms responsible for the SAT.

Finally, there are a few behavioral effects from empirical studies that might be worth studying using our model. In some sequential sampling tasks, participants develop a bias towards one choice option after repeatedly choosing (or being rewarded for choosing) that option [91, 96], even when task instructions indicate the independence and equality of options on each trial. These biases may reflect prior knowledge about option values, differential reward rates for each option, or other cognitive mechanisms based on DM heuristics. The structure of our model should accommodate a variety of mechanisms related to cognitive bias: in fact, we already simulated a mechanism for value bias through the parameter S. However, we did not investigate the effects of this parameter, since the empirical dataset from [73] did not feature the development of biases across trials. In general, using our model to investigate the idiosynchratic properties of human DM, and to test hypothesis about their neural bases, is a worthwhile endeavor.

4.6 Conclusion

In this chapter, we presented a neural model of value accumulation and DM that performed a sequential sampling task. The structure of our model was based on the functional neuroanatomy of DM in the human brain, and its cognitive mechanisms realized two decision criteria that influence DM, urgency and uncertainty. Four key parameters in our model governed how simulated agents approached the speed-accuracy tradeoff: the decision threshold, the ramp rate for accumulating evidence, the strength of temporal urgency, and the strength of decision confidence. We explored how these parameters explained differences in human behavior with regards to the SAT: in particular, we found that we could reproduce (a) the behavior of individual humans who followed characteristic decision strategies, and (b) trends related to the SAT across a population of humans performing the task. Our model thus provides a realistic account of the relationship between the neural mechanisms underlying valuation and DM, and the behavior of individuals performing a complex DM task. We analyzed the relationship between model parameters and emergent SATs using several techniques, and tentatively concluded that increasing urgency driven by the passage of time was the strongest determinant of agents' speed and accuracy. We concluded by discussing several theoretical and practical extensions for our model. In the next chapter, we shift our focus away from value integration and action selection, and towards the process of value estimation and error-driven updating.

Chapter 5

Reinforcement Learning and the Trust Game

Author's Note: some of the of content in this chapter was previously published as a journal article in the Proceedings of the 44th Annual Conference of the Cognitive Science Society [54]. Code is available on GitHub.

5.1 Introduction

In the final research chapter of this thesis, we explore how the brain estimates the value of candidate actions based on trial, error, and external feedback. Learning the relationship between one's actions, the state of the environment, and the consequences of those actions is central to the process of value estimation. We study this process using reinforcement learning (RL), a popular and successful framework for training intelligent agents to act in complex environments. RL is widely believed to occur within various areas of the brain, helping animals develop behavioral strategies as they act in, and react to, changes in the environment. We extend previous neural models of RL in several ways, notably by creating a valuation network with biological constraints, then applying this network to a social task where individuals interact with adaptive software opponents in a two-player game. Our focus in this chapter is developing cognitive agents and applying them to social task that involve learning: to validate our models, we also ran a human experiment that provided a dataset for comparison and analysis. Our results suggest a qualitative match between simulated learning and human learning; our agents adopted strategies that resemble human

strategies at a high-level, especially with regards to prosocial activities like generosity and reciprocation.

In many ways, the NEF network we present in this chapter is the culmination of the models we developed in the previous chapters. Online learning within the network uses the PES rule, in conjunction with error populations and inhibitory gates, to update the synaptic weights that realize value encoding. Although this learning is not exactly associative, it does map states of the world and candidate actions onto learned value estimates, and uses many of the components and insights from the amygdala model in Ch. 2. In order to manage the real-time processes of perception, action, and learning, our network uses gated working memories to store and clear information from a cognitive workspace, and uses functional connections to route and transform information between model components. Our work with biologically-plausible networks in Ch. 3 shows that such functional models may be implemented in biological detail without losing cognitive capacity. Finally, our model uses a simple decision system to select actions once valuation is complete: while the focus of this model is updating value estimates based on feedback, we do implement a decision system that simplifies the action selection system in Ch. 4 in order to close the loop between perception, action, and environment. The resulting model is broadly applicable to cognitive tasks that involve learning and behavior, and the insights we gain about high-dimensional state representation and cognitive control help us understand the challenges faced by the brain when performing difficult, multi-step tasks.

We begin this chapter by reviewing the mathematics of RL and the associated functional neuroanatomy. We then introduce the trust game (TG) and discuss how it has been used to study prosocial behavior and the development of trust in psychology and economics. We then conduct an empirical experiment on human participants, producing a behavioral dataset that we later use to validate our model. After explaining the experiment and the data collection process, we introduce our cognitive agents, and explain how they use RL to play the TG. In order to gain a broader understanding of how RL may be realized in cognitive systems, we design and simulate three classes of agents using three different cognitive frameworks: deep neural networks, ACT-R, and the NEF. For each of these architectures, we step through the relevant cognitive components and introduce the free model parameters. Next, we apply our models to the TG, showing how parameterized agents learn through trial and error. Finally, we simulate heterogeneous populations of agents who play the TG against adaptive opponents. Comparing the simulated and empirical data, we find many high-level similarities but many low-level discrepancies. We conclude by discussing the successes and failures of our models, and introducing possible extension for future work.

5.2 Background

5.2.1 Reinforcement Learning

Reinforcement Learning (RL) is a widely-used framework for understanding how humans and other animals update their behavior based on external feedback [230]. To briefly summarize the theory behind RL, an agent learns some relationship between the *actions* it takes, the *state* of the environment, and the *rewards* it receives. When feedback is provided by the environment, in the form or rewards or punishment, the agent uses a learning rule to minimize the differences between its predictions and the observed outcomes. When making a decision, an agent observes the current state of the environment, considers the value of each candidate action, and chooses one with high estimated value. Over the course of learning, the likelihood that the agent chooses a rewarding action should increase.

Numerous studies have shown that the signals and learning rules proposed by RL map onto reward-prediction errors (RPEs) and synaptic changes in the brain [84]. In general, these studies find strong correlations between neural activity in dopaminergic neurons, such as those in the ventral tegmental area (VTA), and the RPEs used in many RL algorithms. For instance, animals generate dopaminergic signals whose magnitude is proportional to the differences between expected reward and observed reward. During early training, an animal does not expect to receive a reward when it chooses an action, so this RPE signal is positive, and is generated immediately following the delivery of reward. As the animal learns, it begins to predict that a reward will be received whenever that action is taken. This has two measurable effects. First, when an expected reward is *not* received, the signal generated by the brain will have a negative (decoded) value, as would be predicted by the sign of the RPE in an RL algorithm. Second, the timing of this signal shifts: instead of appearing when the reward is observed, it appears when the action is chosen (i.e., when the animal generates a prediction about imminent reward). Studies have also shown that RPEs in the brain are associated with synaptic update in various areas of cortex (those implicated in value estimation, see Sec. 1.3): the notion that these RPEs drive value updating is also consistent with the neurmodulatory action of dopamine with respect to synaptic plasticity.

In addition to these low-level signatures of RL in the brain, several high-level theoretical frameworks describe how the large-scale organization of the brain is consistent with RL. For example, the reinforcement learning and decision making framework (RLDM, [81]) describes three independent learning and decision making systems in the brain: the Pavlovian system, the habitual system, and the goal-directed system. The Pavlovian system controls reflexive responses to stimuli that were common and adaptively significant in our evolutionary history, such as fight-or-flight reflexes or sexual behavior. In this system,

candidate actions are inborn, inflexible, and preprogrammed, and may often be triggered by a fixed decision structure. However, RL may leverage this system to generalize these responses to novel situations (environmental states outside the original evolutionary context) that require similar reactions. A perfect example of this is associative learning and fear conditioning in the amygdala: while the structure of this system is fixed (with respect to the types of stimuli that may be associated, and the types of behavior responses that may be triggered), it can learn new associations between these stimuli through external feedback and reinforcement.

Next, the habitual system can be used to learn associations between external stimuli and behavioral responses, such that the animal repeats actions that previously produced desirable outcomes in similar situations. RLDM argues that this kind of learning dominates later stages of learning, and is largely independent of working memory, cognitive manipulation, and motivational state: this approach directly associates states and actions with rewards, and does not require explicit forward planning. The anatomical basis of the hypothesized habitual system is similar to the value estimation and integration networks we reviewed in Sec. 1.3. Finally, the goal-directed system uses learned associations between actions and outcomes to infer which actions will produce the best outcomes from the perspective of an agent's current goals. This system is deliberative, in that it uses working memory, motivational state, and imagination to plan action sequences that will achieve specific goals, rather than produce statistically-favorable outcomes like the habitual system. RLDM predicts that the goal-directed system will dominate at the early stages of learning, when an animal is actively learning the dynamics of the environment and exploring different subgoals and strategies. Value modulation and executive areas like the dlPFC are important in facilitating goal-directed computations.

5.2.2 Q-learning

To model human learning in the TG, we chose an RL algorithm of moderate complexity, a model-free approach called *Q-learning*. In Q-learning, an agent interacts with the environment by choosing actions that it expects to produce high rewards, both immediately and in the future. The agent represents the *state* of the environment at any given time as a high-dimensional vector s, and must choose an *action* a in order to proceed. Q-learning agents learn a model called a *value function* $Q(s, a)$, which estimates the value of being in state s and taking action a. As the agent explores the environment, it observes how taking action a in state s produces *rewards* $R(s, a)$ and transitions the environment into a new state s'. It then estimates $Q(s, a)$ by combining $R(s, a)$ with an estimate of the value of possible future states, $Q(s', a^*)$, where a^* is any available action in the new state s'. The

value $Q(s, a)$ is greatest when (s, a) leads to both high immediate reward *and* puts the agent into a new state which has high estimated value.

To choose an action, an RL agent must also have a *policy* $\pi(s, a)$, which maps between the current state of the environment s and the probability of choosing each candidate action a. If a Q-learning agent learns a good value function (one the accurately estimates immediate rewards and expected future values), then a good policy (one that leads to high rewards over an episode containing many actions and state transitions) is the so-called *greedy* policy. This policy simply chooses the action that maximises $Q(s, a)$, given the current state s. Not only is this policy intuitive and easy to implement, but various theorems in RL guarantee that it is, in fact, the optimal policy: it will generate higher total rewards than any other policy. Therefore, once an agent has successfully learned $Q(s, a)$, it should adopt the greedy policy.

This leaves two questions: how does the agent learn a good value function, and what policy should the agent adopt while it is learning the value function? These questions are actually closely related: in order to learn a good value function, the agent must *explore* the state and action space thoroughly, gaining as much experience as possible to build statistically (or dynamically) robust models of reward and state transitions. RL theory guarantees that a value function will only truly model the environment if the agent visits each (s, a) an infinite number of times. Obviously, this is impossible in practice: at some point, the agent must cease exploration and being to *exploit* its learned value function via a (more) greedy policy. The tradeoff between exploration and exploitation is a complex one, and has been approached in numerous ways, both computationally and biologically. One common technique to balance exploration and exploitation is to adopt a policy that initially favors exploration, but which transitions to exploitation over time. Exploratory policies favor the selection of novel actions and the investigation of new environment states. Three common exploratory strategies are random action selection, which randomly selects a and causes the agent to inadvertently explore many novel s; curiosity-driven exploration, which artificially increases the value of unknown (s, a) to encourage the agent to investigate them; and optimal sampling methods, which explicitly track an agent's uncertainty about (s, a) and attempt to reduce this uncertainty over time.

The second question is how to learn a good value function $Q(s, a)$. In this chapter, we train our networks using a temporal difference (TD) learning rule:

$$\Delta Q(s, a) = \alpha\big[R(s, a) + \gamma \max_{a^*} Q(s', a^*) - Q(s, a)\big], \tag{5.1}$$

where s is the current state, a is the chosen action, s' is the observed next state, R is the observed reward. $\max_{a^*} Q(s', a^*)$ is the maximum of the value function over all

possible actions in the next state, α is the learning rate, and γ is the *discount factor*, a free parameter used to control the relative value placed on immediate rewards versus expected future value. Conceptually, this rule updates the value function based on the difference between observed rewards $R(s, a)$ and expected rewards $(-Q(s, a))$, as well as the value of future states. In the so-called *terminal state*, the final (s, a) before an episode ends, this rule should produce a value estimate equal to the average reward obtained in that state. For all preceeding states, Q should account for the average immediate reward, and the expected discounted value of future states. This learning rule is often referred to as TD(0), because it only accounts for a single (s, a, r, s') transition: extensions of this rule can be used to update Q based on a memory of many previous transitions, but we do not investigate those learning rules here.

5.2.3 Social Value Orientation and the Trust Game

One interesting challenge for models of learning and decision making in social contexts, and for RL in particular, is to explain the emergence of prosocial behavior. A great deal of recent research focuses on cooperative and altruistic behavior in social contexts, analysing prosociality from a categorical perspective [194, 205], a computational perspective [36, 243], and a neuroanatomical perspective [180, 231, 239]. We reviewed the neuroanatomy of proself versus prosocial decisions in Sec. 1.3.2; a central theme of our summary was that the brain contains distinct systems for evaluating the proself and prosocial value of candidate actions and environmental states. These divisions have also been repeatedly noted from a theoretical perspective, and the internal and external variables associated with prosocial decisions have been extensively analyzed [194, 205, 46].

Social dilemmas like the prisoner's dilemma, in which individuals are driven by competing proself and prosocial objectives, are frequently used to study the emergence of prosocial behavior [184, 205, 243]. By recording the neural activity of humans and animals playing these games, researchers have also used social dilemmas to clarify the neural basis of these strategic decisions [154, 195]. Finally, RL models that relate these activities to specific signals and computations have help clarify the symbolic nature of these neural representations [94, 18, 150]. However, more computational work is needed to elucidate the relationship between RL algorithms and emergent prosocial behavior. To understand the origins and consequences of social value orientation (SVO), we need models that (a) are cognitively and neurally plausible, (b) can be generalized to multiple cognitive architectures and behavioral tasks, and (c) explain a variety of empirical results.

We study the relationship between SVO, RL, and biological cognition using computational models that learn to play the trust game (TG), a two-player, turn-based game

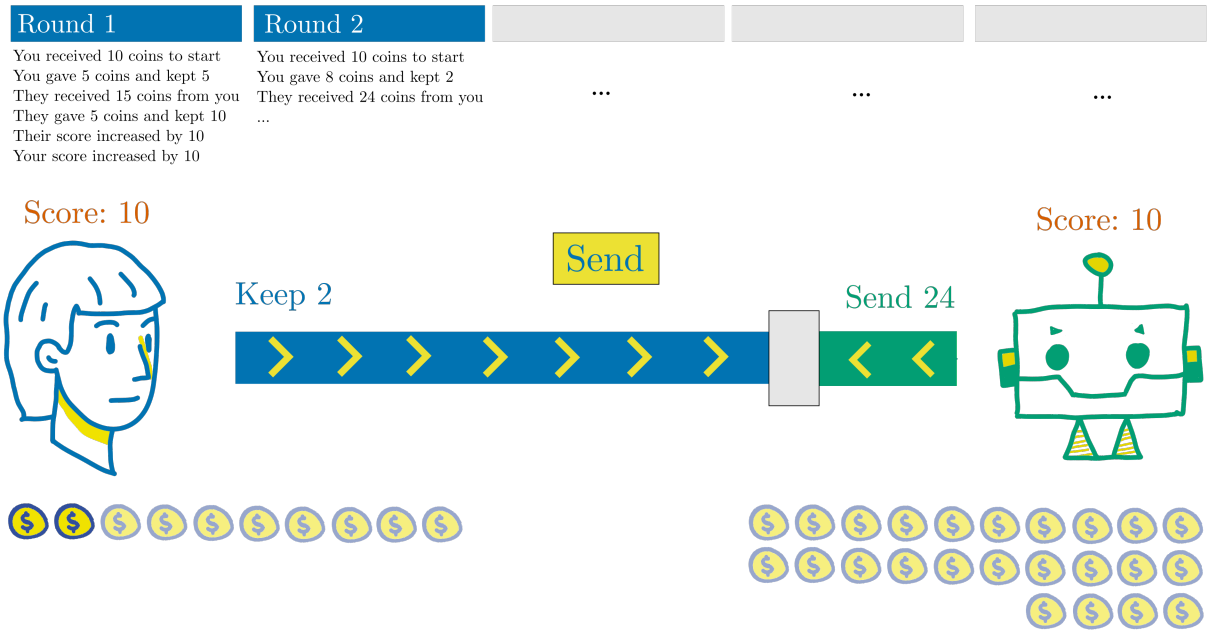

Figure 5.1: Our trust game consists of five consecutive rounds played against the same opponent. The investor begins each round with 10 coins, and their investment is multiplied by three. The total score is the sum of coins collected across all five rounds. Participants are randomly selected to play the investor or trustee. Opponents are adaptive computer agents.

in which individuals repeatedly receive and reallocate resources. In each round (or turn) of the game, the first player (the "investor"), receives 10 coins, then gives some of these coins to the second player (the "trustee"), keeping the rest. The trustee receives three times this many coins. Finally, the trustee returns some number of the resulting coins to the investor. A single game consists of five rounds. Each player's final score is the total number of coins collected across all five rounds. In the TG, greater rewards are earned if both players invest and return generously, but each player will only do so if they trust their opponent to reciprocate in the future. Fig. 5.1 provides a visual description of the TG.

Human behavior in the TG has been widely studied from numerous perspectives, including personal and cultural analyses [115], neural and cognitive analyses [47], determinants of prosocial tendencies [236], and more. Behavior in the TG has a clear prosocial component that is distinct from maximizing personal rewards: while investor behavior is most closely correlated with expectations of repayment and perceived trustworthiness, trustee behavior

is most closely correlated with prosocial tendencies [11]. For instance, in one-round versions of the TG, many trustees will return coins even though their generosity cannot be repaid in future interactions. Prosocial behavior is motivated by numerous drives, including reciprocity, inequality-aversion, and altruism [184, 47], and the activity of numerous brain structures correlates with prosocial values estimates and prosocial behaviors, including the amygdala, striatum, TPJ, mPFC, and dlPFC [97, 112]. Evidence from psychology and neuroscience confirms that learning is a critical component in social dilemmas: individuals adapt their strategies in response to specific instances of betrayal [152], to the opponent's behavior in the recent past [70], and to trust estimates made in previous games [43, 240].

We should note that "trust" is a complex phenomenon that is defined in many different ways, and is studied using many different methods. In keeping with previous TG research [115], we consider trust to be operationally defined: it can be measured by counting the coins transferred between participants under various conditions. Needless to say, the economic definition of trust as "a willingness to bet that another person will reciprocate a risky move (at a cost to themselves)" [33] is fairly limited, with respect to the many cognitive processes and social norms that influence trust-related behaviors in the real world. Nonetheless, we consider the TG to be a valuable tool for studying prosocial behavior.

In this chapter, we use computational models to investigate the emergence of prosocial behavior. To model SVO, we incorporate two additional terms into the reward function of RL agents, which cause agents to explicitly value the rewards obtained by others. In keeping with other computational models of SVO [112, 42, 164], an agent's overall reward is a weighted combination of self-reward, other-reward, and reward-inequality, where the relative weighting determines the agent's degree of SVO. Using this reward function, we train three distinct cognitive architectures to play the TG. We introduce each architecture by explaining its mechanisms and parameters, and by reviewing its cognitive realism. We then simulate and train a heterogeneous population of agents, then compare the distribution and dynamics of simulated data with human data in the TG. We conclude by discussing whether RL, and specifically our implementation of SVO, is a suitable framework for studying prosocial decision making, independent of the computational implementation of agent's internal model.

5.3 Methods

To model human learning in the TG, we used Q-learning to train agents from three different computational architectures. Each architecture has access to the same state information from the environment, and selects between the same number of candidate actions. However,

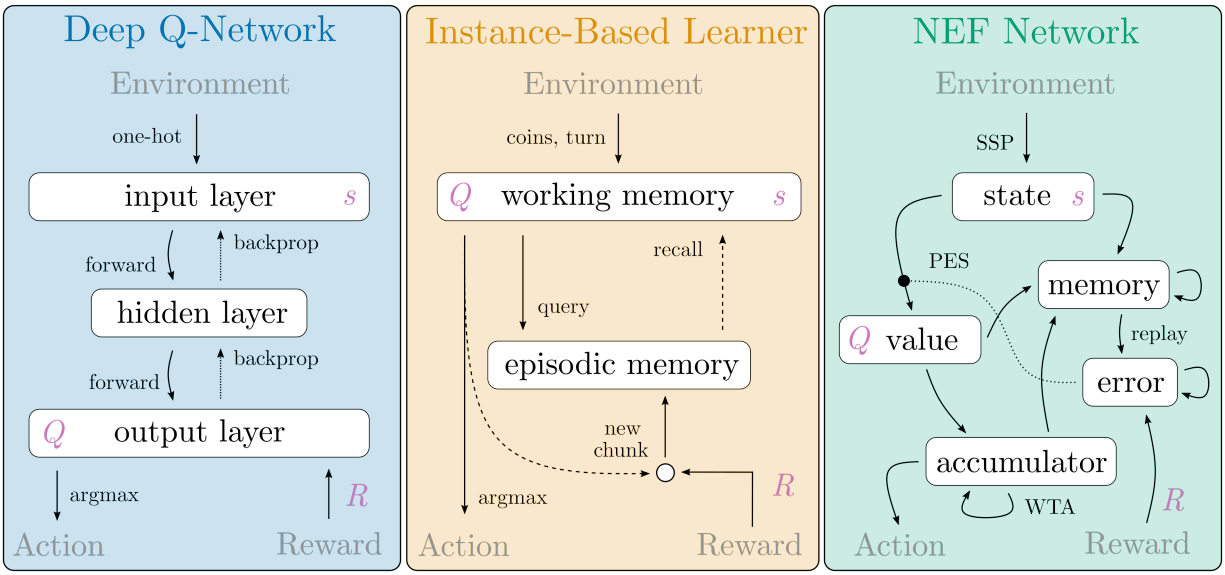

Figure 5.2: Network architectures for the three RL agents. For the deep Q-network, each box represents a layer of ReLU neurons, which form feedforward connections in a dense matrix. The network is trained with backpropagation (dotted lines) driven by Q-learning. For the instance-based learner, each box represents a list of chunks, which are formed after every transition, appended to the list during formation or recall, and updated based on a modified Q-learning rule. Dashed lines indicate blending operations. For the NEF network, each box represents a population (or network) of LIF neurons, and curved arrows indicate synaptic connections with lowpass filters. The dotted line applies the PES learning rule to the decoders in *state*, realizing Q-learning. Grey text indicates model inputs and outputs; pink text indicates the important RL variables.

the internal representations used, the models that are learned, and the mechanisms of the TD(0) learning update, differ significantly between these architectures. By investigating how all three architectures learn to play the TG, we gain a broader understanding of how RL and SVO relate to prosocial behavior on this social task. These architectures are presented graphically in Fig. 5.2.

5.3.1 Social Value Orientation and the Reward Function

In the TG, the state contains two pieces of information: the current round of the game (1-5) and the number of coins currently available (0-30). Each round, the agent's action

determines the number of coins they will give away versus keep. In order to explore the (s, a) space of the TG, we use an ϵ-greedy strategy: agents take random actions with a probability ϵ, which decays from one to zero over the course of learning. If a random action is not taken, the agent chooses the action with the greatest Q-value. To model SVO, agents compute a weighted sum between the rewards they themselves received, the rewards their opponent received, and any inequalities between them:

$$r = w_s r_s + w_o r_o - w_i |r_s - r_o|, \tag{5.2}$$

where w_s, w_o, and w_i are the weights for self-reward, other-reward, and inequality, and the rewards r_s and r_o are the coins earned on any given turn. For convenience, we fix $w_s = 1$ and interpret w_o and w_i as an agent's SVO. The free parameters used by all three agents include w_o, w_i, the learning rate α, and the discount factor γ.

5.3.2 Deep Q-Network

Our first agent architecture is a deep Q-network (DQN), a three-layer feedforward network with rectified linear units. Networks like the DQN have previously been used to learn complex, humanlike behavior in multiplayer games [253]. The DQN agent represents s as a one-hot vector. The first layer contains 156 neurons (one neuron for each possible value of s), the middle layer contains 30 neurons, and the last layer contains one neuron for each possible action (11 or 31). Weights between neurons in each layer are initialized by sampling from a Gaussian distribution. To train the network, we record each transition (s, a, r, s') that the agent experienced, then do a batch update at the end of each episode. Loss is equal to the sum of ΔQ^2 for each transition (Eq. 5.1), and we use Adam backpropagation to update weights within the network. To facilitate comparisons between the learning trajectories of DQN agents and our other agents, we did not use an experience replay buffer; preliminary tests suggested that using a replay buffer made little difference in final performance. Note that we also ran some preliminary experiments on a tabular Q-learning agent, but found that it behaved similarly, so we only report results from the DQN. We also experimented with "deeper" Q-networks that included more layers, but we found that three layers was sufficient to learn the TG; our resulting network is therefore not particularly "deep", but we refer to it as a DQN to emphasize the connection between this architecture and deep learning.

5.3.3 Instance-Based Learner

Our second architecture is based on ACT-R [5], an integrated theory of cognition supported by an extensive history of cognitive models validated by human behavioral data. In this architecture, the retrieval and utilization of episodic memories is governed by instance-based learning (IBL) and blended retrieval [237], two mechanisms that model human memory and decision making [88] and have previously been used by cognitive agents to play social games [137]. Our instance-based learner (IBL) contains an episodic memory and a working memory, which work together via queries and retrievals to recall relevant pieces of information from previous experiences. These "chunks" contain information about the state s, the selected action a, the returned reward r, and the estimated value Q. Note that the storage of information in these chunks is perfect: each episodic memory contains the exact information used in the simulation. When choosing an action, the agent looks through all chunks in episodic memory and loads into working memory those chunks which satisfy two criteria: the chunk has sufficient activation due to recent or frequent use, and its state is sufficiently similar to the current state. The activation of a chunk in episodic memory A_c is given by the standard ACT-R equation:

$$A_c = X(\mu, \sigma) + \ln \sum_k t_k^{-d}, \tag{5.3}$$

where the summation is over all previous appearances k of the chunk, t_k is the amount of time elapsed since the k-th appearance of the chunk, d is a decay rate, and $X(\mu, \sigma)$ is a random number drawn from a logistic distribution with mean μ and scale σ. The similarity S_c between a recalled chunk c and the current state is zero unless they (a) took place on the same turn, and (b) had an identical number of available coins. Overall, a chunk is recalled if

$$\text{recall} = \begin{cases} \text{yes}, & \text{if } A_c > thr_A \text{ and } S_c > 0 \\ \text{no}, & \text{otherwise} \end{cases}, \tag{5.4}$$

where the activation threshold thr_A is a free parameter of the IBL agent.

The value of each potential action is calculated using blended retrieval,

$$\hat{Q}(a) = \sum_i^M Q_c(a) A_c, \tag{5.5}$$

that is, all chunks c where action a was taken are queried for their estimated value Q_c, weighted by their activation A_c, and summed. The Q value assigned to each new chunk is equal to the reward returned for that action plus the discounted expected value of "future"

chunks. To calculate this expectation, the agent recalls all chunks j with sufficient similarity to the game's *next* state s' and blends their values

$$Q_c = r + \gamma \sum_{j}^{M} Q_j A_j, \tag{5.6}$$

where the sum is over the chunks j that pass the activation and state similarity thresholds for s'. Conceptually, the IBL agent estimates the expected discounted value of a future state by averaging the values of states that followed from the current (s, a): to simulate imperfect recall, the agent only "envisions" future states that have sufficient activation and similarity to s'. Finally, the IBL agent selects the action with the maximum estimated value $\hat{Q}(a)$.

In our IBL implementation of Q-learning, the agent does not update the value of previous chunks when it receives feedback from the environment. Instead, whenever the agent takes an action, it forms a new chunk whose value is given by Eq. 5.6, then the blended values of old and new chunks are used to estimate the overall value of an action (Eq. 5.5). The result is that old memories retain their original (out-of-date) Q-values, but new (up-to-date) memories make a more significant contribution to $\hat{Q}(a)$ as the activation of older chunks decays to zero (Eq. 5.3). We discuss the implications of this mechanism in Sec. 5.6.2.

5.3.4 Neural Engineering Framework Agent

Our third architecture is a spiking neuron model built using the NEF. This network uses an online, error-driven learning rule to implement Q-learning within the network, short-term memories to recall previous states, and an independent-accumulator for action selection. In contrast to the DQN and IBL agents, which use external software and data structures to compute ΔQ and recall previous transitions, our goal with this agent was to simulate all the relevant computations and cognitive processes within the neural network itself. Broadly speaking, our network contains numerous populations networked with feedforward, lateral, and feedback connections. Activities within each of these populations dynamically represent quantities such as s, $Q(s)$, and ΔQ, while the connections between the populations compute functions such as the state-value mapping, WTA competition between candidate actions, and RPE. To manage these computations while dynamically playing the TG, the NEF agent uses a control structure involving several inhibitory gates and working memory buffers. We first describe the overall structure of the model (summarized in Figs. 5.2-5.3), then detail how neurons represent the required quantities and how connections realize cognitive control.

To begin, state input about the TG is encoded as a high-dimensional vector and passed to the *state* population, which represents $\hat{s}$; we describe this encoding scheme in detail below. The *state* neurons connect feedforward to a *value* population, which represents $\hat{Q}(s)$. The connection between *state* and *value* computes the value function, and must be updated using a RPE calculated according to Eq. 5.1. The *value* population connects to the *choice* network, which selects the best action using WTA dynamics. We use the independent accumulator from Fig. 1.4 for *choice* for this purpose: this network integrates the estimated value of each action over time until one hits threshold, then inhibits the values of all non-winning actions, producing a one-hot vector representing the model's choice. Note that we also tried using the BG network (Fig. 1.3) to implement action selection in our NEF agent, but found that its temporal dynamics and numerical precision were ill-suited for generating clean one-hot vectors on the relevant timescales. After making this choice, the NEF agent observes the reward returned by the environment, and calculates the error $\Delta\hat{Q}(s,a)$.

Calculating this error, and applying it to update the appropriate weights on the *state-value* connection, is challenging due to the passage of time. Eq. 5.1 requires that an agent observes the new state s' before updating $Q(s,a)$. At this point, s, a, and $Q(s,a)$ are no longer dynamically represented in the *state*, *value*, or *choice* populations. To accommodate this, we introduce several gated working memories to temporarily store these quantities. We use the gated working memory network from Fig. 1.2 to flexibly store and clear these values. To calculate $\Delta\hat{Q}(s,a)$, the error network recalls these values from the buffers, then uses the PES learning rule to update the decoders on the *state-value* connection. This update is also challenging because "indexing" is difficult in the NEF: to calculate the appropriate $\Delta\hat{Q}(s,a)$ and update only the decoders associated with representing s, we must transform the signals in several ways, described below. However, once the appropriate error signal is represented, it is straightforward to apply PES to update the network's value estimate.

To represent the state of the environment in our network, we translate our external input (round number and available coins) into a special high-dimensional vector called a spatial semantic pointer (SSP). SSPs were recently developed in the context of NEF networks as an encoding scheme to represent continuous variables, such as spatial position, audio frequency, or status in a social hierarchy [129, 127]. These vectors contain rich semantic information that can be symbolically manipulated using mathematical operations realizable with the NEF. We chose to represent s with SSPs for several reasons. First, SSPs effectively capture differences between states of the environment: for example, if an agent has 10 available coins in s_1, 9 available coins in s_2, and 1 available coin in s_3, the dot product (a common similarity metric) between the SSP representations of s_1 and s_2 will

be higher than the dot product between s_1 and s_3. This means that neural activities in the *state* population will faithfully encode these differences. When we decode value estimates from these activities, the differences between value estimates $\hat{Q}(s, a)$ will therefore smoothly reflect differences in the underlying states. This allows for a greater degree of generalization in our models than is possible with one-hot encoding, where all states are maximally dissimilar. Furthermore, we can use SSPs to tune the degree of generalization across states by (a) changing how s is encoded into the SSP, and (b) changing the tuning curves of neurons in the *state* population to control the degree of pattern separation (as we did in Ch. 4). The dimensionality of this SSP, the length scales used to encode state information, and the sparsity of the neurons in the *state* population, were all free parameters of the model.

Another advantage of using SSPs in our NEF model is that previous work has successfully applied SSPs to simple problems in RL, including spatial navigation in 2D and 3D environments [16, 128]. Compared to other encoding schemes, the symbolic advantages of SSPs mentioned above lead to noticeable improvements in the performance and speed of RL agents. Finally, SSP encoding is biologically plausible: when populations of spiking neurons represent SSPs, they exhibit patterns of activity that closely resemble cells in hippocampus, including place cells and grid cells [58]. Given that these hippocampal cells are implicated in pattern separation, mapping of continuous variables, and navigation through complex environments (both spatial and social), and given that other NEF models using SSPs and grid-cell-like representations have successfully solved navigation and planning problems [57], SSPs may capture the functional representations used in these parts of the brain.

During each round of the TG, the network cycles through three phases. These phases correspond to different dynamical states of the network, and help organize learning and action selection.

- During phase one, the agent perceives the current state of the environment s', and chooses an action a'. Rather than select the highest-value action from the output of *choice*, the network also considers the value of exploring other actions. We realize this by adding a vector N to $\hat{Q}(s')$, effectively boosting the value of some actions compared to others. In our simulations, we compute N externally: with probability ϵ, this is a random one-hot vector, and is otherwise the zero vector. The result is ϵ-greedy action selection: we feed $\hat{Q}(s') + N$ into *choice*, which selects the action with the greatest combined value, a'. The agent takes this action. The network also stores the expected future reward, $\hat{Q}(s', a')$, into WM_{value}.
- During phase two, the model calculates ΔQ and engages PES learning. To do this,

it recalls from WM the values of s and a, recalls $\hat{Q}(s', a')$, and views the rewards returned as a result of its previous action, $R(s, a)$. Using the recalled (s, a), the network then estimates the value of the previous state, $\hat{Q}(s, a)$. With all these quantities dynamically represented within the network, the model computes $\Delta\hat{Q}(s, a)$ according to Eq. 5.1. This error is passed through the PES learning rule, updating the decoders on the connection between *state* and *value*.

- During phase three, the network stores s' and a' in WM buffers. This step must happen after learning, since it overwrites the previously stored information s and a.

In order to progress through these phases during the course of a simulation, we drive the network with two external control signals, which we label *replay* and *buffer*.

- The *buffer* signal tells the network whether or not to replace the contents of WM buffers with new information. When this signal is turned off, neurons in the *diff* populations of the gated WM networks are inhibited, preventing any external information from affecting the stored representations. When this signal is turned on, inhibition is released, and the current contents of WM are replaced with new values. In phase one, *buffer* causes $\hat{Q}(s', a')$ to be saved in WM_{value}, and in phase three, *buffer* causes s' and a' to be store in WM_{state} and WM_{choice}.

- The *replay* signal switches the input to *state* between s' (the current external input) and s (the recalled previous state stored in WM_{state}). In phase two, *replay* prompts the network to compute $\Delta\hat{Q}(s, a)$ and disinhibits PES learning, causing update of the *state-value* connection. Otherwise, in phases one and three, *replay* inhibits learning.

Fig. 5.3 summarizes theses phases and learning signals with a detailed network architecture.

5.3.5 Human Experiment

As we discussed in Sec. 5.2.3, the TG has been widely studied using both empirical and computational approaches. However, the majority of this work is concerned with characterizing trust as a static quantity and relating this quantity to prosocial behavior. In contrast, we are interested in how individuals dynamically estimate the trustworthiness of others, how this relates to SVO, and how we can understand the process of trust formation through computational models of RL. To investigate these questions, we require data that describe how individual behavior *changes* over time and in response to external conditions. Datasets that include such information are rare, and are often focused around

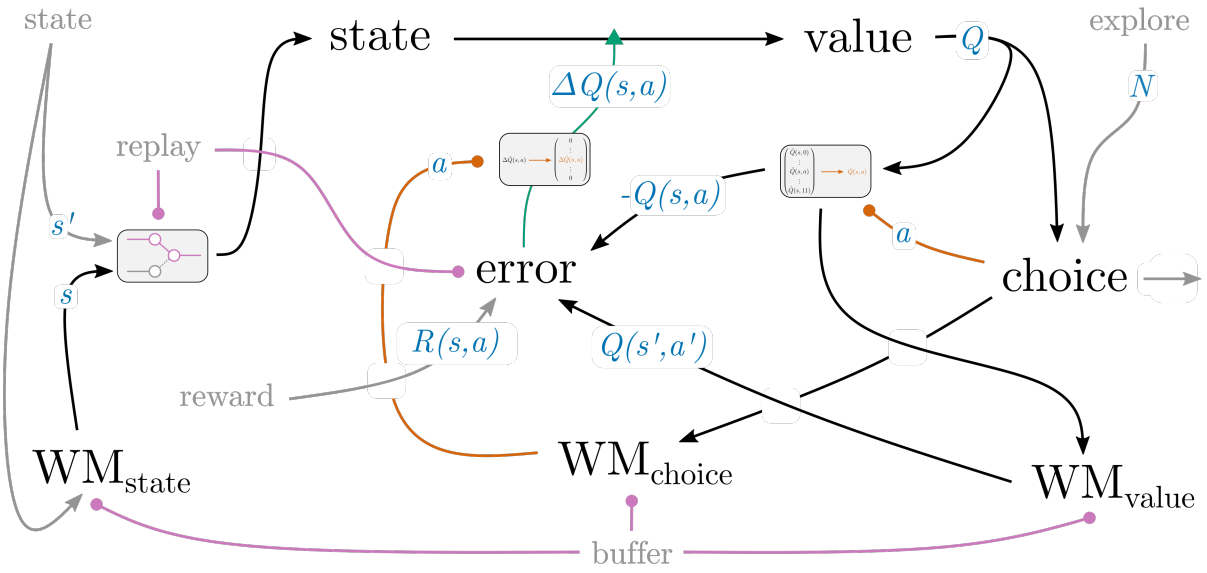

Figure 5.3: Detailed network architectures for the NEF agent. Grey text and arrows are external inputs, while black text and arrows are neural populations and synaptic connections. Colored arrows are specialized control operations: the green arrow implements PES learning; pink arrows inhibit postsynaptic neurons; and orange arrows index neural representations based on action choices. Specialized neural networks, indicated by gray boxes, facilitate these operations. Blue text labels the represented variables at key points in the network. WM populations use the architecture described in Fig. 1.2, and the *choice* population uses the architecture described in Fig. 1.4. The *state* population simulates neurons that are sensitive to each dimension of s, while the remaining populations represent each dimension of Q or a independently.

specific questions that are only tangentially related to our investigation. For example, [152] study how trust-related behavior changes in response to breaches of trust that occur early versus late in a repeated prisoner's dilemma, while [70] study how trust-related behavior changes over time in a repeated TG which has a definite versus indefinite end. While these data and conclusions are interesting, they cannot be used to validate our models.

In order to acquire validation data for our model, we ran a human experiment in which participants learned to play the TG against simulated opponents. Participants in the experiment completed the following steps.

1. Participants reviewed the information form, which detailed the goals of the study,

the requirements for participants, and the contact information for the study lead, supervisor, and ethics board. They then reviewed and signed a consent form.

2. Participants completed an interactive tutorial that introduced the rules and strategy of the TG.

3. Participants completed a pre-trial survey asking for age, gender, income, level of education, and previous experience with economic games. The survey also asked participants to rate how strongly they agreed or disagreed with several statements, designed to measure empathy, altruism, and risk-taking. All questions were multiple-choice, and answering was optional.

4. Participants played thirty games against computer opponents, described below. Fig. 5.1 depicts the web interface, and shows what information was visible to participants. Importantly, participants viewed the increases in score (and real money earned) following each move, providing direct feedback about the quality of their decisions.

5. Participants completed a post-trial survey that queried their beliefs about the objective of the game, the identity of the opponents, and the progress of their learning. All questions were multiple-choice, and answering was optional.

220 participants were recruited through Amazon Mechanical Turk; we made the study available to all MTurk users, without any constrains on their ratings, history, or location. Participants earned \$0.10 per game plus \$0.003 per coin they collected, incentivizing them to play strategically rather than quickly. During data analysis, we eliminated all participants who either (a) failed to complete all thirty games and answer the post-trial survey questions, or (b) failed to change their behavior (i.e., learn) over the course of the experiment; details are provided in Sec. 5.4.2. This resulted in a total of $N = 97$ participants in the final dataset. This study was reviewed and received ethics clearance through a University of Waterloo Research Ethics Committee (ORE#42531).

5.3.6 Participant SVO

During preliminary analyses, we found that a number of features identified in our surveys correlated with TG behavior, with varying degrees of statistical significance and interaction strengths. In this chapter, we focus on analyzing the effects of participant SVO; future work should investigate the influence of other demographic and personality factors. In the post-trial survey, participants were asked "What was your objective when playing the trust game?", then selected one of the following three responses:

- I tried to earn as many points for myself as possible, without considering my opponent's score
- I tried to achieve a high score for both myself and my opponent
- I chose behaviors at random, playing as quickly as possible

We used this answer to classify participants as "proself" or "prosocial"; this division constituted our first independent variable.

5.3.7 Simulated Opponents

To further investigate the relationship between SVO, learning, and prosocial behavior, we introduced a second independent variable: opponent strategy. Before participants started playing games, they were secretly sorted into two groups: the first group faced opponents who could profitably be exploited with a greedy strategy, while the other group faced opponents where the best strategy was to be consistently generous. We were curious whether proself and prosocial participants would learn different strategies against these two classes of opponents, and whether the dynamics of trust formation would differ across these groups. Initially, we were reluctant to use simulated opponents to investigate learning and behavior in the TG: previous studies have shown that humans are less likely to behave prosocially when playing against computer opponents with fixed strategies [170]. This tendency appears to arise from (a) a lack of empathy towards opponents who do not have "feelings", and (b) the inability to develop mutually-cooperative strategies with partners who behave according to fixed (non-adaptive) rules.

To minimize these antisocial effects and encourage natural levels of prosocial behavior, we strove to make our agents humanlike. Both our "greedy" and "generous" agents played according to an adaptive "Tit-for-Tat" (T4T) strategy: they increased the amount they transferred when the human was generous, and decreased it when the human was greedy. We parameterized our T4T agents such that "greedy" agents barely changed their behavior to reward or punish participant's choices, while "generous" agents strongly rewarded generous behavior and punished greedy behavior. Note that "greedy" and "generous" refer to the *human strategy* that can best exploit this agent, not the strategy of the agent itself. Agents also began each game with a random internal state, and had randomized response times. Alg. 1 gives the pseudocode for our T4T agents, and Table 5.1 gives the default parameter values for greedy and generous agents. Altogether, our agents exhibited (a) heterogeneous behaviors across turns and games and (b) exploitable versus cooperative

behaviors in the two experimental groups. A post-experiment survey revealed that 36% of participants believed they played against a human opponent in one or more games, although this metric was significantly higher (over 50%) in some experimental groups.

Algorithm 1: Pseudocode for T4T Agents. Agents begin each game with an initial state O. In each round of the TG, the agent assesses the human's generosity G, and updates its internal state S based on whether this generosity was lesser or greater than the generosity X that it considers "fair". The agent then updates its state by an amount that depends on the forgiveness parameter F (if the human was generous) or the punishment parameter P (if the human was greedy). The number of coins transferred is then calculated based on S and the available coins C.

Input : Available Coins C
Output: Number of coins to *give* and *keep*
initialize state $S = O$;
while *game has not ended* **do**
  opponent generosity $G = \frac{\text{give}}{\text{give+keep}}$;
  state change $\Delta = G - X$;
  if $\Delta > 0$: forgive $S \mathrel{+}= \Delta * F$;
  if $\Delta < 0$: punish $S \mathrel{+}= \Delta * P$;
  if *investor*: clip state $0 \leq S \leq 1$;
  if *trustee*: clip state $0 \leq S \leq 0.5$;
  give = integer $(C * S)$;
  keep = C − give;

5.4 Empirical Results

We begin by analyzing the human data. We first describe how human behavior changes over time as participants gain experience playing the TG, pointing out interesting examples of exploration, exploitation, and adopted strategies. We then analyze the distribution of final strategies adopted by our participants, paying particular attention to how these strategies differ with SVO and opponent strategy. We identify several high-level trends related to generosity, performance, and reciprocity that differ across groups; we later use these trends as validation targets for our simulated RL agents.

Table 5.1: T4T agent parameters for greedy and generous agents. The greedy investor began in a trusting state and barely punished human greediness. The greedy trustee began in a non-trusting state, and largely retained this state regardless of human behavior. The generous investor began with modest investments, increasing them significantly if the human trustee reciprocated and decreasing them significantly if the human trustee defected. The generous trustee began by keeping more than half of the investor's transfer, but transitioned to equal returns if the investor persisted with generous transfers.

	Greedy Investor	Greedy Trustee	Generous Investor	Generous Trustee
O – Initial State	0.8 - 1.0	0.1 - 0.3	0.6 - 0.8	0.3 - 0.5
X – Expected Generosity	0.5	0.5	0.5	0.5
F – Forgiveness	1.0	0.0 - 0.1	0.8 - 1.0	0.4 - 0.6
P – Punishment	0.1 - 0.3	0.2	1.0	1.0

5.4.1 Learning Trajectories

Participants exhibited a wide range of strategies when playing the TG, and these strategies developed in significantly different ways over the course of the 15 games each human played against the simulated opponents. In this section, we examine "learning trajectories", which plot two dependent variables as a function of the game number (i.e., their cumulative experience playing the TG). Our first dependent variable is score, or the number of coins participants earned in each round of the game (0-30 coins). Our second dependent variable is "generosity", which is a normalized measure (0-1) of coins transferred to the opponent on each round of the game:

$$G = \frac{C_{\text{give}}}{C_{\text{give}} + C_{\text{keep}}}, \tag{5.7}$$

where C_{give} is the number of coins transferred to the opponent, and C_{keep} is the number of coins kept. $G = 0$ is a greedy action, and $G = 1$ is a generous one. In the following figures, we plot a line displaying the mean score, and a histogram over generosity, as a function of experience. Learning is said to occur if participant's strategy converges over time (their generosity distribution narrows), or if their mean score increases (relative to its variance across rounds of the game).

Some participants exhibited little or no learning during the experiment: some people chose a strategy on the first game and repeated this strategy exactly for each subsequent game (Fig. 5.4), while the strategies and scores of others did not noticeably converge over 15

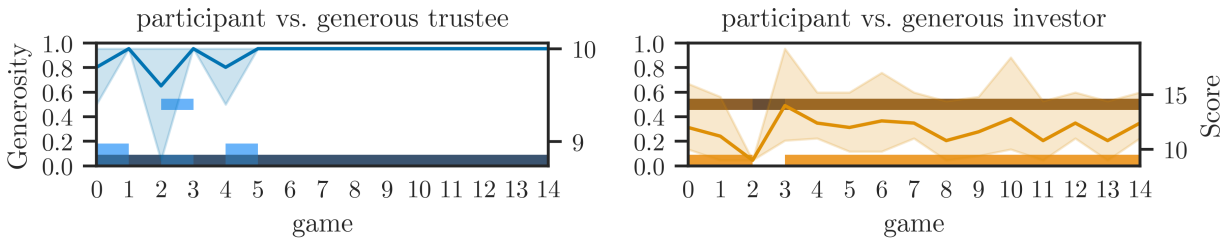

Figure 5.4: Example learning trajectory from a participant who adopts a strategy immediately and does not adapt it given external feedback. Score is plotted as a line with 95% confidence intervals, and generosity is plotted as a histogram across transfers in each game. The left panel (blue) shows games when the participant played as the investor, and the right panel (orange) shows games played as the trustee.

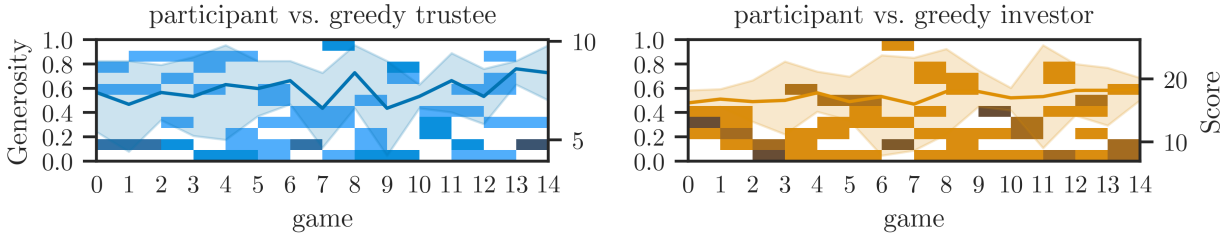

Figure 5.5: Example learning trajectory from a participant whose strategy does not converge over time. Such behavior may be random or reflect continuing exploration.

games (Fig. 5.5). Note that the web interface for the TG required participants to manually select how many coins they transferred on each game, so participants could not play quickly by repeatedly selecting a "default" option. Furthermore, different participants exhibited different "fixed" strategies, insofar as the number of coins they transferred was consistent across rounds and games, but different across participants. For "random" players, it is difficult to distinguish truly random strategies from strategies with sustained exploration: it is entirely possible that participants who did not converge after 15 games were still actively learning. In support of this hypothesis, some players only adopted a low-variance, high-scoring strategy in the final two or three games, indicating that the transition between exploration and exploitation may occur abruptly and towards the end of the experiment (Fig. 5.6).

In many cases, human learning in the TG is best characterized as (a) slow but consistent improvement and policy convergence, or (b) rapid convergence during the first few games, followed by occasional exploratory actions. Slow learning is most apparent when

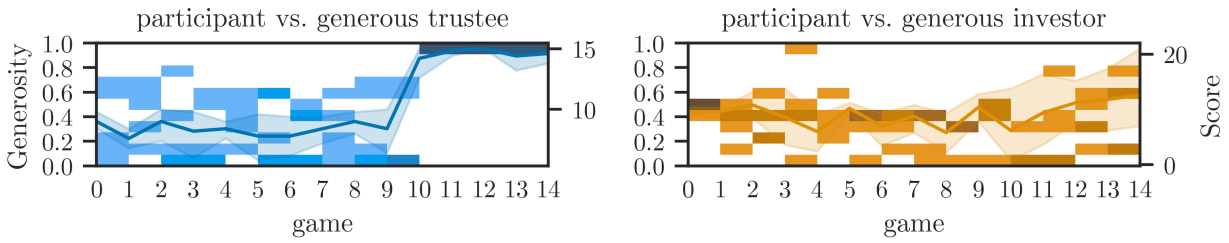

Figure 5.6: Example learning trajectory from a participant whose explores for most of the experiment, then abruptly finds, adopts, and maintains a high-performing strategy.

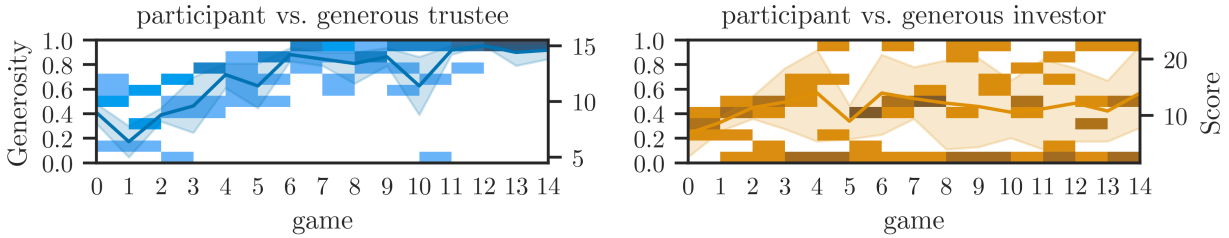

Figure 5.7: Example learning trajectory from a participant who gradually discovers a good strategy following systematic trial-and-error.

agents play the investor: in our TG against our simulated T4T agents, the optimal human strategy was to either invest everything (when paired with the generous trustee) or invest nothing (when paired with the greedy trustee). Many participants experimented with making exchanges of moderate generosity, and some gradually learned that extreme generosity (zero or one) lead to slightly better long-term outcomes. While Fig. 5.6 showed an abrupt transition after a gestalt realization near the end of the experiment, some participants exhibited a smoother learning trajectory (Fig. 5.7). In contrast, many participants exhibited rapid exploitation, converging to a largely-invariant strategy after just a few games (Fig. 5.8). For participants who adopted a high-scoring strategy early on, this exploitation lead to large cumulative rewards; however, in many (or most) cases, participants who exploited early settled on suboptimal strategies, such as investing nothing when playing a generous trustee. These two examples show that learning to play the TG against our adaptive T4T is not trivial, and that participants learn at different speeds with varying degrees of success.

Two other patterns are readily apparent from the learning trajectories of individual participants. First, many participants continued to make at least one generous investment every game, even when those investments were never repaid by the greedy opponent

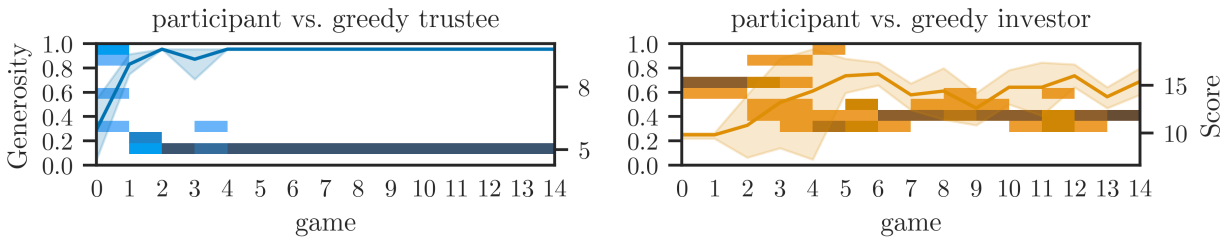

Figure 5.8: Example learning trajectory from a participant who rapidly settles on a strategy after minimal exploration. In this example, the participant's policy is slightly suboptimal.

(Fig. 5.9). Strategically, this behavior offers the opponent a chance for "redemption" by giving them the capital necessary for mutually-beneficial cooperation. Similarly, some individuals occasionally took a greedy action, to see if they could earn a few extra coins without incurring punishment. Second, only a fraction of participants discovered (or chose to adopt) a strategy we refer to as "late defection". This strategy takes advantage of the finite duration of the TG: on the fifth (final) round of the game, when playing as the trustee, a late defector will transfer zero coins back to the investor, keeping the full amount for themselves. An example of discovering late defection is shown in Fig. 5.10: the small amount of generosity density at $G = 0$ when playing as the trustee reflects a single defection on the final turn. This strategy is profitable because (a) the investor has no opportunity to punish this greedy behavior later in the game, and (b) in every game, participants face a new instance of the T4T agent, which has no memory of previous interactions. However, an individual might not adapt this strategy for several reasons: they may not discover it; they may choose not to adopt it out of a sense of fairness or equality for their opponent; or they may envision facing the same opponent in future games and being punished. In our TG experiment, all identities were anonymized, and T4T agents were coded to behave in a humanlike manner: this may have encouraged fair-minded or cautious behavior in some of our participants.

5.4.2 Eliminating Non-learners

Our investigation of participant learning trajectories revealed that a significant number of our participants did not change their behavior much over the course of the experiment. Because our computational models focus on the learning process, we wanted to eliminate these data from subsequent analyses. To do so, we used regression (Scipy's `linregress` function) to characterize the relationship between each participant's score and game number. This lead to a variety of fits: slopes ranged from positive (improved performance with

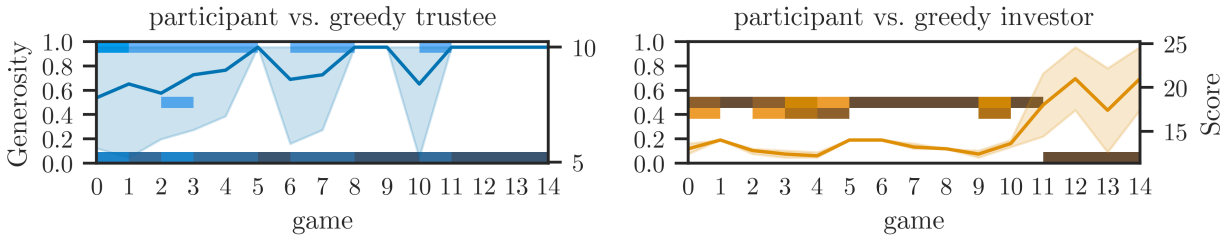

Figure 5.9: Example learning trajectory from a participant who perpetually offers the opponent a few generous transfers, even when this generosity is not reciprocated.

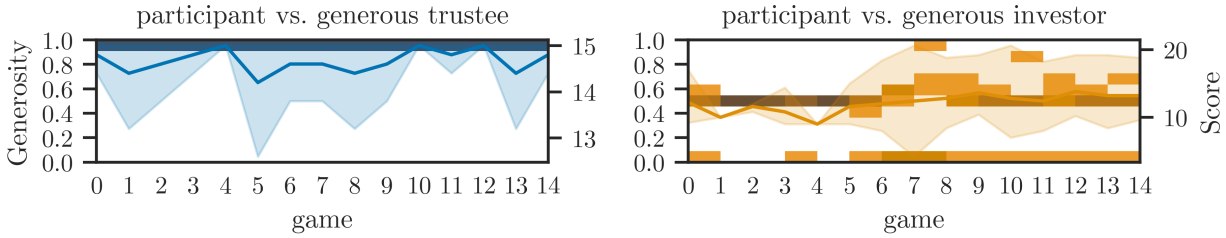

Figure 5.10: Example learning trajectory from a participant who discovers late defection when playing as the trustee, leading to large score increases after game five.

learning), to zero (fixed strategies), to negative (decreased performance after exploiting a bad strategy). After some experimentation, we decided to include only participants whose learning trajectories had positive slopes (greater than 0.05) and were statistically significant (at the $p < 0.1$ level). These criteria were fairly loose, and a number of questionable participants were still included in the final data. Surprisingly, this cleanup eliminated roughly half of all participants from our analysis, leaving 43 proself and 54 prosocial participants. For completeness, we repeated the analyses in the next two sections for the unrestricted data, and all the major trends held (in fact, several metrics were significant at higher confidence levels in the unrestricted data).

5.4.3 Final Strategies

Having characterized several interesting patterns in participant learning trajectories, and eliminated participants who were not fully engaged learning during the experiment, we can look more systematically at final strategies. We are particularly interested in whether these strategies differ between participants who faced greedy versus generous opponents, and between participants who we classified as proself versus prosocial. Fig. 5.11 com-

pares the generosity distribution of participants in the final three games across our four experimental conditions, and as a function of SVO. We first observed that participants who played against generous opponents learned different strategies than participants who played against greedy opponents: participants playing against generous opponents were, on average, more generous than participants who played against greedy opponents (when playing as either the investor or the trustee). This confirms that participants learned to use strategies that were adapted to their particular opponents.

Next, we compared the generosity distributions of proself and prosocial participants in each condition. Using independent t-tests to compare the means of the generosity distributions, we found statistically significant differences between the generosity of proself and prosocial participants in two of our four experimental conditions. Fig. 5.11 shows the results: SVO-related differences in generosity were significant when participants played against the generous trustee or the greedy investor, where generosity distributions resembled bimodal distributions. When playing against the greedy trustee, both proself and prosocial participants discovered that generous investments were selfishly exploited by the opponent: over 50% of participants in both groups eventually adopted a strategy where they kept all the available coins on each round. In contrast, both proself and prosocial participants struggled to find effective strategies when playing against a generous investor.

To see whether these strategic differences were associated with differences in performance, we performed the same analysis in Fig. 5.12, but replaced the dependent variable of generosity with coins earned. Interestingly, we found that proself individuals achieved higher average scores than prosocial individuals when playing against greedy opponents, but prosocial individuals achieved higher scores against generous opponents. Comparing the score and strategy distributions, we found that proself individuals were more likely to adopt high-scoring greedy strategies ($G \simeq 0$) against greedy opponents. In contrast, prosocial individuals were much more likely to adopt high-scoring generous strategies against cooperative opponents. One interpretation of this result is that prosocial players' commitment to exploring generous actions allowed them to discover that, although the first two investments of $G = 1$ against the cooperative opponent lead to scores of less than 15 coins, the final three investments of $G = 1$ lead to score of around 15 coins, beating out the purely greedy strategy that guaranteed 10 coins per turn. Fewer proself players made this discovery, so most adopted the purely greedy strategy, or explored intermediate strategies that were largely ineffective.

Finally, we wanted to quantify the amount of late defection in proself versus prosocial participants. To do this, we repeated the analysis in Fig. 5.11, but restricted the data to transfers made in the final turn when participants played the trustee. The results in Fig. 5.13 show that late defection is more prevalent in proself than prosocial participants,

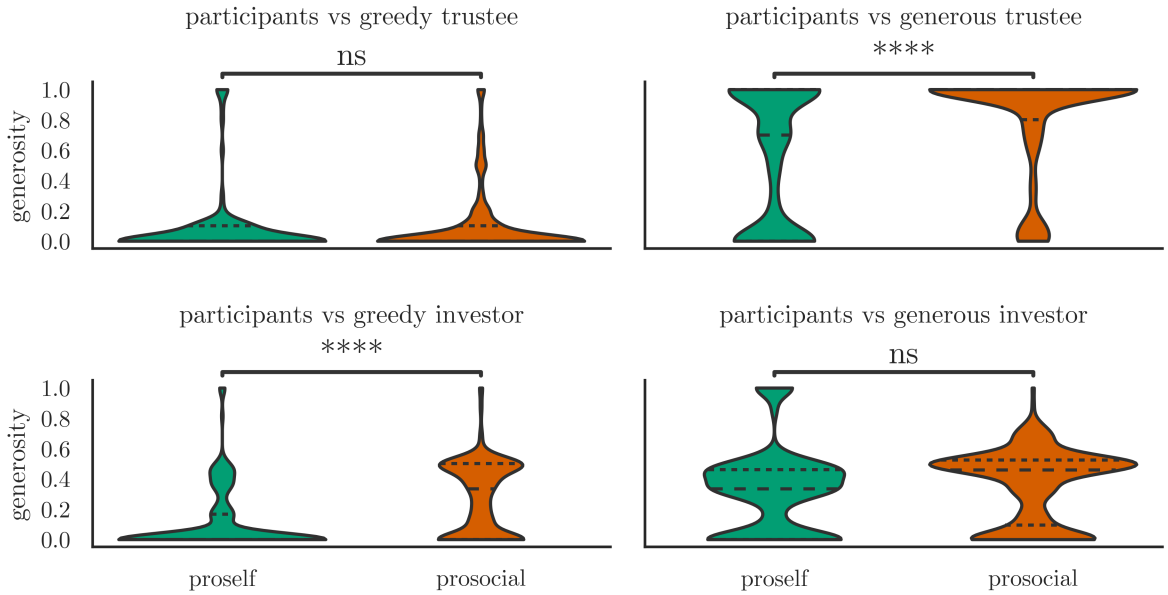

Figure 5.11: Generosity distributions of participants as a function of SVO. Each plot shows the distribution of generosities from the indicated population of participants in the final three games of the TG experiment. Dashed lines indicate populations means, and dotted lines indicate quartiles. In each of the four experimental conditions, we use a t-test to determine whether the population means are statistically different. We observe differences only in two of the conditions: prosocial participants are significantly more generous when playing against a greedy investor and against a generous trustee. Stars indicate statistical significance: (ns) corresponds to $p > 0.1$ and (****) corresponds to $p < 0.0001$.

but only when playing against the greedy investor; there was no statistically significant difference when playing against the generous investor. We were somewhat surprised that this trend was not evident in the latter case, but we suspect that the difficulty of this opponent, and the noisiness across games, made it difficult to discover the advantages of this strategy through trial-and-error.

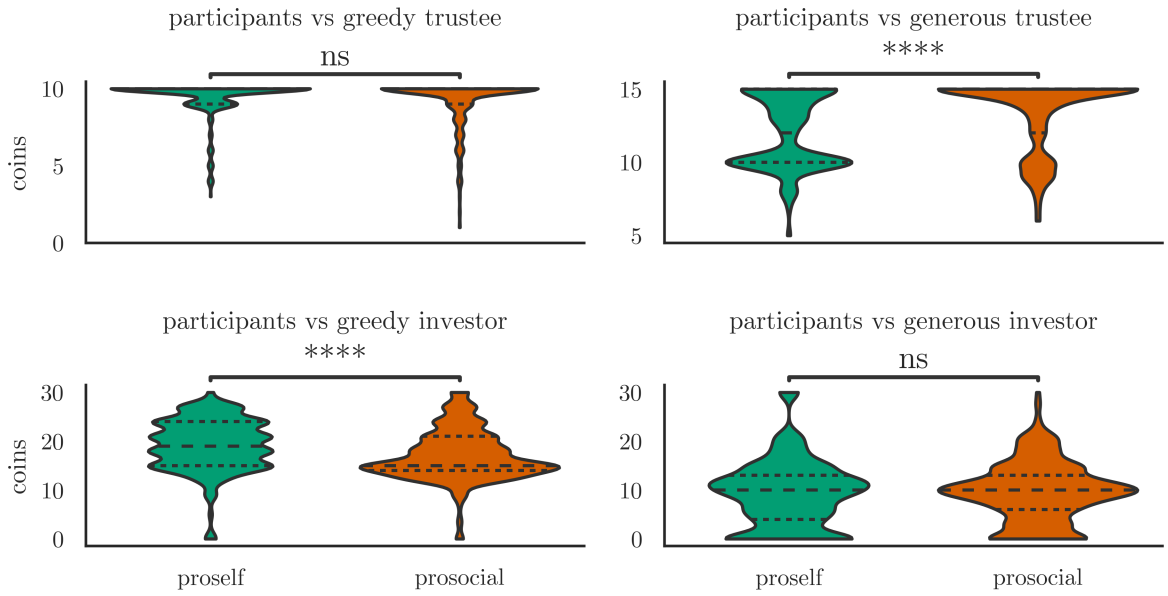

Figure 5.12: Score distributions of participants as a function of SVO. As in Fig. 5.11, we found that SVO only significantly affected participants' scores when playing against the generous trustee or the greedy investor.

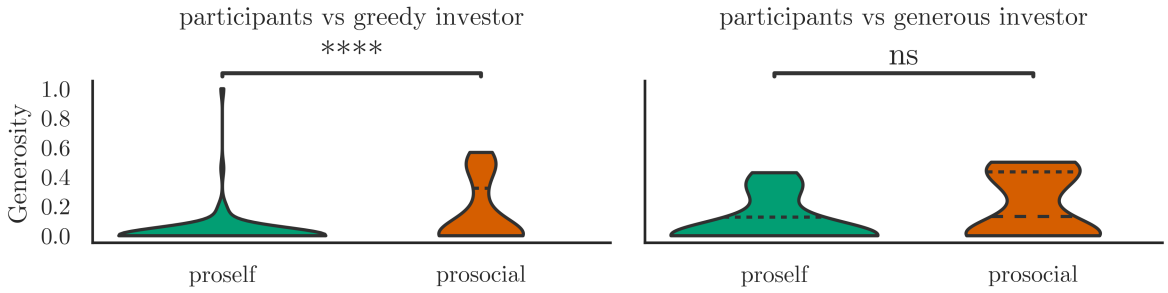

Figure 5.13: Generosity distributions showing late defection when playing as the trustee. Proself participants were more likely to keep all available coins on round five, while proself participants were more likely to maintain an egalitarian strategy on this round. This effect was only statistically significant when playing against a greedy investor.

5.5 Simulated Results

Our empirical study was successful in gathering data about human learning and behavior in the TG: we characterized a number of distinct learning trajectories and identified several interesting patterns in the final strategies adopted by participants. In particular, we showed that SVO has a significant effect on the strategies learned by human participants, in multiple experimental conditions and for multiple dependent variables. In this section, we simulate and train our cognitive agents with three objectives. First, we show that, under ideal conditions, agents may learn good strategies in the TG; this result establishes that our learning algorithms are working as intended despite the constraints imposed by each architecture. Second, we compare the learning trajectories of simulated agents to the empirical learning trajectories, and show that when agents face the same constraints as humans playing the TG, they exhibit similar patterns of behavior. Finally, we analyze these simulated data to identify high-level trends related to SVO, generosity, performance, and late-game defection.

5.5.1 RL Agents Learn Optimal Strategies

We begin our computational study by showing that agents from all three architectures may learn good strategies in the TG. Due to the complexity of the parameter space and the nonlinearities that emerge when we train RL agents against adaptive opponents, we do not attempt to systematically characterize the relationship between model parameters and agent behavior. Instead, we seek any set of parameters that produce coherent learning trajectories and high-scoring final policies. We chose the easiest experimental condition, in which agents played as the investor against a greedy trustee. In this condition, the optimal policy is to keep all available coins on every turn. Most of our participants adopted this strategy during the experiment, so discovering this policy is certainly an achievable goal.

Because RL algorithms often require substantial training data, we first trained our agents over the course of 150 games against the greedy trustee (10 times the number of games played by our human participants). Fig. 5.14 plots the learning trajectories for 20 agents from each architecture. The majority of agents converged to the optimal strategy: in the final games, most agents select the greedy action $G = 0$ on most turns, confirming that our RL agents can discover high-scoring policies under ideal conditions. Fig. 5.15 repeats this experiment, but limits agents to 15 training games, and enforces more biologically-realistic parameters in the agent architectures (e.g., higher memory decay in the IBL agent and SSP representations in the NEF agent). While the learning trajectories in this

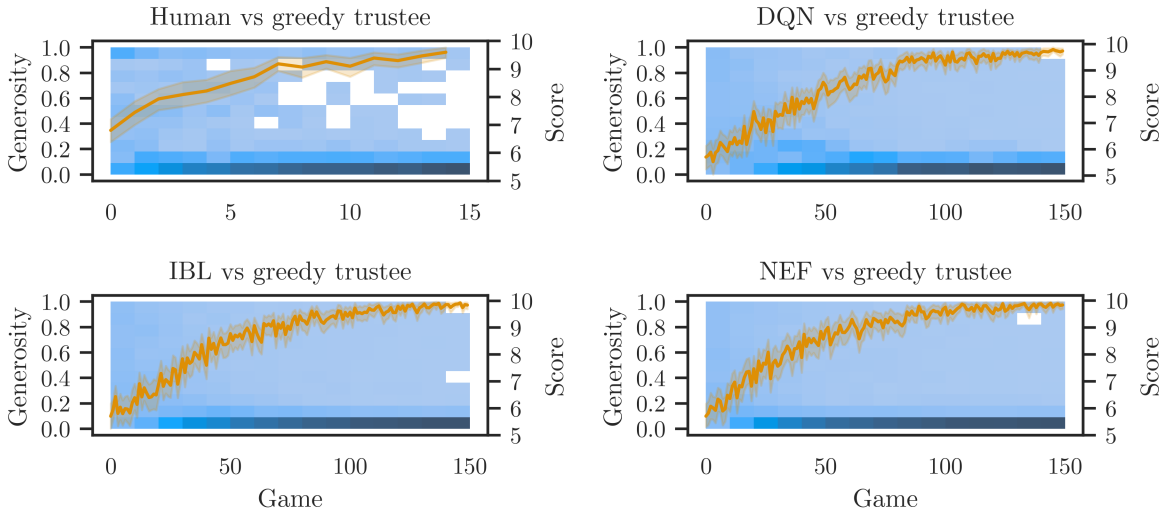

Figure 5.14: Learning trajectories for humans and agents playing against a greedy trustee. The optimal policy against this opponent is to invest zero coins on every turn. After playing 150 training games, most agents discover this strategy, confirming that RL can teach cognitive agents to play the TG under ideal conditions.

experiment still trend towards the optimal policy, agents do not have enough training data or cognitive capacity to reliably discover the best strategy, resulting in noisy and suboptimal policies in the final games. This result showcases the difficulty of using RL to learn the TG given real-world constraints.

5.5.2 Learning Trajectories under cognitive constraints

Do agent learning trajectories resemble those of our human participants? We investigated this question using two approaches. First, we selected two representative human trajectories from Sec. 5.4.1 as empirical targets, and used an optimization process to search for model parameters that reproduce this trajectory. The results are shown in Fig. 5.16. Comparing the simulated and target trajectories, we can see that agents may indeed recreate the learning histories of individual participants. However, additional investigation showed that these trajectories were strongly overfit to the target data: minor variation in agent parameters lead to poor fits and sometimes prevented learning entirely.

This overfitting motivated a second approach. We generated a population of agents

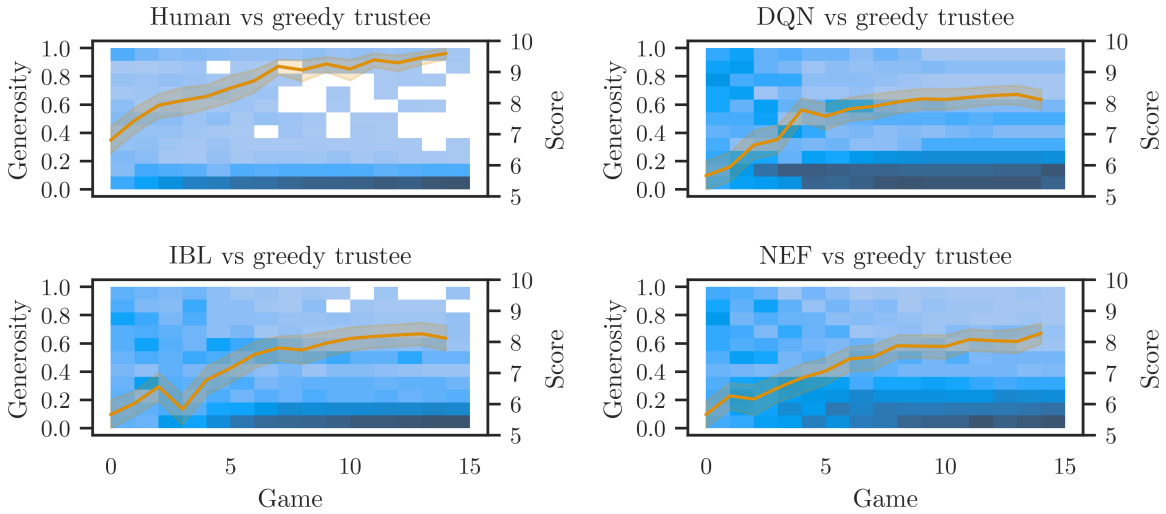

Figure 5.15: Learning trajectories for humans and agents playing against a greedy trustee. After playing 15 training games, only some agents discover the optimal strategy, while others learn slightly noisy, suboptimal strategies. This showcases the difficulty of using RL, specifically ϵ-greedy exploration, to learn the TG given limited training data.

from each architecture, and trained them to play the TG in the usual way. We then examined the learning trajectories of individual agents, and selected ones that qualitatively matched the types of behaviors shown in Sec. 5.4.1. Fig. 5.17 shows a few examples of agent behaviors from all three architectures; while these trajectories do not quantitatively match the behavior of any individual human, they do show that agent learning may converge quickly or slowly, to optimal or suboptimal solutions. We confirmed that these agents were not overfit to the training data by reinitializing individual agents and training them against a new set of T4T opponents. Overall, these results show that different classes of agent behavior arise from differences in RL and cognitive parameters, and that these behaviors overlap with the empirical data to some degree.

5.5.3 High-level Trends

Finally, we analyzed simulated behavior across the entire population of agents and compared these data to the empirical data. Again, we generated a population of unique agents for each cognitive architecture. For each agent, we drew parameters from random distribu-

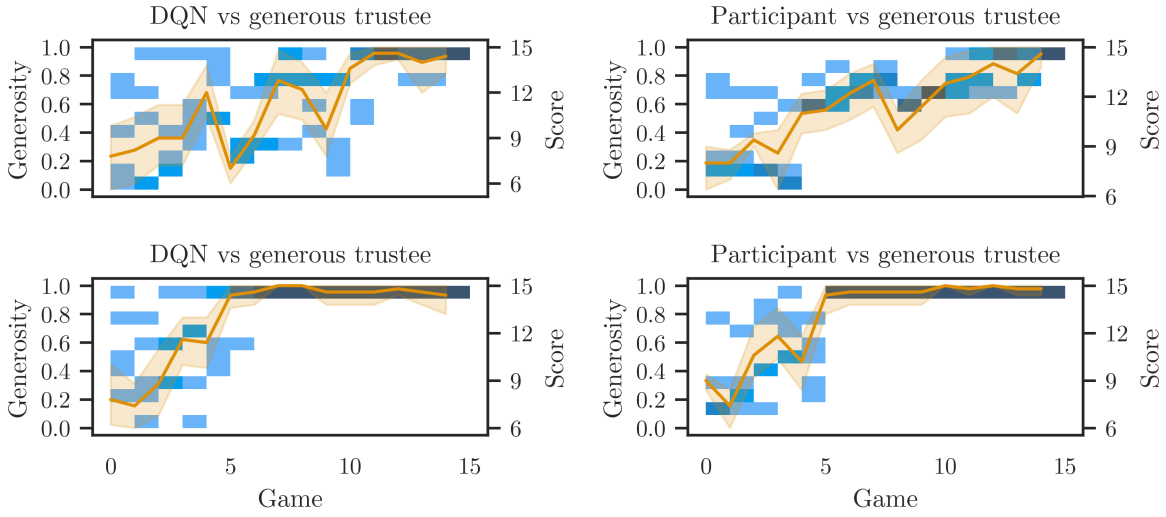

Figure 5.16: Learning trajectories for two DQN agents and the target empirical data. While our optimization produced behavior that closely resembled the human data, the resulting agents were overfit to the training data, and, when perturbed slightly or trained against a different opponent, did not reliably learn to play the TG.

tions that covered the range of cognitively plausible values: these included RL parameters (including γ, τ, and α), architecture parameters (number of neurons, size of the state space, memory decay, etc.), and SVO parameters (w_o and w_i). Each agent then played 15 games against the T4T opponents. We observed that, as with the human data, many individuals failed to learn over the course of the simulated experiment: some immediately adopted a policy and did not update it, while others never settled on a coherent policy. We therefore performed the same procedure as in Sec. 5.4.2 to eliminate non-learners from the agent population; this procedure removed about half of the simulated agents. Following this selection, we labelled the remaining agents as "proself" or "prosocial" based on their values of w_o and w_i: we chose an arbitrary threshold of $thr_{\text{svo}} = 0.3$, and any agent with $w_o + w_i < thr_{\text{svo}}$ was classified as proself, with the remainder were classified as prosocial. With these data and labels in hand, we proceeded with several high-level analyses of the simulated data.

First, we compared the generosities of proself and prosocial agents in the final three games of the experiment. Figs. 5.18-5.19 compare simulated generosities across agent architectures (and human participants) when playing against a greedy versus generous

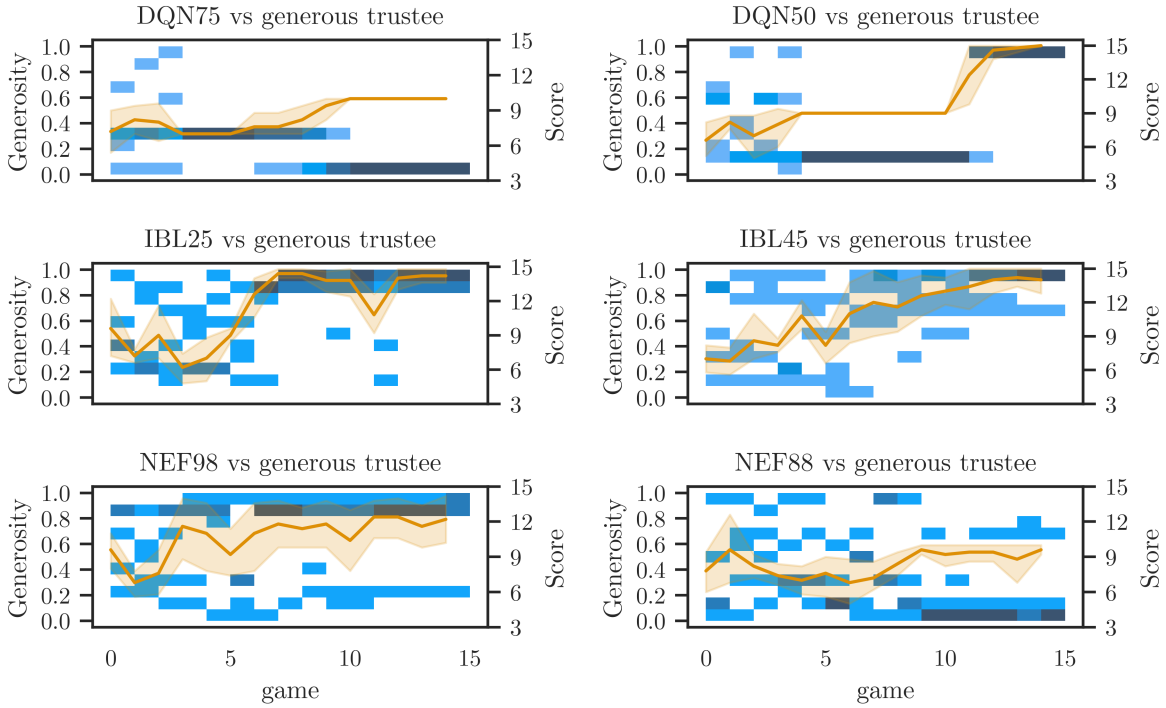

Figure 5.17: Sample learning trajectories from DQN, IBL, and NEF agents playing against a generous investor. We initialized a population of unique agents from each architecture, then trained each agent in the usual way. We observed many types of learning trajectories within each population, including fast versus slow learners, individuals who discover optimal versus inefficient strategies, and different degrees of noise. These patterns qualitatively match the dynamics of human learning we identified in Sec. 5.4.1.

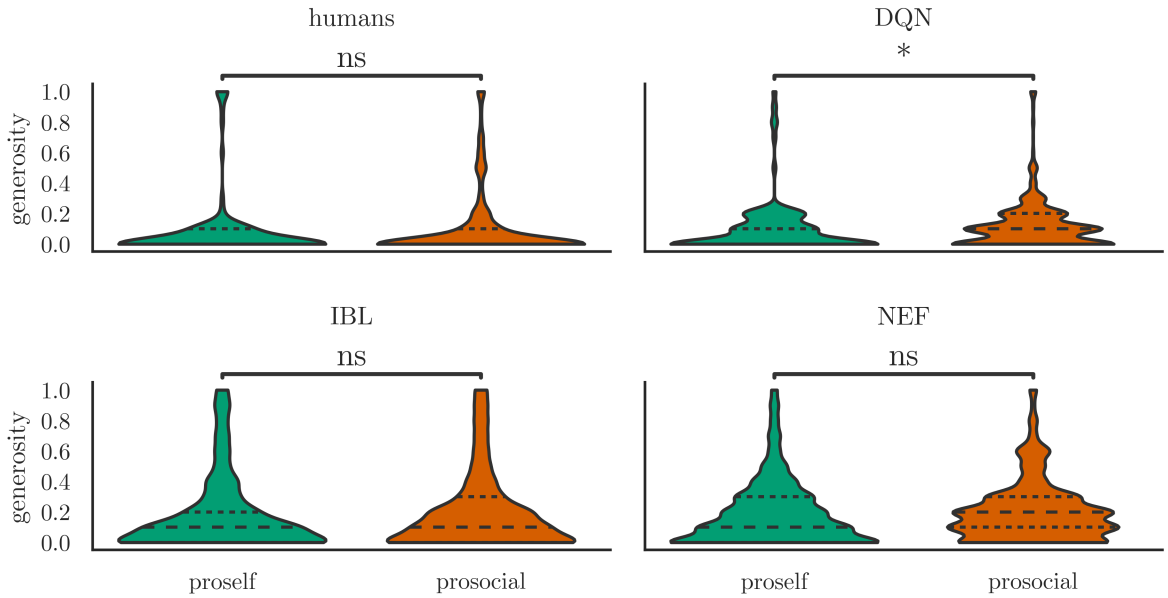

Figure 5.18: Agent generosity distributions when playing against greedy trustees. For human participants and all three agent architectures, SVO did not make a significant difference in participant mean generosity: individuals of both orientations learned to avoid exploitation by the opponent by keeping all the available coins.

trustee. The former plot shows that agents playing the greedy trustee predominantly discover the optimal strategy ($G = 0$): the distribution of transfers in the final policies are unimodal, with a peak near $G = 0$ that falls off quickly for larger G. Both proself and prosocial agents learn similar strategies, showing that SVO has little effect when playing against an opponent that consistently punishes generous behaviors. The human data are also unimodally distributed, and the difference between proself and prosocial humans was not statistically significant. In contrast, the latter plot shows that prosocial agents were more likely to discover a profitable, cooperative strategy against a generous trustee than were the proself agents. The shapes of agent generosity distributions from all three architectures are either unimodal or bimodal: proself agents sometimes adopt generous strategies but often become stuck playing suboptimal greedy stategies, leading to peaks near $G = 0$ and $G = 1$. However, prosocial agents are more likely to learn generous strategies, leading to a greater density of behaviors near $G = 1$ and relatively few greedy transfers. These data are also consistent with the bimodally-distributed human data:

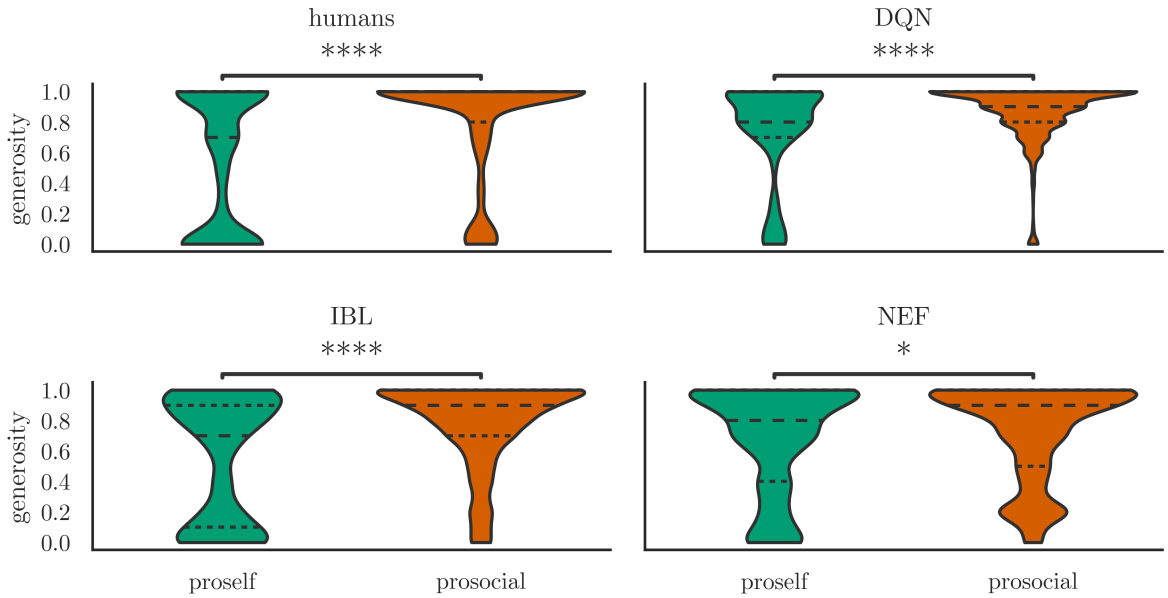

Figure 5.19: Agent generosity distributions against generous trustee. For human participants and all three agent architectures, SVO made a significant difference in participant mean generosity: prosocial individuals were more generous on average than their proself counterparts.

participant SVO influences the probability that an individual tends towards greediness or generosity, with proself participants more likely to choose $G = 0$.

To generate the agent populations described above, we performed some modest parameter optimization. During preliminary testing, we chose the ranges for the randomly chosen agent parameters by hand (e.g., the mean and variance of a normal distribution of γ). We found that, for most of our simulations, agent SVO had the expected effect: prosocial agents were more generous than proself agents (or mean generosity was statistically indistinguishable). However, arbitrarily choosing the range of random parameters often led to final generosity distributions that did not resemble the empirical data. This was not surprising, given that we have no prior knowledge about the variance in parameters like learning rate or state dimensionality within the human population. We addressed this uncertainty by optimizing the range of certain random parameters, namely α, γ, τ, architecture parameters, and thr_{svo}. In the optimization procedure, we generated a population of agents according to the range of random parameters for that trial, trained them

as normal, and labelled them according to thr_{svo}. We then measured the similarity between the final generosity distribution of proself agents and humans by calculating the KS divergence; we repeated this for prosocial agents and humans. Using this procedure, we were able to find population parameters that more closely reproduced the distribution of human behaviors in the final games.

Finally, we trained agents from our three architectures to play as the trustee. As evidenced by the human data, and by the larger state- and action-space in this condition, playing the TG as the trustee is significantly more challenging than playing as the investor. Our agents often struggled to learn effective strategies, especially when we applied cognitive constraints to agent parameters. Our preliminary tests showed that agent policies rarely converged after only 15 games, so to make comparison to the human data more interesting, we trained agents over 150 games instead, then compared the distribution of generosities in the final three games as usual. We were especially interested in looking at whether agent SVO would produce differences in late defection. Fig. 5.20 shows the generosity distributions of humans and agents on the final round of the TG, separated according to SVO. As expected, proself agents were more likely to keep all the available coins on the final round for a larger payoff, while prosocial agents were more likely to continue their generous behaviors on the final round. These trends are qualitatively consistent with the human data. However, we found that humans shifted their behavior by a greater amount on the final turn than did any of the agents.

5.6 Discussion

In this chapter, we studied how humans use external feedback when learning to play a simple social game, the trust game (TG). Our goal was to incorporate RL into existing cognitive architectures and show that the resulting agents exhibit humanlike learning and behavior in the TG. We chose three architectures for this investigation: deep neural networks, the ACT-R framework, and the NEF. Incorporating RL into each architecture posed a number of unique challenges, and resulted in agents with significantly different cognitive mechanisms for representation, evaluation, action selection, and learning. Nonetheless, we showed that all three agents were capable of learning to play the TG, with varying degrees of success based on the difficulty of the opponent and the number of cognitive constraints we enforced.

To investigate whether agent behavior aligned with human behavior, we also designed and ran an experiment with human participants, who played the TG online under the same experimental conditions as our RL agents. Analyzing the human data, we identified a few

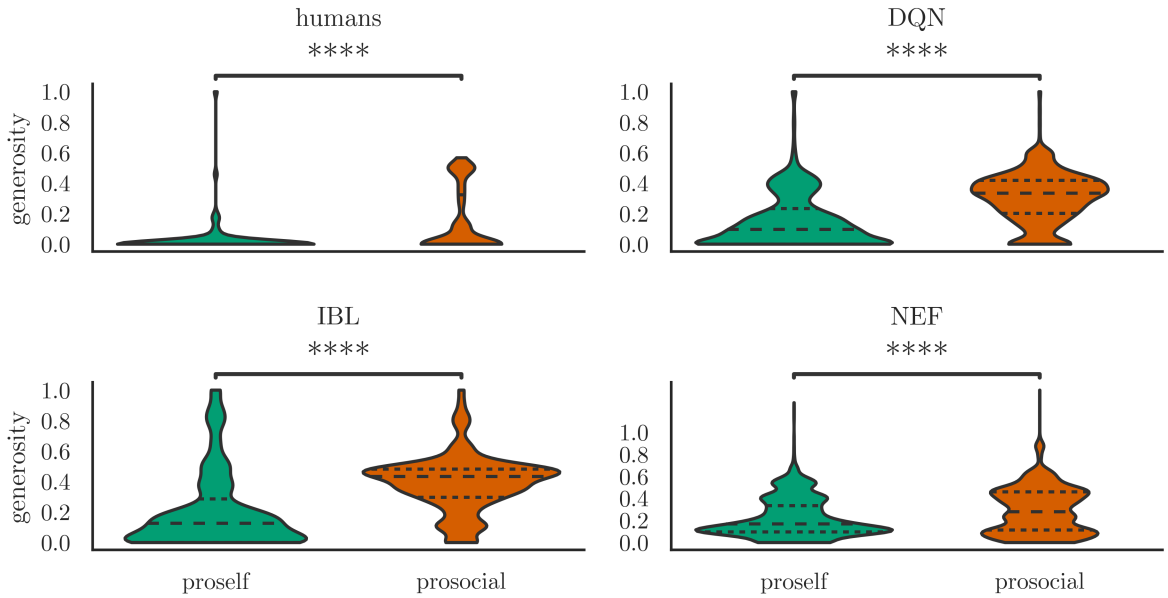

Figure 5.20: Generosity distributions showing late defection when playing as the trustee. Proself individuals transfered fewer coins on round five, while proself participants were more likely to maintain an egalitarian strategy on this round. However, human strategies on the final round were noticeably different than their strategies in the first four rounds, while agent strategies changed less noticeably (not shown here).

typical learning trajectories and established several high-level trends across the population. These became our empirical targets for model validation. One particularly significant trend was the distinction between proself and prosocial individuals: the former self-identified as seeking to maximize their own TG rewards regardless of the rewards earned by the opponent; while the latter self-identified as seeking to earn high TG rewards for both themselves and the opponent. Previous research in social cognition has identified SVO as an important factor in social decision making, so we decided to also incorporate SVO into our RL agents. We accomplished this in an architecture-independent way by modifying agent reward functions to include two additional terms: one that rewarded the agent when their opponent earned high rewards; and another that punished agents when the different between the two players' rewards was large. The weight that an individual agent placed on these two terms was used to classify the agent as proself or prosocial, allowing us to analyze the effects of SVO within the agent population.

In this section, we first analyze the learning and behavior of each agent architecture separately, discussing their cognitive realism and empirical validation. We then discuss the challenges of incorporating RL (and SVO) into cognitive architectures more broadly, identify shortcomings in our approach, and present avenues for future research.

5.6.1 DQN

The DQN is a standard feedforward neural network trained with backpropagation, in which the error term used to calculate the gradient is given by the Q-learning formula. This agent is simply a neural network that evaluates candidate actions given the current state of the environment as input. While this model is realized by "neurons", the neural realism of the model is minimal, as it relies on several highly-engineered representations: the input is a one-hot vector, where each possible state of the environment is a unique vector that is orthogonal to every other state; and the last layer of the DQN contains one neuron for each possible action. As backpropagation trains weights in the hidden layers of the network, different states of the environment activate each neuron in the output layer to a different degree. To make a selection, we compare the activation of each output neuron, and select the one with maximal activity to determine the action taken; during exploration, we select a random action instead. We externally calculate the error according to Eq. 5.1, square this term to compute the loss, and backpropagate this value through the network. Importantly, action selection and error computation occur *outside* the network; because these processes are not subject to representational or mnemonic constraints, they should be considered cognitively implausible. Similarly, backprograpation is not a biologically-plausible learning rule. Overall, this agent is the least cognitively plausible, but still represents an important class of cognitive architecture that is often used to approximate human decision making and play social games [253].

The learning trajectories of DQN agents bore several important similarities to those of human participants. We observed that simulated trajectories frequently had a short exploratory phase, followed by an abrupt transition to a final policy, which they maintained with little variance through the remainder of the experiment. Several human participants had similarly abrupt transitions to their final policies. However, a closer inspection of the human data revealed that the suboptimal policies adopted by some participants often consisted of optimal transfer on some turns and occasional random, exploratory, or forgiving transfers. In contrast, DQN suboptimal policies were usually invariant across turns, but the transfers were often one or two coins away from the optimal transfer against that opponent. Thus, while the distribution of human and DQN transfers across turns within a population often looked similar to the human data, there were important low-level features

of the data that the DQN agent did not reproduce. Note that the abrupt convergence of DQN policies was often driven by high rates of exploration decay τ, which our optimization procedures frequently selected when trying to match DQN data to the human data. Future work should investigate to what extent the DQN architecture itself favors abrupt transitions, versus particular parameter settings.

The DQN agent could be extended by adding additional components like RNNs or LSTMs, or by using more advanced architectures like transformers and convnets. These extensions would allow the network to parse the input space in different ways and to store information about previous rounds of the game, allowing learned strategies to potentially depend on sequences of actions taken by the agent and the opponent. While these components would likely lead to higher-scoring policies, the neural and cognitive realism of DNNs is still suspect, and this architecture may do little to inform our understanding of how brains learn to make social decisions.

5.6.2 IBL

Our ACT-R agent uses an instance-based learning system (IBL) to store, retrieve, and use information in working memory and episodic memory. Several equations govern how these information "chunks" are manipulated, and apply mnemonic constraints that make the IBL agent more cognitively plausible. While the agent does not use neurons to represent chunks, it accounts for mnemonic effects at high level of abstraction. The probability of retrieving a chunk from episodic memory depends on the recency and frequency with which it has been activated in the recent past: this captures well-documented relationships between memory usage and ease of recall. The influence of a recalled chunk also depends on this ease of recall: when multiple chunks that fit the current context are retrieved, those with greater activation are given greater weight, then all chunks are averaged, or "blended", together in a weighted fashion. This reinforces the recency effect, such that the IBL agent's decisions are most strongly influenced by the actions it has taken most recently, but will generally be influenced by actions that have been rewarded through its entire history.

We faced two major decisions when designing the IBL agent: what information to store in memory chunks; how to implement RL within the architecture. We chose to have chunks store the state information directly. Unlike the one-hot and SSP representations, this gave the IBL agent direct access to the relevant state information. Chunks also recorded the chosen action and the immediate reward. Finally, the chunk needed to store some information about the value of (s, a). Although many IBL models use the immediate

reward $R(s, a)$ to approximate the value of (s, a), we wanted to extend this representation to include the discounted value of future states, as is required for Q-learning. We therefore chose to estimate $Q(s, a)$ for each chunk. This required a more complex update than is typically used in IBL models. Unlike for the DQN and NEF agents, the IBL agent does not have a well-defined Q-function for all (s, a), so it was not obvious how to calculate $\max_{a^*} Q(s', a^*)$. Conceptually, we wanted the IBL agent to estimate the value it could obtain in states that followed from the current (s, a). We therefore designed a novel blending mechanism for estimating the value of future states that was inspired by the canonical retrieval mechanism in IBL. In this update, agents wait until they observe s', then retrieve all chunks similar to s', blending their values by their activations to estimate the expected future value. Unlike Eq. 5.1, which uses a maximum over all future values, the resulting update is a weighted sum over all chunks that were similar to s'.

The learning trajectories of IBL agents were more chaotic than the DQN agents: IBL agents would continue to select seemingly-random actions even late into the experiment. While these behaviors were of course influenced by the exploration decay rate τ (which tended to be lower for IBL agents than DQN agents), they also arose from the noisy mnemonic properties of episodic retrieval in IBL. Depending on the mnemonic noise σ, decay rate d, and activation threshold thr_A, IBL agents would sometimes fail to retrieve a complete list of memories about past instances of the current state from episodic memory. This would skew the estimated value of candidate actions and cause the agent to repeat actions that were somewhat successful at some point in the past, while ignoring actions that were reliably and highly successful in the past. While this mnemonic performance may seem to fall well-below human standards, the learning trajectories of IBL agents did resemble some of the slow-learning human participants. These individuals took many seemingly-random actions late in the experiment, and occasionally repeated actions they had taken in the past, despite having discovered higher-scoring actions in the subsequent turns. The chaotic learning trajectories of IBL agents thus provide an alternate explanation for the dynamics of learning observed in humans.

Among a population of unique IBL agents, the learned distribution of generosities was also consistent with the human data. As with the DQN agent, both proself and prosocial IBL agents learned to transfer few coins to a greedy trustee; SVO did not make a significant difference across the IBL population. Similarly, proself IBL agents were less generous when playing against the generous trustee than their prosocial counterparts, and they were more likely to engage in late defection when playing as the trustee. In general, we observed less convergence within IBL populations than we did for DQN populations; a higher fraction of IBL agents adopted suboptimal policies in every condition we examined. In some conditions, these broader generosity distributions were consistent with the human

data: for instance, a sizeable fraction of proself IBL agents maintained a purely greedy strategy against the generous trustee, a figure that better reproduced the human data than the corresponding DQN population. In other conditions, these broad generosity distributions poorly reproduced the human data: when playing the greedy trustee, many IBL agents continued giving generously, whereas relatively few DQN agents and humans did so. Overall, the cognitive realism of the IBL agents seemed to recreate several classes of mistakes made by human participants, but the extent of these errors did not quantitatively match the human data in many conditions.

The psychological realism of the IBL agent could be improved in several ways. First, information storage in memory chunks could be made more sophisticated. We used a highly engineered representation that does not generalize to other tasks; future work should experiment with more cognitively-plausible representations, especially ones which allow the agent to naturally and flexibly compare the similarity of states (such as SSPs). Second, alternative mechanisms for computing activation and blending memory chunks should be explored. Our agents were limited in their ability to update existing chunks: because each chunk was associated with states, actions, next states, and values, it was unclear how to update a chunk if a new chunk matched with some, but not all, of these components. To avoid these difficulties, we simply created a new chunk on every round of every game. Unfortunately, this somewhat defeated the purpose of the activation equation (Eq. 5.3), which assumes that a chunk will be periodically reactivated throughout an episode, leading to multiple triggers that increase the likelihood of future recall. As a result, the retrieval probability of our chunks depended solely on how recently they had been formed (plus activation noise). Furthermore, the time scale of our simulations was ambiguous: while Eq. 5.3 is typically applied for an agent performing a task with a millisecond timescale, our simulations were turn-based, and hence had no natural timescale; this made specification of the parameters d and thr_A somewhat arbitrary. Future work should explore more parsimonious mechanisms for chunk reactivation and retrieval. Finally, we proposed one mechanism for implementing Q-learning within IBL, but many other systems may be used to estimate the expected value of future states, and to calculate the blended value of potential actions accordingly.

5.6.3 NEF

Our NEF agent dynamically realizes state perception, value estimation, action selection, and value updating with a network of LIF neurons. We encode the current state as an SSP using a preprocessing step (external to the network), then feed this vector into the state population. The connection between the *state* population and the *value* population

computed the Q-function: these connection weights were updated by the PES learning rule, allowing feedback to change the state-value mapping such that the representation in *value* gave better estimates of immediate reward and expected future value. These estimates were sent from *value* to *choice*, where WTA action selection occurred using an independent accumulator network; this network was also provided with an external input that prompted exploration. After implementing the chosen action, the agent received feedback about the associated reward, then recalled the previous state and previous action, calculated the error, generated an error signal, and transmitting that error signal to the appropriate synapses on the *state-value* connection.

Despite the complexity of the network, and the noise introduced by neural representation, imperfect memory, and dynamic switching of network states, the NEF agent performed fairly well. Against the easiest opponent (the greedy trustee), the NEF agent reliably learned policies that approached the optimal strategy ($G = 0$), especially when trained over 150 games rather than 15. As with the DQN and IBL agents, learning trajectories from a population of NEF agents were quite diverse, depending on the parameters chosen and the course of random exploration. While we observed NEF learning trajectories with both fast and slow convergence, and with humanlike patterns of noise, we also noticed that NEF agents were more likely to become fixated on a single suboptimal action than were the other agents (see the bottom panels of Fig. 5.17). Examining the neural representations $\hat{Q}(s, a)$ over the course of the experiment, we concluded that this behavior arose from uneven neural representation across the input space. When initializing the *state* population, we did our best to distribute neural encoders so that all regions of the input space were accurately represented. However, due to the sparsity constraints we applied to neural representation, and due to the overlapping nature of SSP representation, some regions of the space invariably activated a larger number of neurons than other regions. When the PES learning rule updated the decoders associated with this region of space, a greater number of neurons had their weights increase, leading to an abnormally large increase in the value of $\hat{Q}$ for the associated (s, a). When this occured early during learning, exploration of alternative actions could rectify this error, normalizing the Q values back to a reasonable level. However, when agents only had 15 games worth of training experience, these abberant updates often persisted until the end of the experiment, causing the NEF agent to select the same action over and over, reinforcing the error. We did not observe this pattern of error within the learning trajectories of human participants.

The effect of SVO on NEF agents was similar to that of other agents. Prosocial agents discovered greedy policies against greedy trustees with the same regularity as proself agents, but prosocial agents were more likely to make generous transfers to a generous trustee than were their proself counterparts. Prosocial NEF trustees were also less likely to defect on

the final round of the TG than were proself agents. Overall, the distribution of behaviors among populations of NEF agents was broader than the human data, as expected given the prevalance of noise in this representation.

Of our three agents, the NEF agent is the most biologically and cognitively plausible. First, it acts in real time: each phase of the simulation lasts 100ms, during which time neural dynamics continuously drive changes in the represented variables. While we do not have specific evidence for how long it takes the brain to evaluate potential actions, choose a high-value action, buffer variables in working memory, and perform updates based on RPEs, figures between 100ms and 1s seem appropriate. Second, all operations performed by the agent take place directly within the network, including the storage and retrieval of memories about past states, the selection of the best action, and the online updating of connection weights. Third, the representations used, and the tuning curves of neurons in the *state* population, are consistent with biological representations of continuous feature spaces, such as the encoding of 2D space in the hippocampus. Fourth, the architecture of the network is not engineered specifically for the TG: it is a general system that uses RL to learn mappings between input states, value estimates, and selected actions, and could be applied to other tasks without modification.

The complexity and plausibility of the NEF agent introduced many challenges that should be addressed in future work. The network relies on external signals and phased operations to manage online RL. We imagine that, in biological brains, the signals instructing the network to, for example, overwrite information in a memory buffer, are generated according to specific processes in separate regions. For instance, it may be that executive areas in the PFC monitor various internal states of the brain, and modulate the networks we simulated according to higher-order schedules. Future versions of our agent could incorporate additional levels of monitoring and control. It is also not clear how the brain deals with the temporal delay between taking an action and observing its consequence: our agent relies on a fixed temporal schedule between action and reward to manage learning, but such a schedule does not exist in most real world tasks, a constraint which again motivates the existence of executive systems for monitoring and control. It is even possible that the brain computes RPEs without having to explicitly recall past states: the multiphase design of our NEF agent was quite cumbersome, and future work should investigate alternative means of updating connection weights that require less complexity.

Other neural models have addressed the issue of using online learning to change the value of previously-visited states in several ways, but each of these models has its drawbacks. For instance, in another NEF model of RL, [188, 189] used a parallel stream that computes Q in a second *value* population, which is fed with information about the previous state, while the main *value* population receives the current state input. With this setup,

the network is capable of computing all the quantities required for Eq. 5.1 in a single step, eliminating the need for multiple phases. However, the downside is that this network requires both *value* populations to be identical, which requires initializing the network in a certain way, and transferring weights between the two populations during the simulation. The authors managed to achieve this using a secondary learning rule, and were able to solve hierarchical navigation problems with this system, demonstrating its utility. In yet another NEF model of RL, [157] used a Legendre Delay Network (LDN), a different class of working memory network that stores rich information about windows of past history, to facilitate Q-learning. Specifically, the authors use LDNs to remember the neural activities associated with previous states of the environment: with these activities readily available, the network can update the appropriate connection weights as soon as feedback from the environment is received, without performing a secondary learning phase. However, this system requires the network to remember a vector encoding the neural activities of an entire neural population, which is a somewhat nonintuitive biological representation; it is unclear how well this representation would scale to larger networks. While the authors have shown that these networks may solve several classic problems in RL, many of the operations required are performed outside the network, notably action selection and error computation. Overall, our multi-phase approach is an interesting alternative to these two NEF networks for RL, and we hope that future work combines the insights from each approach to produce a more efficient, powerful system.

Another major challenge with the NEF agent is dealing with noise. This is an important challenge to overcome for any agent that claims to be biologically or cognitively plausible, and our model was modestly successful in this respect. However, we had to take several shortcuts to achieve this success, and these shortcuts limit the plausibility and scalability of our agent. First, unlike in previous chapters, we used rate-based LIF neurons instead of spiking neurons; this was done purely to reduce noise within the network. Second, in most of our populations, neurons were tuned to be sensitive to only a single dimension of the represented state. This greatly improved the accuracy of our memory buffers, independent accumulators, and specialized networks that changed the dimensionality of signals for computing error. While this further reduced noise in the network, it reduced the network's biological plausibility, as real neurons are usually sensitive to multiple dimensions of complex features. Third, we relied on dimensionality-changing networks to generate error signals whose dimensionality was consistent with the PES learning rule; we suspect that the brain includes more sophisticated representations that obviate the need for these specific network components.

5.6.4 Summary of Contributions

Taken together, the similarities between the empirical data we gathered from human participants, and the simulated data we generated from our cognitive agents, suggest a few conclusions. First, our models provide three mechanistic examples of how cognitive agents may use external feedback to improve their performance. When we applied constraints to agent cognition and learning, we observed that they struggled to learn effective TG strategies. In many cases, agents made similar mistakes as did our human participants, settling too quickly into suboptimal strategies or repeating exploratory moves that failed in the past. In other cases, agent mistakes were distinctly non-human, such as when agents made a good move on most turns but reliably made the same bad move once every game. We showed (through parameter fitting) that the learning trajectory of an individual agent could quantitatively recreate the learning trajectory of an individual human. However, we found that these agents were overfit to the data: they often performed poorly when their parameters were adjusted slightly, or when they played against a novel T4T agent. Similarly, we were able to generate populations of agents that, following learning, recreated some of the high-level trends in the human data, particularly the empirical distributions of generosities in the final games. However, closer inspection of these results revealed systematic differences between human and agent behavior, both at the individual and group level.

One parameter that did reliably predict differences in agent and human behavior was SVO: prosocial individuals were more generous (than proself participants) when playing against generous opponents. Many other social experiments have found that individuals who are prosocially oriented are more likely to cooperate with opponents in social dilemmas like the TG. Given that SVO is an important determinant of interesting social behaviors like cooperation, altruism, and reciprocity, we wanted to include mechanisms and parameters in our cognitive agents that permitted prosocial thinking. Following recent examples of SVO in computational models [43, 42, 116], we implemented SVO by having agents explicitly value the rewards earned by opponents, with weights determining the relative value of self-reward, other-reward, and reward-inequality. When we generated populations of unique agents, these weights (w_i and w_o) were randomly chosen, and we labelled agents as proself or prosocial based on whether the sum of these parameters exceeded a specific threshold. We found that this classification significantly influenced final agent behaviors and performance: as with the human data, prosocial agents were more generous and scored higher against generous opponents.

This is particularly interesting in the context of learning agents: given sufficient exploration of the (s, a) space, a proself RL agent should discover that generous strategies have

the highest expected value, and hence learn to adopt them. Why did our proself agents fail to discover these high-scoring strategies? We believe the answer relates to the cognitive constraints we placed on agents: their memories were imperfect, and they had limited opportunities to explore different actions. To achieve the full benefits of cooperation against our generous T4T opponent, our agents needed to make generous transfers several turns in a row; each turn, the T4T player would return a greater amount, but would become greedy if the agent made a greedy transfer. The consequence of this behavior is that a greedy strategy would lead to a sure payout of 10 coins every round, a strategy of mixed greed and generosity would lead to a typical payout of less than 10 coins, and a consistently generous strategy would lead to an average payout of more than 10 coins. This created a large region of the (s, a) space where the best way to improve a strategy was to become more greedy, leading to local maxima in the value landscape. Greedy agents discovered and exploited this strategy. In contrast, generous agents were additionally rewarded based on the coins earned by the opponent, which changed the shape of the value landscape, greatly increasing the chances that these agents could discover the global maximum. In other worlds, SVO encouraged agents to explore cooperative strategies with the potential for greater long-term benefits. However, this came at a price: prosocial agents were more susceptible to exploitation by greedy T4T opponents. While our prosocial agents were able to avoid this exploitation in some experimental conditions (playing against a greedy trustee), their generosity led to significantly lower scores in other conditions (playing against a greedy investor). We believe that similar tradeoffs exist in many real-world social interactions, and hope that future work continues to (a) design experiments where individuals must navigate these complex decisions, and (b) investigate how humans learn to navigate these tradeoffs using RL and SVO.

Our models, particularly our NEF agent, make several predictions that can be tested in future empirical experiments. First, our SVO mechanism implies that the internal reward signal experienced by an individual should be given by Eq. 5.2; we expect that the activity of midbrain dopaminergic neurons correlates with these three terms (self-reward, other-reward, and reward-inequality) when performing a social task. We predict that prosocial individuals would contain neurons whose activity increased when others received higher rewards, and decreased when reward inequality rose, given a fixed amount of self-reward; we also predict that these neurons would be less common, or their changes in activity would be less extreme, in proself individuals. Second, the multi-phase learning schedule of our NEF agent implies that the brain retrieves an explicit estimate of previous value estimates during learning. We predict that if an experimenter decoded a value estimate from neurons in vmPFC (or working memory buffers elsewhere in cortex) immediately following external feedback, then decoded another value estimate from the same neurons following learning

and consolidation, that the magnitude of the difference in these estimates would correlate with the magnitude of RPE neuron activity immediately following feedback. Finally, our use of spatial semantic pointers in the NEF agent implies that the agent generalizes value estimates across game states. This generalization could be measured empirically and compared to our model outputs using the following scheme. A simulated investor would play as normal, except they would never send exactly nine coins to the trustee. This would leave a specific gap in the trustee's value estimate for this game state. We predict that the value assigned by our NEF agent to this game state, and the action taken in that state, would reliably correlate with the values and actions from neighboring game states (e.g., values estimates and generosity when presented with eight or ten coins), and that this same relationship could be observed in human data (measured neural activity for value estimates, or measured generosity for actions).

Overall, while we maintain that our models help inform our understanding of the cognitive mechanisms of social reinforcement learning and decision making, and are capable of reproducing some human behaviors, we do not claim that our models quantitatively explain trends in human data, nor have we thoroughly explored which of the many parameters and mechanisms in our models are primarily responsible for the effects we observed. Our approach is best viewed as an initial exploration of how RL may be incorporated into cognitive models and how these models can be validated with empirical data.

5.6.5 Summary of Limitations and Future Work

Our empirical experiment was successful in gathering human data for validation, but could be improved in several ways. First, we were surprised at how few participants displayed coherent learning trajectories over the course of the experiment. While empirical studies will always include inattentive or impatient individuals, changing the difficulty of the task may encourage higher-quality participation and promote more gradual learning, which would be more amenable to analysis and reproduction. The difficulty of our task was determined by two factors: the random behavior of agents, which required participants to sample (s, a) several times before they could accurately estimate the mean value; and the adaptive nature of agents, which required participants to examine sequences of actions to determine the value of candidate actions (e.g., generous actions were initially unrewarding, but highly rewarding if repeated). In retrospect, agent randomness made analysis unduly difficult, and may have discouraged participants who got "unlucky" several times in a row; future experiments should find deterministic means of making agent behavior more humanlike. In contrast, we may have made the task too easy by using simulated opponents against whom the optimal policy was pure greed or pure generosity. Many participants

adopted these policies very quickly, leading to learning trajectories with few interesting features. Future work should design social experiments where the optimal policy can only be discovered by repeated trial and error. The learning task should be easy enough that individuals can clearly assess how rewards correlate with states and actions, encouraging them to experiment and find better strategies, but hard enough that discovering the optimal strategy is not guaranteed.

Our T4T agents were designed with these considerations in mind, but could be improved. We observed, in particular, that human learning when playing the trustee was extremely inconsistent. This is partly due to the fundamental complexity of playing as the trustee: this player has a larger (s, a) space (the number of coins available changes based on the investor's transfer, and candidate actions range from 0-30 instead of 0-10). However, it was likely also due to agent randomness, and to large state updates by the T4T agents (greed and generosity would be strongly punished or rewarded, leading to large swings in behavior between every round). Experimentation with the T4T update parameters F and P could alleviate this issue.

Future experiments should also investigate human-human interactions to promote realism and prosocial thinking. Experimental confederates could be used in lieu of of software opponents to provide control within the study; alternatively, the experiment could simulate free-for-all play between participants, then conduct a thorough data analysis to identify trends related to SVO, opponent behavior, etc. Finally, future work should analyse the effects of personality factors beyond SVO, for instance by using standardized questionnaires or personality tests to classify individuals. While our SVO classification was significant in many of our experimental conditions, it was assigned using only a single question, which required self-reporting after participants played the TG: other means for assessing participant SVO may be more robust.

Our cognitive agents could be expanded in multiple ways. We discussed architecture-specific extensions above, but several higher-level changes may be applicable to all three agents. First, our agents represented the state of the world by encoding the current round of the TG and the number of coins available. While this should be sufficient information to discover good strategies against our simulated opponents, other representations should be explored. Real social interactions feature the exchange of many social, emotional, and linguistic cues; perceiving these cues and using them to guide decision making is an important part of social decision making, even in constrained social interactions like the TG. Second, our agents did not include specialized valuation systems, like those we discussed in Sec. 5.2.3; instead, our agents combined all state information into a single complex representation, then learned a single mapping between that representation and action values. It would be interesting to divide evaluation into distinct subsystems, then

combine these systems during value integration: doing so would help break up the problem of learning action values (in an environment containing multiple features, and for agents with multiple goals) into manageable chunks. Future work should also endow agents with other learning mechanisms, such as frequentist models (e.g., learning the average reward of each action, independent of state) or model-based systems (e.g., mentalizing about how an action affects the opponent's state of mind and subsequent behavior).

On the topic of modulation, we would like to investigate the dynamic control of SVO parameters w_i and w_o. In our agents, these parameters were fixed at initialization, but neural evidence suggests that human prosocial tendencies may be updated on-the-fly, for instance by temporarily favoring social outcomes when interacting with a new social group, or quickly becoming indifferent to the wellbeing of others in response to betrayal [47]. The role of the amygdala in estimating trust and coordinating fear responses may be particularly relevant for such modulation [17]. Incorporating this modulation would not only make our cognitive agent more neurally and cognitively plausible, but it could potentially eliminate an un-humanlike error that all our agents made: repeatedly transferring one or two coins to a greedy opponent (thus satisfying their prosocial reward function) despite never receiving greater personal rewards for this generosity. Prosocial humans express their generosity by occasionally make one highly-generous transfer, then immediately returning to greediness if the opponent does not reciprocate. Such behavior could be explained by the rapid modulation of proself versus prosocial goals.

Another area where agent behavior departs from human behavior is exploration. While it is hard to quantify the process for human exploration, it seems likely that animals use sophisticated strategies for exploration and exploitation, including curiosity-driven learning, bayesian-optimal foraging, and the like [233, 221]. In contrast, our agents use the simplest possible exploration scheme: they chose random actions in the beginning of the experiment, and chose actions according to their policy towards the end. Not only is this exploration scheme wildly inefficient, it also requires external knowledge about how long the experiment will last, in order to properly set the ϵ-decay schedule. Thus, for both cognitive plausibility and improved performance, future work should investigate more advanced schemes for exploration.

A detailed parameter exploration would also improve our analyses. We showed that the SVO parameters w_i and w_o recreate differences between proself and prosocial humans across several experimental conditions. For completeness, future work should also attempt to recreate these differences using other model parameters. Our claims about the explanatory power of SVO would be greatly strengthened if only w_i and w_o could reproduce the empirical trends we observed. Future work might also explore alternative approaches for establishing the relationship between agent SVO and human SVO. For instance, we could

optimize model parameters (including SVO) to reproduce individual learning trajectories, then compare each human's SVO with the best-fit SVO parameters: if the two were reliably correlated, we could claim that our SVO mechanism predicted individual differences.

Finally, it is important to apply our models to other social tasks. While we have claimed that our architectures and representations are sufficiently general to apply to other types of social cognition, these claims should be tested. To confirm the general learning capabilities of our agents, we should train them to perform standard RL tasks, like gridworld or box-car. We should then teach our agents to play other social games, like the prisoner's dilemma. Finally, we would like to train our agents to play against other learning agents in tournament-style competitions, and apply the techniques of multi-agent reinforcement learning to study how social norms develop in a society filled with independent, cognitively-plausible learners.

5.7 Conclusion

In this chapter, we extended our analysis of learning and decision making in social contexts by investigating how humans learn to play the trust game. Drawing on the learning rules and network components we developed in previous chapters, we created an NEF agent with multiple cognitive capabilities: our network (a) encoded external inputs into a high-dimensional space that supports generalization over continuous variables (b) evaluated the utility of multiple potential actions, (c) compared these actions and selected the best one, and (d) learned based on feedback from the environment combined with (e) memories of previous states and actions. All these operations were computed online within the neural network. We also simulated two agents derived from totally different cognitive architectures, a deep neural network and a symbolic ACT-R agent. By comparing the cognitive mechanisms and performance of these agents, both with each other and with a human dataset, we studied the extent to which human learning could be recreated and explained by reinforcement learning.

We found that our agents recreated several empirical patterns of learning and behavior, but that our agents fell short in other respects. In particular, we showed that it is possible to operationalize social value orientation within the RL framework by endowing agents with an explicit preference for the wellbeing of others and for social norms. When we simulated populations of agents containing both proself and prosocial individuals, we found that the differences in their behavior mirrored high-level differences between proself and prosocial humans. Overall, our models and experiments show that cognitive agents trained using

RL are useful tools for studying human social behavior, and suggest several conclusions regarding exploration and cooperation that should be fleshed out in future work

In the concluding chapter of this thesis, we look back on the models we have developed, and synthesize our findings into a coherent story. We discuss the strengths and weaknesses of our approach to neurocognitive modelling, and point to alternative methods used in computational neuroscience. We conclude by identifying several open questions that arise from our results, and imagining what kinds of models and experiments might be used to answer them.

Chapter 6

Conclusion

6.1 Contribution

Our goal in this thesis was to explore social cognition from the perspective of learning and decision making. We sought to describe the cognitive algorithms that govern social behavior by proposing mechanistic theories for social information processing, identifying the neural and anatomical correlates of these mechanisms, and showcasing how biological brains might implement them. To do so, we built computational models that realized algorithms for learning and decision making, trained these models to perform simple cognitive tasks from social psychology, and validated them by comparing model outputs to neural and behavioral data from social animals.

In Ch. 1, we motivated the study of social decision making, reviewed the functional neuroanatomy of social cognition in the brain, and introduced a theoretical framework for building biologically-plausible computational models. In particular, we endorsed the value based framework as a theory for social decision making, pointed to biological evidence that supported this theory, and suggested how the NEF might be used to implement the required components.

In Ch. 2, we modeled associative learning in the amygdala, focusing on the well-documented examples of fear conditioning and extinction. Using an anatomically-detailed model, we recreated several nuclei within the amygdala, specified their connectivity, and ran several fear conditioning experiments on our simulated agents. We validated our model by comparing the tuning properties of our simulated neurons to the neural responses of rodents and primates, both in normal conditions and under pharmacological or physiological perturbation. We observed that our simulated neurons exhibited realistic sensitivity

to external stimuli at various stages of conditioning, and that perturbing certain model components during training produced plausible changes in these responses. We also compared the behavioral responses of our model to empirical data, and reproduced several classic results in the fear conditioning literature. Finally, we leveraged the unique representational properties of our model to investigate how fear generalized to a spectrum of stimuli resembling the conditioned stimulus. We found that the model exhibited humanlike generalization gradients, and that these gradients could be shifted to model people with anxiety disorders by weakening pattern separation within the model.

In Ch. 3, we took a closer look at computational models with a high degree of biological detail. This chapter was motivated by a concern that incorporating such details might limit the cognitive capacity of our models. We proposed a theoretical extension to the NEF that used supervised, online learning, together with offline optimization, to discover the synaptic weights and time constants that would accommodate biologically-detailed networks. We validated our method by training these detailed networks to perform several operations that are essential to cognitive systems. We also constructed a highly-detailed model of working memory in PFC, then showed that its mnemonic properties were comparable to those of simple animals performing a delayed match-to-sample task. While this chapter did not investigate social cognition per-se, it showed that our models may accommodate additional biological details, allowing us to investigate how social decision making is affected by emotional neuromodulation in future work.

In Ch. 4, we modeled the speed-accuracy tradeoff in decision making by building a network that performed several interrelated cognitive operations. Our network tracked the changing value of two choice alternatives, monitored its uncertainty and external time pressures over time, and made a decision once a flexible decision criterion was met. The model was informed by theories of inference and decision making under time pressure, and its anatomical structure was consistent with valuation and decision making systems in cortex and basal ganglia. We validated the model by comparing its behavior to a human dataset; we recreated the behavior of individual participants, captured the speed-accuracy tradeoff, and identified which model mechanisms were most influential.

In Ch. 5, we combined many of the network components developed throughout this thesis into an agent with multiple cognitive capabilities. The agent encoded external inputs into a high-dimensional space that represents continuous variables, evaluated the utility of multiple potential actions, compared and selected the best action, and updated its mental representations based on feedback from the environment combined with memories of previous states and actions. We trained this agent to play the trust game, a simple social dilemma that is often used in psychology and economics to study the relationships between trust, cooperation, and social decision making. We also simulated agents from two other

influential cognitive architectures, a deep neural network and an ACT-R agent. All three of these agents were trained using reinforcement learning, endowed with parameterized social value orientation by adding prosocial terms into the reward function, and played the trust game against adaptive simulated opponents. Furthermore, we conducted a human experiment where participants played the trust game, and were classified according to their social value orientation. We analyzed the learning trajectories of simulated agents and human participants, and noted several similarities and differences. Within the populations of simulated agents and humans, some individuals experimented with different strategies and only settled into routine behavior late into the experiment, while other individuals quickly decided upon a strategy and did not explore alternatives. Similarly, some individuals discovered high-scoring strategies against the simulated opponent they faced, while others became fixated on suboptimal strategies. We found that, compared to their proself counterparts, prosocial participants and prosocial agents were more generous and scored higher against cooperative opponents. We discussed the implications of these results for the emergence of cooperative behavior, but pointed out ways in which our agents were cognitively or behaviorally unrealistic.

In this chapter, we discuss the overall strengths and weaknesses of our modelling approach, and point out alternative methods used in computational neuroscience. We also identify ways to increase biological and cognitive realism, other social tasks we would like to investigate, and new analysis techniques that would solidify our results.

6.2 Critiquing our Modelling Approach

In previous chapters, we discussed the strengths and weaknesses of our individual models. Rather than review those discussions here, we reflect on the high-level strengths and weaknesses of NEF models, giving examples from the thesis as appropriate, and comparing them with other influential frameworks in computational neuroscience.

6.2.1 Strengths

As we discussed in Sec. 1.2, computational models are powerful tools for studying social cognition: they provide a mechanistic account of brain function, rigorously specify cognitive theories, contrast competing hypotheses, and synthesize disparate theories using a common framework. The NEF, in particular, is a framework that lets modellers generate and test *functional* hypotheses about the brain's cognitive mechanisms. When we build brain

models using the NEF, we frequently do so using a top-down approach, where we specify the functional roles of certain populations and connections, then connect these components to realize a network-level cognitive algorithm. This approach to "neural engineering" makes it easy to generate models that perform a variety of functions, ranging from small-scale mathematical operations to large-scale symbolic cognition. The models we presented in Chs. 2-3 used connections between a few neural populations to associate input stimuli and store short-term memories, while the models in Chs. 4-5 used multiple accumulators, memories, and selectors to perform entire cognitive tasks. For researchers interested in studying the brain from a functional perspective, the NEF provides a flexible theory of neural information processing, and Nengo facilitates the rapid construction and testing of cognitive models.

One useful property of the NEF and SPA is the ability to represent symbols at various levels of abstraction, and to transform those symbols in mathematically precise ways. In Ch. 4, we represented choice alternatives as two one-dimensional variables: this simple encoding allows easy comparison between choice values, for the purposes of calculating certainty and selecting the highest-value action. In Ch. 2, we represented external sensory stimuli as one- to five-dimensional vectors, and learned associations between these stimuli using online learning rules for updating synaptic weights. The moderate dimensionality of this representation allowed us to present a variety of stimuli to the network without hard-coding the response properties of neural populations, a common practice in computational models of fear conditioning. Furthermore, it allowed us to study how the model responded to stimuli outside the training set, permitting a novel investigation of fear generalization that would be impossible in models with simpler representations. Finally, in Ch. 5, we represented the entire game-state of the trust game using a 150-dimensional vector. This symbolic representation used spatial semantic pointers to encode socially-relevant information in a continuous feature space, which (a) allowed the network to learn sophisticated evaluations of candidate actions that generalized across similar game-states, (b) can easily be extended to other cognitive tasks without restructuring the network, and (c) is analogous to neural representation of continuous variables in hippocampus. Taken as a whole, our thesis shows the power and flexibility of NEF-style representations for cognitive modelling.

Another advantage of NEF models is the ability to integrate many functional components into a larger cognitive network. In our models, neural activity represents all internal states of the model, and neural connection weights realize all symbolic manipulations; because these operations share the same substrate, and because they are governed by the same dynamical equations, combining these operations is straightforward. Ch. 3 showed that these representations and operations can be supported by biologically-detailed net-

works, justifying our description of NEF models as brain models. Chs. 4-5 then showcased the integrative potential of these networks for social cognition: both networks performed cognitive tasks requiring perception, internal deliberation, and behavior. Other NEF and SPA models, notably SPAUN [66], have extended the notion of unified networks further: these models realize perception through a dedicated visual system and realize behavior through a motor system, and are capable of performing (and switching between) multiple tasks based on external cues. Our thesis was focused on developing and validating the cognitive components required for social cognition in particular; as we discuss below, our future work will continue to synthesize these components into larger, SPAUN-like agents that are capable of performing many social tasks.

In addition to their symbolic and functional capabilities, NEF networks accommodate many important biological constraints. While Ch. 3 showed just how detailed such networks could be, our remaining models also resemble real brains in many respects. To begin, all our models dynamically simulated populations of LIF neurons with unique tuning curves, and these neurons were connected with current-based exponential synapses; this led to neural representations that were imperfect and continuously changed as the network evolved through time. In Ch. 2, we recreated the anatomical structure of the amygdala: we simulated numerous neural populations corresponding to the various amygdala nuclei, and specified their internal connectivity, external connectivity, and representational properties to match with empirical data. In Ch. 4, we modelled a decision making circuit in cortex and basal ganglia: the connections in this network were anatomically-plausible, and the effects of urgency and uncertainty were realized using neural control according to leading theories of biological decision making. Finally, in Ch. 5, we modeled reinforcement learning in the value based framework; while we did not commit to a particular anatomical mapping, both theories are broadly consistent with the organization of the brain.

Finally, NEF models can be validated with a broad array of empirical data. Ch. 2 compared the neural response curves of simulated amygdala neurons to electrophysiological recordings taken during fear conditioning and extinction experiments. It also compared behavioral data from these experiments to the decoded output of the model. Ch. 3 compared the mnemonic properties of our detailed working memory model to animals performing a delayed response task, and compared the individual variability of our models to the range of values reported across species and experiments. Ch. 4 compared the speed-accuracy tradeoff of simulated agents to humans performing an identical sequential sampling task, both at the individual level and across the population. Similarly, Ch. 5 compared the learning trajectories and final strategies of simulated agents to human participants playing the trust game, at the individual and group level. In all these experiments, we observed important similarities and differences between the simulated and empirical data, suggest-

ing that our models broadly explain the relevant cognitive mechanisms, but leaving room for model extensions and improved analyses.

Overall, the spectrum of models we present in this thesis showcases the versatility of NEF models with respect to functional capacity, symbolic representation, biological plausibility, and empirical validation.

6.2.2 Weaknesses

Although the functional aspects of NEF models are scientifically useful in many respects, they are not without their drawbacks. Some researchers have complained that NEF models are over-engineered: modellers know what they want the network to do, and can specify all the representations and connections to ensure that it does what is required. These researchers advocate for an alternative, bottom-up approach, where few features of the network are specified, and the model must learn how to solve a task from scratch. It is true that, in this thesis, we often assume that specific functions are performed by specific parts of the network. We would defend our approach in three ways.

First, it is valuable to study the behavior of a network, even if we do not know how that network came to exist. For instance, even though the model in Ch. 2 has a fixed structure and does not include learning, we still generated insights into the cognitive factors that influence the speed-accuracy tradeoff: we showed how high-level parameters like decision threshold, temporal urgency, and choice certainty could be realized in a neural network, and studied how variance in these parameters could produce behavioral variance within a population of individuals. Second, many parts of the brain (perhaps most) are not undifferentiated structures capable of learning any function: the genetic coding of nervous system development specifies how certain populations connect and at least partly dictates the functions that certain networks compute. One notable exception is the cortex, which is highly flexible, and adapts to the sensory inputs and cognitive demands placed on brains during development. Future work should explore whether NEF networks can realize the general-purpose, model-based learning that occurs in cortex with minimal top-down constraints. Third, NEF models are not always top-down: they include a significant degree of learning, and the functions learned by the networks are not always comprehensible in closed-form functional terms. For instance, our amygdala model in Ch. 2 learned a straightforward mapping between the presence of one external stimuli and the presence or absence of another, but our cognitive agent in Ch. 5 learned a complex mapping between its unique internal representation of the external world and its behavioral response. Therefore, while NEF models often leverage top-down functionality when designing brain models, they are

also capable of explaining how neural networks may learn to perform cognitive operations with minimal engineering.

6.2.3 Alternatives

The NEF is one of many computational neuroscience frameworks that are used to study the cognitive mechanisms of social decision making. These frameworks differ greatly with respect to the desirable computational and cognitive features we outlined in Sec. 1.2. A detailed analysis of the theories and models that accompany each framework is beyond the scope of this chapter, but we briefly mention some of their strengths and weaknesses to provide additional context to our results.

Deep neural networks are widely used in artificial intelligence and robotics, and the last several decades have seen enormous growth in their functional capabilities. With the advent of DNN tools like convnets, transformers, and LSTMs, these networks are capable of solving a multitude of complex problems. For instance, the networks developed by Google DeepMind have now surpassed human performance in strategic games like Chess and Go, and large language models like ChatGPT can generate text that mimics human writing. If cognitive ability is defined strictly as the ability to perform complex tasks, then DNNs are undoubtedly powerful cognitive systems. However, DNNs are generally built for the purposes of solving specific tasks, rather than explaining biological cognition. As a result, the dynamical processing of information in DNNs, and the cognitive mechanisms they employ, bear little resemblance to brains. For example, backpropagation is not a biologically-plausible learning rule, and the connectivity of neural populations in DNNs is not anatomically realistic. Of course, there are exceptions: DNNs have recreated many aspects of the rodent visual system [211], some language models appear to solve linguistic tasks in human-like ways [117], and DNNs trained with RL exhibit humanlike behavior in multiplayer games [253]. Overall, DNNs are versatile AI systems, but must be highly constrained if they seek to inform our understanding of human cognition.

The Adaptive Control of Thought - Rational (ACT-R) architecture is an integrated theory of cognition that functions on a symbolic level but implements subsymbolic operations and includes important cognitive constraints [5]. ACT-R models symbolically specify the information used within a system and the production rules that dictate how to use this information to perform actions. Through a series of subsystems that implement episodic memory, procedural memory, working memory, and pattern matching, ACT-R models realize many interconnected aspects of cognition, and can solve tasks end-to-end. Importantly, the subsymbolic processing that occurs within ACT-R subsystems is governed

by equations that impose cognitive constraints on information processing, such as noisy information retrieval and temporal delays. These equations are motivated by the cognitive constraints imposed by the brain, but are enacted at a subsymbolic level of analysis rather than a neural level. Thus, while ACT-R can be used to study domains including learning, memory, decision making, language, and attention, it abstracts away from the neural basis of cognition, and cannot model many phenomena that are induced by neural representation. For example, while ACT-R models could probably recreate many of the behavioral findings regarding fear conditioning and extinction from Ch. 2, they could not be used to reproduce the response curves of neurons within the amygdala, study the effects of inactivating certain populations, or interpret fear generalization as arising from pattern separation within neural populations. Despite these shortcomings, ACT-R is supported by an extensive history of cognitive models that are validated by behavioral data, and is one of the leading frameworks for studying human cognition.

The Efficient Balanced Networks (EBN) and FORCE methods simulate neural dynamics using spike space and state space equations that resemble the NEF equations, and may be used to either specify the target dynamics (top-down) or learn them given appropriate error signals (bottom-up). In the conclusion of Ch. 3, we discussed in detail how the biological realism and cognitive capacity of these methods compared to the NEF. Overall, these methods have similar strengths and weaknesses as the NEF, but the NEF has a larger corpus of computational models spanning a wider range of neural and cognitive phenomenon, especially in large-scale integrated systems like SPAUN. This is partly due to the advanced tools for symbolic representation, and dedicated components for memory, decision making, and cognitive control, that have developed alongside the NEF and are used in the SPA. While EBN and FORCE models may accommodate the dynamics of high-dimensional vectors, they have not yet developed comparable tools for cognitive analysis.

The Local, Error-driven and Associative, Biologically Realistic Algorithm framework (LEABRA) simulates neurally- and anatomically-realistic models of learning in brain areas like hippocampus, prefrontal cortex, and basal ganglia [178]. LEABRA models focus on Hebbian learning mechanisms in networks with bidirectional connectivity, and in general take a bottom-up approach to cognitive modelling: LEABRA models begin with undifferentiated connections, and seek to show how a core set of computational mechanisms facilitate learning of cognitive operation like memory, vision, and attention. This focus has produced numerous models that are tightly-coupled with low-level empirical data, and inform our understanding of how cognition arises given strong biological constraints. Overall, LEABRA is more concerned with biological realism and the emergence of specific cognitive operations in the brain, whereas the NEF is more concerned with high-level cognition and

symbolic processing in integrated systems.

Finally, computational models of predictive coding and Bayesian inference are becoming increasingly popular. These models seek to explain how the brain realizes two important cognitive operations: updating beliefs about the likelihood of events in the external world, and active prediction of future states. As we did in Ch. 5, these models use prediction errors to update mental models of the external world: however, these models typically use more advanced learning rules, and more highly-structured neural circuits, to implement Bayesian or model-based computations. The mechanisms that are hypothesized to underlie predictive coding vary significantly between different computational models, and no single perspective has dominated this field of research [216]: two prominent examples are process theory [181] and the free energy principle [79]. Unfortunately, computational models of predictive coding frequently prioritize functional capacity over biological realism: while some models do use plausible neural networks or anatomical connections, few are concerned with comparing model signals with neural data, or validating model outputs against behavioral data. On the other hand, many of the Bayesian inference models we discussed in Sec. 4.2.3 are both cognitively powerful and biologically plausible, and have been validated with behavioral data [123, 124, 125, 187]. These models assume that an agent's beliefs about the external world can be described using probability distributions, and provide methods for updating those beliefs based on Bayes' rule. These rich representations of the world may serve as the basis for a wider variety of cognitive operations than the simple representations we used in the DM model in Ch. 4. However, these representations and computations often require a significant degree of prior knowledge about the mathematical functions that generate events in the real world; it is unclear whether animals possess this prior knowledge, or whether they can achieve comparable belief updates through more ad-hoc computations. It is also worth noting that the NEF may be used to build neural networks that perform certain Bayesian computations [210], and that Spatial Semantic Pointers may be used to perform probabilistic computations within a vector symbolic architecture [80]. In fact, our application of SSPs in the NEF agent from Ch. 6 captures some of the benefits of Bayesian inference and belief updates, such as the ability to (a) estimate values for unexplored regions of the state space, and (b) update multiple value estimates in a single learning step. This suggests that predictive coding and Bayesian inference may be realizable within biological NEF networks.

6.3 Future Work

To conclude, we collect the suggestions for future work we have mentioned throughout this thesis, and propose other projects where computational models may improve our understanding of social decision making.

6.3.1 Sensitivity Analysis

One of the challenges with analyzing and validating complex models is that they include many free parameters. Throughout this thesis, we have described the cognitive relevance of our model parameters, fit these parameters to match individual data, and varied them across a population of agents to simulate a diverse population. However, we did not conduct any thorough sensitivity analyses on these parameters, and only in Ch. 4 did we investigate the relative contribution of each model parameter. In general, simpler models are preferable to complex models, if both can explain the data equally well. Future work should systematically remove various parameters and mechanisms in our models, and test whether our conclusions still hold. When doing so, it is important to validate against various classes of data, such as neural response curves, lesion studies, and behavioral studies: while a simpler model may appear better than a complex one if it is sufficient to explain one of these domains, complex models may be necessary to simultaneously capture data from multiple domains.

6.3.2 Biological Improvements

Our models simulated varying degrees of biological realism, and we attempted to connect our findings back to biological brains through comparison to neural and behavioral data. While computational brain models can always be made more biologically realistic, there are a few specific aspects of biology that we feel are especially important for future work. First, we want to more clearly define how the neural populations in our cognitive agents (Ch. 4-5) map onto the anatomical divisions of the brain. This will involve restructuring our neural populations so that their representations and computations more closely align with functional neuroanatomy. For instance, in our trust game agent, we could define separate processing streams for proself and prosocial valuation. Structural changes like this will make it easier to compare the activity of simulated populations with electrophysiological measurements from behaving animals, giving us a clearer picture of exactly how cognitive operations are distributed through the brain.

In order to develop broader theories for how different neurotransmitters are able to modulate and control cognition, we would also like to include more biophysical mechanisms in our cognitive agents. The tools we developed in Ch. 3 allow the simulation of such details in NEF networks, but we did not develop a theory for how these details functionally alter cognition. We think that a framework for understanding how neuromodulators (especially those associated with emotional states) exert control over multiple cognitive systems would be particularly insightful when studying social behavior. At a lower level, we also want to understand whether division between excitatory and inhibitory synapses is crucial for social cognition, or merely an artefact of the brain's evolutionary history that can be approximated with mixed-weight connections.

Lastly, while we strove to validate our models against neural and behavioral data, we were unable to find experiments that simultaneously measured both quantities in our chosen domains. Datasets that include both measurements would provide a more unified point of comparison for our models, allowing us to make stronger claims about the relationship between neural activity, symbolic representation, and behavior. Scientific collaboration between empirical researchers and theoretical modellers is perhaps the best way to ensure that all the relevant experimental variables are aligned; in future work, we hope to design models alongside experiments, rather than build models that explain empirical results *post-facto*.

6.3.3 Cognitive Tasks

In Ch. 1, we outlined the functional neuroanatomy of the value based framework, describing the brain areas associated with value estimation, integration, modulation, selection, and updating. Although the NEF agent we presented in Ch. 5 simulated value estimation, action selection, and reinforcement learning, it should be extended to account for the separate computation, modulation, and integration of specialized value estimates. Apart from increasing the biological realism of the agent, these additions would allow a more nuanced investigation of prosocial cognition: for instance, such an agent could compute the prosocial value of a generous action, then decide whether or not to account for this value based on the current social context. Similarly, adding systems that explicitly track trust in other people, possibly by forming memories associated with particular individuals, could help recreate the complex web of cognitive heuristics that people use in social situations. Both the amygdala model in Ch. 2 and the flexible value accumulator of Ch. 4 could be incorporated into our NEF agent for this purpose. Longer-term, we would also like to simulate the process of mentalizing, which requires learning sophisticated models of the relationships between states, actions, and rewards, and also requires integrating those models into value

estimates by imagining possible action sequences. Modelling these complex cognitive operations will require significant theoretical advances, but offer the possibility of capturing social abilities that are uniquely human.

We would also like to apply our cognitive agents to a greater number of social and nonsocial tasks. Doing so would accomplish two goals. First, it would support our claims about the cognitive generality of our networks: that is, it would show that our representations and operations are not over-engineered for a particular task. Ideally, we could develop a SPAUN-like agent that is capable of performing multiple social tasks without retraining, or that could use knowledge from one task when learning to perform another. Second, we could use such an agent to investigate a host of other social phenomena. For instance, it would be helpful to apply our agent to cognitive tasks where learning is central to performing the task correctly. While we tried to construct such a task in our trust game experiment, we found that humans often used strategic shortcuts, rather than exploring multiple strategies and learning which one was superior. Comparing simulated and empirical learning on such a task would provide better insights than many of our current experiments, which were restricted to comparing the final performance of simulated agents to behaving animals. We are particularly interested in learning tasks that involve the social interaction of multiple intelligent agents, as these more closely resemble real-world social interactions than strategic games like the prisoner's dilemma. It would be interesting to apply the methods of multi-agent reinforcement learning to populations of neurocognitive agents, for example to study the emergence or dissolution of cooperation given the presence of a few greedy individuals.

6.3.4 Theoretical Extensions

In addition to the extensions listed above, there are several mathematical and conceptual extensions that would increase the cognitive plausibility of our models. Some of these are aimed at addressing the weaknesses we identified in Sec. 6.2.2: we want to ensure that our models are not over-engineered for particular tasks, and to show that our models can learn cognitive operations that are not specified during development. For example, future work could investigate whether it is possible to learn the internal representations that are necessary for a particular task. Previous NEF models have used encoder learning to show how the representation of a neural population may be adjusted to align with the features of a set of input stimuli [251]. It would be interesting to explore whether these methods could be applied throughout a network that contains dedicated components, such as working memories or action selectors.

To further improve the cognitive plausibility of our models, we would like to investigate how candidate actions are generated and compared. In Chs. 2 and 4, we presented models where choices were binary: outputs from the amygdala model represented the degree of expressed fear (a one-dimensional scale), while the outputs from the sequential sampling model represented a binary choice. In Chs. 3 and Ch. 5, we presented models where each possible action was encoded as a separate variable, and the model output was chosen by identifying the variable with the highest value. In all these models, we specified the possible actions ahead of time. While this is standard practice in computational neuroscience, we suspect that biological brains do not meticulously evaluate every possible candidate action, then compare them simultaneously. Future work should explore ways in which the network itself might generate candidate actions. A related topic is continuous action spaces: although many cognitive tasks force participants to choose between discrete actions, behavior in natural environments is usually dynamic and reactive. For instance, both animal foraging and human conversation require brains to continuously generate action sequences that are distinct from previous actions taken by the animal. Some RL models have focused on evaluating and choosing actions in continuous spaces, for instance by combining weighted basic actions into a real-valued action (e.g., combining left/right/up/down actions into a 2D heading vector); future work should apply these insights to neurocognitive models.

There were also several types of learning that we did not investigate in this thesis. Perhaps the most important is "social learning", in which agents acquire skills and knowledge by observing and copying others. Given that a significant portion of human learning occurs by imitation (especially in novel contexts or tasks), or second-hand experience (via written or spoken language), understanding social learning is critical for understanding social behavior. Social learning has been widely studied in economics, psychology, and biology [106, 15], but usually from a behavioral perspective, rather than a neurocognitive one. However, there is significant overlap between the theories posed in the social learning literature, and the computational methods used in several subfields of RL. For example, simulated agents may use imitation learning, inverse learning, and transfer learning to infer the value functions and policies that drive the behavior of other agents, to transfer their knowledge from one domain or task into another, or to copy the behavior of other agents [111, 9, 259]. In future work, we would like to explore the connection between social learning and RL, with the goal of expanding the learning abilities of our neural agents.

Finally, there are several cognitive phenomena that are important to social cognition, but are currently too complex and poorly understood to be a central part of leading cognitive frameworks. As we mentioned above, model-based learning and deductive reasoning are important aspects of human cognition that we have not addressed here. These forms of learning contribute to advanced forms of social interaction, and are likely involved in

mentalizing. Future work should try to apply the insights from predictive coding to realize these processes (in a neural substrate) such that the resulting network can be incorporated into larger cognitive models. Recent work with neural models of Bayesian inference has been applied to study limited forms of theory of mind, in which multiple agents attempt to model and influence the belief of others in economic games [125]. Expanding this work to other cognitive tasks, and ensuring that the neural networks are consistent with known neuroanatomy, could be one approach to studying the neural basis of mentalizing. Language is another area that has received a great deal of attention in dedicated models, but which has not yet been fully integrated into models that perform a wider array of cognitive tasks. It would be interesting to incorporate language models into cognitive agents that play the trust game, allowing them to communicate with one another, and use the internal representations generated by linguistic processing when making decisions. Adding linguistic abilities to cognitive agents would also allow researchers to apply these models to a wider array of cognitive tasks that better represent real-world social interactions. Emotions also play a central role in our social lives; while many theories and computational models of emotion exist, few are concerned with studying the neural and functional basis of emotional processing, and fewer still feature emotional modulation in agents that perform other tasks. We hope to incorporate insights from the leading theories of emotional processing into future NEF models, and study the interplay between emotion and social cognition in a variety of domains.

6.4 Conclusion

In this thesis, we studied social cognition by building biologically-plausible computational models of learning and decision making. Our goal was to develop mechanistic explanations of the underlying cognitive capacities, to test our theories by simulating neural networks, and to validate our models by comparing to human and animal data. Over the course of four chapters, we presented models with various levels of biological realism, cognitive capacity, and empirical support. These models shared an underlying theoretical framework, which allowed us to explain the dynamics of our neural networks in functional terms, and to integrate network components into increasingly complex models of learning and behavior. We compared our simulated results to empirical data ranging from neural activities to behavioral performance, noting both successes and failures. In future work, we will build upon these models to investigate a broader range of social phenomenon, such as affective control and multi-agent interaction.

References

[1] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[2] John P Aggleton and Andrew W Young. The enigma of the amygdala: On its contribution to human emotion. *Cognitive Neuroscience of Emotion*, 2000.

[3] Alireza Alemi, Christian Machens, Sophie Denève, and Jean-Jacques Slotine. Learning arbitrary dynamics in efficient, balanced spiking networks using local plasticity rules. *arXiv preprint arXiv:1705.08026*, 2017.

[4] Alireza Alemi, Christian Machens, Sophie Denève, and Jean-Jacques Slotine. Learning arbitrary dynamics in efficient, balanced spiking networks using local plasticity rules. *arXiv preprint arXiv:1705.08026*, 2017.

[5] John R Anderson, Daniel Bothell, Michael D Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An integrated theory of the mind. *Psychological Review*, 111:1036, 2004.

[6] John R Anderson and Christian J Lebiere. *The atomic components of thought*. Psychology Press, 2014.

[7] Marta Andreatta, Estelle Leombruni, Evelyn Glotzbach-Schoon, Paul Pauli, and Andreas Mühlberger. Generalization of contextual fear in humans. *Behavior Therapy*, 46(5):583–596, 2015.

[8] Jorge L Armony, David Servan-Schreiber, Lizabeth M Romanski, Jonathan D Cohen, and Joseph E LeDoux. Stimulus generalization of fear responses: Effects of auditory cortex lesions in a computational model and in rats. *Cerebral Cortex*, 7(2):157–165, 1997.

[9] Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021.

[10] Solomon E Asch. Effects of group pressure upon the modification and distortion of judgments. *Organizational Influence Processes*, 58:295–303, 1951.

[11] Nava Ashraf, Iris Bohnet, and Nikita Piankov. Decomposing trust and trustworthiness. *Experimental Economics*, 9(3), 2006.

[12] Lee Averell and Andrew Heathcote. The form of the forgetting curve and the fate of memories. *Journal of Mathematical Psychology*, 55(1):25–35, 2011.

[13] Petra JJ Baarendse, Danielle S Counotte, Patricio O'donnell, and Louk JMJ Vanderschuren. Early social experience is critical for the development of cognitive control and dopamine modulation of prefrontal cortex function. *Neuropsychopharmacology*, 38(8):1485–1494, 2013.

[14] Iris Bakker, Theo Van Der Voordt, Peter Vink, and Jan De Boon. Pleasure, arousal, dominance: Mehrabian and Russell revisited. *Current Psychology*, 33(3):405–421, 2014.

[15] Albert Bandura and Richard H Walters. *Social Learning Theory*. Prentice Hall, 1977.

[16] Madeleine Bartlett, Terrence C Stewart, and Jeff Orchard. Biologically-based neural representations enable fast online shallow reinforcement learning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44, 2022.

[17] Thomas Baumgartner, Markus Heinrichs, Aline Vonlanthen, Urs Fischbacher, and Ernst Fehr. Oxytocin shapes the neural circuitry of trust and trust adaptation in humans. *Neuron*, 58(4):639–650, 2008.

[18] Timothy EJ Behrens, Laurence T Hunt, and Matthew FS Rushworth. The computation of social behavior. *Science*, 324(5931):1160–1164, 2009.

[19] Trevor Bekolay. Learning in large-scale spiking neural networks. Masters thesis, University of Waterloo, 2011.

[20] Trevor Bekolay, James Bergstra, Eric Hunsberger, Travis DeWolf, Terrence C Stewart, Daniel Rasmussen, Xuan Choo, Aaron Voelker, and Chris Eliasmith. Nengo: A python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*, 7:48, 2014.

[21] Trevor Bekolay, James Bergstra, Eric Hunsberger, Travis DeWolf, Terrence C Stewart, Daniel Rasmussen, Xuan Choo, Aaron Russell Voelker, and Chris Eliasmith. Nengo: A python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*, 7:48, 2014.

[22] Trevor Bekolay, Carter Kolbeck, and Chris Eliasmith. Simultaneous unsupervised and supervised learning of cognitive functions in biologically plausible spiking neural networks. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 35, 2013.

[23] Trevor Bekolay, Mark Laubach, and Chris Eliasmith. A spiking neural integrator model of the adaptive control of action by the medial prefrontal cortex. *Journal of Neuroscience*, 34(5):1892–1902, 2014.

[24] Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. In *Advances in Neural Information Processing Systems*, pages 787–797, 2018.

[25] James Bergstra, Dan Yamins, and David D Cox. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in Science Conference*, pages 13–20, 2013.

[26] Daniel W Bloodgood, Jonathan A Sugam, Andrew Holmes, and Thomas L Kash. Fear extinction requires infralimbic cortex projections to the basolateral amygdala. *Translational Psychiatry*, 8(1):1–11, 2018.

[27] Marco Bocchio, Sadegh Nabavi, and Marco Capogna. Synaptic plasticity, engrams, and network oscillations in amygdala circuits for storage and retrieval of emotional memories. *Neuron*, 94(4):731–743, 2017.

[28] Martin Boerlin, Christian K Machens, and Sophie Denève. Predictive coding of dynamical variables in balanced spiking networks. *PLoS Computational Biology*, 9(11):e1003258, 2013.

[29] Jelmer P. Borst, Sean Aubin, and Terrence C. Stewart. Effective computing in the brain: A whole-task spiking neural network model of associative recognition. In *Cognitive Computing 2018*, Philadelphia, Pennsylvania, 2018. Cognitive Computing.

[30] Tobias Brosch and David Sander. Neurocognitive mechanisms underlying value-based decision-making: From core values to economic value. *Frontiers in Human Neuroscience*, 7:398, 2013.

[31] Scott D Brown and Andrew Heathcote. The simplest complete model of choice response time: Linear ballistic accumulation. *Cognitive Psychology*, 57(3):153–178, 2008.

[32] David M Buss. *The Handbook of Evolutionary Psychology*. Springer International Publishing, Cham, 2016.

[33] Colin F Camerer. *Behavioral game theory: Experiments in strategic interaction*. Princeton University Press, 2011.

[34] Nicholas T Carnevale and Michael L Hines. *The NEURON book*. Cambridge University Press, 2006.

[35] Maxime Carrere and Frédéric Alexandre. A pavlovian model of the amygdala and its influence within the medial temporal lobe. *Frontiers in Systems Neuroscience*, 9:41, 2015.

[36] Caroline J Charpentier and John P O'Doherty. The application of computational models to social neuroscience: Promises and pitfalls. *Social Neuroscience*, 13(6):637–647, 2018.

[37] Soni Chaturvedi, Rutika N Titre, and Neha Sondhiya. Review of handwritten pattern recognition of digits and special characters using feed forward neural network and Izhikevich neural model. In *2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies*, pages 425–428. IEEE, 2014.

[38] Feng-Xuan Choo. *Spaun 2.0: Extending the world's largest functional brain model*. Ph.d. thesis, University of Waterloo, May 2018.

[39] Thomas B Christophel, P Christiaan Klink, Bernhard Spitzer, Pieter R Roelfsema, and John-Dylan Haynes. The distributed nature of working memory. *Trends in Cognitive Sciences*, 21(2):111–124, 2017.

[40] Stephane Ciocchi, Cyril Herry, François Grenier, Steffen BE Wolff, Johannes J Letzkus, Ioannis Vlachos, Ingrid Ehrlich, Rolf Sprengel, Karl Deisseroth, Michael B Stadler, et al. Encoding of conditioned fear in central amygdala inhibitory circuits. *Nature*, 468(7321):277–282, 2010.

[41] Paul Cisek, Geneviève Aude Puskas, and Stephany El-Murr. Decisions in changing conditions: The urgency-gating model. *Journal of Neuroscience*, 29(37), 2009.

[42] Michael G Collins and Ion Juvina. Trust miscalibration is sometimes necessary: An empirical study and a computational model. *Frontiers in Psychology*, 12, 2021.

[43] Michael G Collins, Ion Juvina, and Kevin A Gluck. Cognitive model of trust dynamics predicts human behavior within and between two games of strategic interaction with computerized confederate agents. *Frontiers in Psychology*, 7, 2016.

[44] Roshan Cools and Mark D'Esposito. Inverted-u–shaped dopamine actions on human working memory and cognitive control. *Biological Psychiatry*, 69(12):e113–e125, 2011.

[45] Clayton E Curtis and Mark D'Esposito. Persistent activity in the prefrontal cortex during working memory. *Trends in Cognitive Sciences*, 7(9):415–423, 2003.

[46] Jo Cutler and Daniel Campbell-Meiklejohn. A comparative fMRI meta-analysis of altruistic and strategic decisions to give. *Neuroimage*, 184:227–241, 2019.

[47] Carolyn H Declerck, Christophe Boone, and Griet Emonds. When do people cooperate? The neuroeconomics of prosocial decision making. *Brain and Cognition*, 81(1):95–117, 2013.

[48] Sophie Deneve, Peter E Latham, and Alexandre Pouget. Efficient computation and cue integration with noisy population codes. *Nature Neuroscience*, 4(8):826, 2001.

[49] Brian DePasquale, Christopher J Cueva, Kanaka Rajan, LF Abbott, et al. full-FORCE: A target-based method for training recurrent networks. *PloS one*, 13(2):e0191527, 2018.

[50] Andreea O Diaconescu, Christoph Mathys, Lilian AE Weber, Lars Kasper, Jan Mauer, and Klaas E Stephan. Hierarchical prediction errors in midbrain and septum during social learning. *Social Cognitive and Affective Neuroscience*, 12(4):618–634, 2017.

[51] Jochen Ditterich. Evidence for time-variant decision making. *European Journal of Neuroscience*, 24(12), 2006.

[52] Philippe Domenech, Jérôme Redouté, Etienne Koechlin, and Jean-Claude Dreher. The neuro-computational architecture of value-based selection in the human brain. *Cerebral Cortex*, 28(2):585–601, 2018.

[53] Peter Duggins and Chris Eliasmith. A spiking neuron model of pharmacologically-biased fear conditioning in the amygdala. In *Proceedings of the Society for Neuroscience*, Chicago, IL, 2019.

[54] Peter Duggins and Chris Eliasmith. Constructing functional models from biophysically-detailed neurons. *PLoS Computational Biology*, 18(9):e1010461, 2022.

[55] Peter Duggins, Dominik Krzemiński, Chris Eliasmith, and Szymon Wichary. A spiking neuron model of inferential decision making: Urgency, uncertainty, and the speed-accuracy tradeoff. In *Proceedings of the Annual Meeting of the Cognitive Science Society*. Seattle: Cognitive Science Society, 2020.

[56] Puck Duits, Danielle C Cath, Shmuel Lissek, Joop J Hox, Alfons O Hamm, Iris M Engelhard, Marcel A Van Den Hout, and Joke MP Baas. Updated meta-analysis of classical fear conditioning in the anxiety disorders. *Depression and Anxiety*, 32(4):239–253, 2015.

[57] Nicole S.Y. Dumont, Jeff Orchard, and Chris Eliasmith. A model of path integration that connects neural and symbolic representation. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44, Toronto, ON, 2022. Cognitive Science Society.

[58] Nicole S.Y. Dumont, Terry C Stewart, and Chris Eliasmith. Spiking neural network model of simultaneous localization and mapping with Spatial Semantic Pointers. In *Proceedings of Computational and Systems Neuroscience*, 2021.

[59] Joseph E Dunsmoor, Stephen R Mitroff, and Kevin S LaBar. Generalization of conditioned fear along a dimension of increasing fear intensity. *Learning & Memory*, 16(7):460–469, 2009.

[60] Joseph E Dunsmoor and Rony Paz. Fear generalization and anxiety: Behavioral and neural mechanisms. *Biological Psychiatry*, 78(5):336–343, 2015.

[61] Daniel Durstewitz, Jeremy K Seamans, and Terrence J Sejnowski. Dopamine-mediated stabilization of delay-period activity in a network model of prefrontal cortex. *Journal of Neurophysiology*, 83(3):1733–1750, 2000.

[62] Sevil Duvarci and Denis Pare. Amygdala microcircuits controlling learned fear. *Neuron*, 82(5):966–980, 2014.

[63] John C Eccles. Chemical transmission and Dale's principle. In *Progress in Brain Research*, volume 68, pages 3–13. Elsevier, 1986.

[64] Chris Eliasmith. *How to build a brain: A neural architecture for biological cognition*. Oxford University Press, 2013.

[65] Chris Eliasmith and Charles H Anderson. *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press, 2003.

[66] Chris Eliasmith, Terrence C Stewart, Xuan Choo, Trevor Bekolay, Travis DeWolf, Yichuan Tang, and Daniel Rasmussen. A large-scale model of the functioning brain. *Science*, 338(6111):1202–1205, 2012.

[67] Chris Eliasmith, Terrence C Stewart, Xuan Choo, Trevor Bekolay, Travis DeWolf, Yichuan Tang, and Daniel Rasmussen. A large-scale model of the functioning brain. *Science*, 338(6111):1202–1205, 2012.

[68] Chris Eliasmith and Oliver Trujillo. The use and abuse of large-scale brain models. *Current Opinion in Neurobiology*, 25:1–6, 2014.

[69] Nathan J Emery and David G Amaral. The role of the amygdala in primate social cognition. *Cognitive Neuroscience of Emotion*, 2000.

[70] Jim Engle-Warnick and Robert Slonim. The evolution of strategies in a repeated trust game. *Journal of Economic Behavior & Organization*, 2004.

[71] Nathan J Evans, Guy E Hawkins, Udo Boehm, Eric-Jan Wagenmakers, and Scott D Brown. The computations that support simple decision-making: A comparison between the diffusion and urgency-gating models. *Scientific Reports*, 7(1):1–13, 2017.

[72] Alan SR Fermin, Takehiko Yoshida, Junichiro Yoshimoto, Makoto Ito, Saori C Tanaka, and Kenji Doya. Model-based action planning involves cortico-cerebellar and basal ganglia networks. *Scientific Reports*, 6(1):1–14, 2016.

[73] Klaus Fiedler, Linda McCaughey, Johannes Prager, Jürgen Eichberger, and Knut Schnell. Speed-accuracy trade-offs in sample-based decisions. *Journal of Experimental Psychology: General*, 150(6):1203, 2021.

[74] Megan M Filkowski, R Nick Cochran, and Brian W Haas. Altruistic behavior: Mapping responses in the brain. *Neuroscience and Neuroeconomics*, 5:65, 2016.

[75] Kate D Fischl. *Neuromorphic models of the amygdala with applications to spike based computing and robotics*. PhD thesis, Johns Hopkins University, 2019.

[76] Richard FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1(6):445–466, 1961.

[77] Luciano Floridi and Massimo Chiriatti. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020.

[78] Michael J Frank. Hold your horses: A dynamic computational role for the subthalamic nucleus in decision making. *Neural Networks*, 19(8), 2006.

[79] Karl Friston. The free-energy principle: A rough guide to the brain? *Trends in Cognitive Sciences*, 13(7):293–301, 2009.

[80] Michael Furlong and Chris Eliasmith. Fractional binding in vector symbolic architectures as quasi-probability statements. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44, 2022.

[81] Filip Gesiarz and Molly J Crockett. Goal-directed, habitual and pavlovian prosocial behavior. *Frontiers in Behavioral Neuroscience*, 9:135, 2015.

[82] Stefano Ghirlanda and Magnus Enquist. A century of generalization. *Animal Behaviour*, 66(1):15–36, 2003.

[83] Aditya Gilra and Wulfram Gerstner. Predicting non-linear dynamics: A stable local learning scheme for recurrent spiking neural networks. *arXiv preprint arXiv:1702.06463*, 2017.

[84] Paul W Glimcher. Understanding dopamine and reinforcement learning: The dopamine reward prediction error hypothesis. *Proceedings of the National Academy of Sciences*, 108:15647–15654, 2011.

[85] Sebastian Gluth, Jörg Rieskamp, and Christian Büchel. Neural evidence for adaptive strategy selection in value-based decision-making. *Cerebral Cortex*, 24(8):2009–2021, 2014.

[86] Mark S Goldman, A Compte, and Xiao-Jing Wang. Neural integrator models. *Encyclopedia of Neuroscience*, pages 165–178, 2010.

[87] Mark S Goldman, Chris RS Kaneko, Guy Major, Emre Aksay, David W Tank, and Hyunjune Sebastian Seung. Linear regression of eye velocity on eye position and head velocity suggests a common oculomotor neural integrator. *Journal of Neurophysiology*, 88(2):659–665, 2002.

[88] Cleotilde Gonzalez, Javier Lerch, and Christian Lebiere. Instance-based learning in dynamic decision making. *Cognitive Science*, 27(4), 2003.

[89] Jan Gosmann, Aaron Voelker, and Chris Eliasmith. A spiking independent accumulator model for winner-take-all computation. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2017.

[90] Jan Gründemann and Andreas Lüthi. Ensemble coding in amygdala circuits for associative learning. *Current Opinion in Neurobiology*, 35:200–206, 2015.

[91] Ankur Gupta, Rohini Bansal, Hany Alashwal, Anil Safak Kacar, Fuat Balci, and Ahmed A Moustafa. Neural substrates of the drift-diffusion model in brain disorders. *Frontiers in Computational Neuroscience*, 15, 2021.

[92] Kevin Gurney, Tony J Prescott, and Peter Redgrave. A computational model of action selection in the basal ganglia. *Biological Cybernetics*, 84(6):401–410, 2001.

[93] Martin Guthrie, Arthur Leblois, André Garenne, and Thomas Boraud. Interaction between cognitive and motor cortico-basal ganglia loops during decision making: A computational study. *Journal of Neurophysiology*, 109(12):3025–3040, 2013.

[94] Alan N Hampton, Peter Bossaerts, and John P O'Doherty. Neural correlates of mentalizing-related computations during strategic interactions in humans. *Proceedings of the National Academy of Sciences*, 105(18):6741–6746, 2008.

[95] Craig Haney, Curtis Banks, and Philip Zimbardo. Interpersonal dynamics in a simulated prison. *The Sociology of Corrections*, pages 65–92, 1973.

[96] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

[97] Masahiko Haruno and Christopher D Frith. Activity in the amygdala elicited by unfair divisions predicts social value orientation. *Nature Neuroscience*, 13(2):160–161, 2010.

[98] John Haugeland. *Artificial intelligence: The very idea*. MIT press, 1989.

[99] Michael J Hawrylycz, Ed S Lein, Angela L Guillozet-Bongaarts, Elaine H Shen, Lydia Ng, Jeremy A Miller, Louie N Van De Lagemaat, Kimberly A Smith, Amanda Ebbert, Zackery L Riley, et al. An anatomically comprehensive atlas of the adult human brain transcriptome. *Nature*, 489(7416):391, 2012.

[100] David J Heeger and Wayne E Mackey. Oscillatory recurrent gated neural integrator circuits (organics), a unifying theoretical framework for neural dynamics. *Proceedings of the National Academy of Sciences*, 116(45):22783–22794, 2019.

[101] Hauke R Heekeren, Sean Marrett, and Leslie G Ungerleider. The neural systems that mediate human perceptual decision making. *Nature Reviews Neuroscience*, 9(6):467–479, 2008.

[102] Richard P Heitz. The speed-accuracy tradeoff: History, physiology, methodology, and behavior. *Frontiers in Neuroscience*, 8:150, 2014.

[103] Dirk Hermans, Michelle G Craske, Susan Mineka, and Peter F Lovibond. Extinction in human fear conditioning. *Biological Psychiatry*, 60(4):361–368, 2006.

[104] Dirk Hermans, Trinette Dirikx, Debora Vansteenwegenin, Frank Baeyens, Omer Van den Bergh, and Paul Eelen. Reinstatement of fear responses in human aversive conditioning. *Behaviour Research and Therapy*, 43(4):533–551, 2005.

[105] Cyril Herry, Stephane Ciocchi, Verena Senn, Lynda Demmou, Christian Müller, and Andreas Lüthi. Switching on and off fear by distinct neuronal circuits. *Nature*, 454(7204):600–606, 2008.

[106] William Hoppitt and Kevin N Lala. *Social Learning: An Introduction to Mechanisms, Methods, and Models*. Princeton University Press, 2013.

[107] James C Houk, C Bastianen, D Fansler, A Fishbach, D Fraser, PJ Reber, SA Roy, and LS Simo. Action selection and refinement in subcortical loops through basal ganglia and cerebellum. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1485):1573–1583, 2007.

[108] Yanping Huang and Rajesh PN Rao. Reward optimization in the primate brain: A probabilistic model of decision making under uncertainty. *PloS one*, 8(1):e53344, 2013.

[109] Z Josh Huang and Anirban Paul. The diversity of GABAergic neurons and neural communication elements. *Nature Reviews Neuroscience*, 20(9):563–572, 2019.

[110] Eric Hunsberger. System identification of adapting neurons. Technical report, University of Waterloo, Centre for Theoretical Neuroscience, 2016.

[111] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys*, 50(2):1–35, 2017.

[112] Cendri A Hutcherson, Benjamin Bushong, and Antonio Rangel. A neurocomputational model of altruistic choice and its implications. *Neuron*, 87(2):451–462, 2015.

[113] Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, 2003.

[114] Yohan J John, Basilis Zikopoulos, Daniel Bullock, and Helen Barbas. The emotional gatekeeper: A computational model of attentional selection and suppression through the pathway from the amygdala to the inhibitory thalamic reticular nucleus. *PLoS Computational Biology*, 12(2):e1004722, 2016.

[115] Noel D Johnson and Alexandra A Mislin. Trust games: A meta-analysis. *Journal of Economic Psychology*, 32(5), 2011.

[116] Ion Juvina, Christian Lebiere, and Cleotilde Gonzalez. Modeling trust dynamics in strategic interaction. *Journal of Applied Research in Memory and Cognition*, 4(3):197–211, 2015.

[117] Ivana Kajic. *Computational mechanisms of language understanding and use in the brain and behaviour*. Ph.d. thesis, University of Waterloo, 2020.

[118] Ivana Kajić, Jan Gosmann, Terrence C Stewart, Thomas Wennekers, and Chris Eliasmith. A spiking neuron model of word associations for the remote associates test. *Frontiers in Psychology*, 8:99, 2017.

[119] Andrew S Kayser, Bradley R Buchsbaum, Drew T Erickson, and Mark D'Esposito. The functional anatomy of a perceptual decision in the human brain. *Journal of Neurophysiology*, 103(3):1179–1194, 2010.

[120] Orion P Keifer Jr, Robert C Hurt, Kerry J Ressler, and Paul J Marvar. The physiology of fear: Reconceptualizing the role of the central amygdala in fear learning. *Physiology*, 30(5):389–401, 2015.

[121] Ann Kennedy, Greg Wayne, Patrick Kaifosh, Karina Alviña, LF Abbott, and Nathaniel B Sawtell. A temporal basis for predicting the sensory consequences of motor commands in an electric fish. *Nature Neuroscience*, 17(3):416–422, 2014.

[122] Daniel P Kennedy and Ralph Adolphs. The social brain in psychiatric and neurological disorders. *Trends in Cognitive Sciences*, 16(11):559–572, 2012.

[123] Koosha Khalvati, Roozbeh Kiani, and Rajesh PN Rao. Bayesian inference with incomplete knowledge explains perceptual confidence and its deviations from accuracy. *Nature Communications*, 12(1):5704, 2021.

[124] Koosha Khalvati, Saghar Mirbagheri, Seongmin A Park, Jean-Claude Dreher, and Rajesh P Rao. A bayesian theory of conformity in collective decision making. *Advances in Neural Information Processing Systems*, 32, 2019.

[125] Koosha Khalvati, Seongmin A Park, Saghar Mirbagheri, Remi Philippe, Mariateresa Sestito, Jean-Claude Dreher, and Rajesh PN Rao. Modeling other minds: Bayesian inference explains human choices in group decision-making. *Science Advances*, 5(11), 2019.

[126] Dongbeom Kim, Denis Paré, and Satish S Nair. Mechanisms contributing to the induction and storage of pavlovian fear memories in the lateral amygdala. *Learning & Memory*, 20(8):421–430, 2013.

[127] Brent Komer. *Biologically inspired spatial representation*. Ph.d. thesis, University of Waterloo, 2020.

[128] Brent Komer, Pawel Jaworski, Steven Harbour, Chris Eliasmith, and Travis DeWolf. BatSLAM: Neuromorphic spatial reasoning in 3D environments. In *41st Digital Avionics Systems Conference*, Portsmouth, VA, USA, 2022. DASC.

[129] Brent Komer, Terrence C. Stewart, Aaron R. Voelker, and Chris Eliasmith. A neural representation of continuous space using fractional binding. In *41st Annual Meeting of the Cognitive Science Society*, Montreal, QC, 2019. Cognitive Science Society.

[130] Minoru Koyama and Avinash Pujala. Mutual inhibition of lateral inhibition: A network motif for an elementary computation in the brain. *Current Opinion in Neurobiology*, 49:69–74, 2018.

[131] Sabine Krabbe, Jan Gründemann, and Andreas Lüthi. Amygdala inhibitory circuits regulate associative fear conditioning. *Biological Psychiatry*, 83(10):800–809, 2018.

[132] Franklin B Krasne, Raphael Zinn, Bryce Vissel, and Michael S Fanselow. Extinction and discrimination in a bayesian model of context fear conditioning (BaconX). *Hippocampus*, 31(7):790–814, 2021.

[133] Juri D Kropotov and Susan C Etlinger. Selection of actions in the basal ganglia–thalamocortical circuits: Review and model. *International Journal of Psychophysiology*, 31(3):197–217, 1999.

[134] Arie W Kruglanski and Wolfgang Stroebe. *Handbook of the history of social psychology*. Psychology Press, 2012.

[135] Pinelopi Kyriazi, Drew B Headley, and Denis Pare. Multi-dimensional coding by basolateral amygdala neurons. *Neuron*, 99(6):1315–1328, 2018.

[136] Claus Lamm and Tania Singer. The role of anterior insular cortex in social emotions. *Brain Structure and Function*, 214(5):579–591, 2010.

[137] Christian Lebiere, Terrence Stewart, and Robert West. Applying cognitive architectures to decision-making. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 31, 2009.

[138] Douglas G Lee and Marius Usher. Value certainty in drift-diffusion models of preferential choice. *Psychological Review*, 2021.

[139] Ed S Lein, Michael J Hawrylycz, Nancy Ao, Mikael Ayres, Amy Bensinger, Amy Bernard, Andrew F Boe, Mark S Boguski, Kevin S Brockway, Emi J Byrnes, et al. Genome-wide atlas of gene expression in the adult mouse brain. *Nature*, 445(7124):168, 2007.

[140] Guoshi Li. Computational models of the amygdala in acquisition and extinction of conditioned fear. In *The Amygdala: Where Emotions Shape Perception, Learning and Memories*. InTech, 2017.

[141] Guoshi Li, Satish S Nair, and Gregory J Quirk. A biologically realistic network model of acquisition and extinction of conditioned fear associations in lateral amygdala neurons. *Journal of Neurophysiology*, 101(3):1629–1646, 2009.

[142] Susan Shi Yuan Li and Gavan P McNally. The conditions that promote fear learning: Prediction error and Pavlovian fear conditioning. *Neurobiology of Learning and Memory*, 108:14–21, 2014.

[143] Johan Lind, Magnus Enquist, and Stefano Ghirlanda. Animal memory: A review of delayed matching-to-sample data. *Behavioural Processes*, 117:52–58, 2015.

[144] Shmuel Lissek, Arter L Biggs, Stephanie J Rabin, Brian R Cornwell, Ruben P Alvarez, Daniel S Pine, and Christian Grillon. Generalization of conditioned fear-potentiated startle in humans: Experimental validation and clinical relevance. *Behaviour Research and Therapy*, 46(5):678–687, 2008.

[145] Shmuel Lissek, Antonia N Kaczkurkin, Stephanie Rabin, Marilla Geraci, Daniel S Pine, and Christian Grillon. Generalized anxiety disorder is associated with overgeneralization of classically conditioned fear. *Biological Psychiatry*, 75(11):909–915, 2014.

[146] Shmuel Lissek, Stephanie Rabin, Randi E Heller, David Lukenbaugh, Marilla Geraci, Daniel S Pine, and Christian Grillon. Overgeneralization of conditioned fear as a pathogenic marker of panic disorder. *American Journal of Psychiatry*, 167(1):47–55, 2010.

[147] Shmuel Lissek and Brian van Meurs. Learning models of PTSD: Theoretical accounts and psychobiological evidence. *International Journal of Psychophysiology*, 98(3):594–605, 2015.

[148] Yunzhe Liu, Shiyi Li, Wanjun Lin, Wenxin Li, Xinyuan Yan, Xuena Wang, Xinyue Pan, Robb B Rutledge, and Yina Ma. Oxytocin modulates social value representations in the amygdala. *Nature Neuroscience*, 22(4):633–641, 2019.

[149] Zhiyuan Liu, Sijia Liu, Shuang Li, Lin Li, Li Zheng, Xue Weng, Xiuyan Guo, Yang Lu, Weiwei Men, Jiahong Gao, et al. Dissociating value-based neurocomputation from subsequent selection-related activations in human decision-making. *Cerebral Cortex*, 2022.

[150] Patricia L Lockwood and Miriam C Klein-Flügge. Computational modelling of social cognition and behaviour - a reinforcement learning primer. *Social Cognitive and Affective Neuroscience*, 16(8):761–771, 2021.

[151] Tina B Lonsdorf, Mareike M Menz, Marta Andreatta, Miguel A Fullana, Armita Golkar, Jan Haaker, Ivo Heitland, Andrea Hermann, Manuel Kuhn, Onno Kruse, et al. Don't fear 'fear conditioning': Methodological considerations for the design and analysis of studies on human fear acquisition, extinction, and return of fear. *Neuroscience and Biobehavioral Reviews*, 77:247–285, 2017.

[152] Robert Lount Jr, Chen-Bo Zhong, Niro Sivanathan, and J Murnighan. Getting off on the wrong foot: The timing of a breach and the restoration of trust. *Personality and Social Psychology Bulletin*, 2008.

[153] Elizabeth K Lucas and Roger L Clem. GABAergic interneurons: The orchestra or the conductor in fear learning and memory? *Brain Research Bulletin*, 141:13–19, 2018.

[154] Jun Luo. The neural basis of and a common neural circuitry in different types of pro-social behavior. *Frontiers in Psychology*, 9:859, 2018.

[155] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

[156] David MacNeil and Chris Eliasmith. Fine-tuning and the stability of recurrent neural networks. *PloS one*, 6(9):e22885, 2011.

[157] Michael P. Furlong Madeleine Bartlett, Nicole Sandra-Yaffa t and Terry C. Stewart. Biologically-plausible memory for continuous time reinforcement learning. In *International Conference on Cognitive Modelling (ICCM) 2022*, Toronto, ON, July 24-27 2022.

[158] Francesco Mannella, Stefano Zappacosta, Marco Mirolli, and Gianluca Baldassarre. A computational model of the amygdala nuclei's role in second order conditioning. In *International Conference on Simulation of Adaptive Behavior*, pages 321–330. Springer, 2008.

[159] Henry Markram, Eilif Muller, Srikanth Ramaswamy, Michael W Reimann, Marwan Abdellah, Carlos Aguado Sanchez, Anastasia Ailamaki, Lidia Alonso-Nanclares, Nicolas Antille, Selim Arsever, et al. Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163(2):456–492, 2015.

[160] Andrea Mattera, Marco Pagani, and Gianluca Baldassarre. A computational model integrating multiple phenomena on cued fear conditioning, extinction, and reinstatement. *Frontiers in Systems Neuroscience*, 14:569108, 2020.

[161] Alexander J McDonald. Functional neuroanatomy of the basolateral amygdala: Neurons, neurotransmitters, and circuits. In *Handbook of Behavioral Neuroscience*, volume 26, pages 1–38. Elsevier, 2020.

[162] William McDougall. *An introduction to social psychology*. Psychology Press, 2015.

[163] Stephen B McHugh, Christopher Barkus, Anna Huber, Liliana Capitao, Joao Lima, John P Lowry, and David M Bannerman. Aversive prediction error signals in the amygdala. *Journal of Neuroscience*, 34(27):9024–9033, 2014.

[164] Kevin R McKee, Ian Gemp, Brian McWilliams, Edgar A Duéñez-Guzmán, Edward Hughes, and Joel Z Leibo. Social diversity and social preferences in mixed-motive reinforcement learning. *arXiv preprint arXiv:2002.02325*, 2020.

[165] Microsoft. Neural network intelligence (version 2.9). https://github.com/microsoft/nni, 2021.

[166] Stanley Milgram and Christian Gudehus. *Obedience to authority*. Ziff-Davis Publishing Company New York, NY, 1978.

[167] Marco Mirolli, Francesco Mannella, and Gianluca Baldassarre. The roles of the amygdala in the affective regulation of body, brain, and behaviour. *Connection Science*, 22(3):215–245, 2010.

[168] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[169] Sandra N Moses, Jon M Houck, Tim Martin, Faith M Hanlon, Jennifer D Ryan, Robert J Thoma, Michael P Weisend, Eric M Jackson, Eero Pekkonen, and Claudia D Tesche. Dynamic neural activity recorded from human amygdala during fear conditioning using magnetoencephalography. *Brain Research Bulletin*, 71(5):452–460, 2007.

[170] Roberta C Ramos Mota, Daniel J Rea, Anna Le Tran, James E Young, Ehud Sharlin, and Mario C Sousa. Playing the 'trust game' with robots: Social strategies and experiences. In *25th International Symposium on Robot and Human Interactive Communication*, 2016.

[171] Ahmed A Moustafa, Mark W Gilbertson, Scott P Orr, Mohammad M Herzallah, Richard J Servatius, and Catherine E Myers. A model of amygdala–hippocampal–prefrontal interaction in fear conditioning and extinction in animals. *Brain and Cognition*, 81(1):29–43, 2013.

[172] Martijn J Mulder, Wouter Boekel, Roger Ratcliff, and Birte U Forstmann. Cortico-subthalamic connection predicts individual differences in value-driven choice bias. *Brain Structure and Function*, 219(4):1239–1249, 2014.

[173] Jeff Muller, Keith P Corodimas, Zhanna Fridel, and Joseph E LeDoux. Functional inactivation of the lateral and basal nuclei of the amygdala by muscimol infusion prevents fear conditioning to an explicit conditioned stimulus and to contextual stimuli. *Behavioral Neuroscience*, 111(4):683, 1997.

[174] Peter R Murphy, Evert Boonstra, and Sander Nieuwenhuis. Global gain modulation generates time-dependent urgency during perceptual choice in humans. *Nature Communications*, 7(1):1–15, 2016.

[175] Jinichi Nagumo, Suguru Arimoto, and Shuji Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the Institute of Electrical and Electronics Engineers*, 50(10):2061–2070, 1962.

[176] Wilten Nicola and Claudia Clopath. Supervised learning in spiking neural networks with force training. *Nature Communications*, 8(1):1–15, 2017.

[177] Lauren A O'Connell and Hans A Hofmann. The vertebrate mesolimbic reward system and social behavior network: A comparative synthesis. *Journal of Comparative Neurology*, 519(18):3599–3639, 2011.

[178] Randall C. O'Reilly, Yuko Munakata, Michael J. Frank, Thomas E. Hazy, and Contributors. *Computational Cognitive Neuroscience*. Online Book, 4th Edition, URL: https://github.com/CompCogNeuro/ed4, 2012.

[179] Randall C O'Reilly, Thomas E Hazy, and Seth A Herd. The leabra cognitive architecture: How to play 20 principles with nature. In *The Oxford Handbook of Cognitive Science*, pages 91–116. Oxford University Press New York, 2016.

[180] Philip Pärnamets, Anastasia Shuster, Diego A Reinero, and Jay J Van Bavel. A value-based framework for understanding cooperation. *Current Directions in Psychological Science*, 29(3):227–234, 2020.

[181] Thomas Parr and Karl J Friston. The anatomy of inference: Generative models and brain structure. *Frontiers in Computational Neuroscience*, page 90, 2018.

[182] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau,

M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[183] Sandeep Pendyam, Christian Bravo-Rivera, Anthony Burgos-Robles, Francisco Sotres-Bayon, Gregory J Quirk, and Satish S Nair. Fear signaling in the prelimbic-amygdala circuit: A computational modeling and recording study. *Journal of Neurophysiology*, 110(4):844–861, 2013.

[184] Jan Luca Pletzer, Daniel Balliet, Jeff Joireman, D Michael Kuhlman, Sven C Voelpel, and Paul AM Van Lange. Social value orientation, expectations, and cooperation in social dilemmas: A meta–analysis. *European Journal of Personality*, 32(1):62–83, 2018.

[185] Charlotte Prévost, Daniel McNamee, Ryan K Jessup, Peter Bossaerts, and John P O'Doherty. Evidence for model-based computations in the human amygdala during pavlovian conditioning. *PLoS Computational Biology*, 9(2):e1002918, 2013.

[186] Srikanth Ramaswamy, Jean-Denis Courcol, Marwan Abdellah, Stanislaw R Adaszewski, Nicolas Antille, Selim Arsever, Guy Atenekeng, Ahmet Bilgili, Yury Brukau, Athanassia Chalimourda, et al. The neocortical microcircuit collaboration portal: A resource for rat somatosensory cortex. *Frontiers in Neural Circuits*, 9:44, 2015.

[187] Rajesh PN Rao. Decision making under uncertainty: A neural model based on partially observable Markov decision processes. *Frontiers in Computational Neuroscience*, 4:146, 2010.

[188] Daniel Rasmussen, Aaron Voelker, and Chris Eliasmith. A neural model of hierarchical reinforcement learning. *PloS one*, 12(7):e0180234, 2017.

[189] Rasmussen, Daniel. *Hierarchical reinforcement learning in a biologically plausible neural architecture*. PhD thesis, University of Waterloo, 2014.

[190] Roger Ratcliff and Gail McKoon. The diffusion decision model: Theory and data for two-choice decision tasks. *Neural Computation*, 20(4):873–922, 2008.

[191] Roger Ratcliff, Philip L Smith, Scott D Brown, and Gail McKoon. Diffusion decision model: Current issues and history. *Trends in Cognitive Sciences*, 20(4):260–281, 2016.

[192] Peter Redgrave, Tony J Prescott, and Kevin Gurney. The basal ganglia: A vertebrate solution to the selection problem? *Neuroscience*, 89(4):1009–1023, 1999.

[193] Reed L Ressler and Stephen Maren. Synaptic encoding of fear memories in the amygdala. *Current Opinion in Neurobiology*, 54:54–59, 2019.

[194] Shawn A Rhoads, Jo Cutler, and Abigail A Marsh. A feature-based network analysis and fMRI meta-analysis reveal three distinct types of prosocial decisions. *Social Cognitive and Affective Neuroscience*, 16(12):1214–1233, 2021.

[195] James K Rilling and Alan G Sanfey. The neuroscience of social decision-making. *Annual Review of Psychology*, 62(1):23–48, 2011.

[196] Ian D Roberts and Cendri A Hutcherson. Affect and decision making: Insights and predictions from computational models. *Trends in Cognitive Sciences*, 23(7):602–614, 2019.

[197] David A Robinson. Integrating with neurons. *Annual Review of Neuroscience*, 12(1):33–45, 1989.

[198] Morgan M Rogers-Carter and John P Christianson. An insular view of the social decision-making network. *Neuroscience and Biobehavioral Reviews*, 103:119–132, 2019.

[199] Edmund T Rolls, Wei Cheng, and Jianfeng Feng. The orbitofrontal cortex: Reward, emotion and depression. *Brain Communications*, 2(2):fcaa196, 2020.

[200] Edmund T Rolls and Fabian Grabenhorst. The orbitofrontal cortex and beyond: From affect to decision-making. *Progress in neurobiology*, 86(3):216–244, 2008.

[201] Charly V Rousseau, Guillaume P Dugué, Andréa Dumoulin, Enrico Mugnaini, Stéphane Dieudonné, and Marco A Diana. Mixed inhibitory synaptic balance correlates with glutamatergic synaptic phenotype in cerebellar unipolar brush cells. *Journal of Neuroscience*, 32(13):4632–4644, 2012.

[202] Alex Roxin. Drift–diffusion models for multiple-alternative forced-choice decision making. *The Journal of Mathematical Neuroscience*, 9(1):1–23, 2019.

[203] Rachael D Rubin, Patrick D Watson, Melissa C Duff, and Neal J Cohen. The role of the hippocampus in flexible cognition and social behavior. *Frontiers in Human Neuroscience*, 8:742, 2014.

[204] David E Rumelhart, James L McClelland, PDP Research Group, et al. *Parallel distributed processing*, volume 1. IEEE New York, 1988.

[205] Tessa Rusch, Saurabh Steixner-Kumar, Prashant Doshi, Michael Spezio, and Jan Gläscher. Theory of mind and decision science: Towards a typology of tasks and computational models. *Neuropsychologia*, 146:107488, 2020.

[206] Marco J Russo, Enrico Mugnaini, and Marco Martina. Intrinsic properties and mechanisms of spontaneous firing in mouse cerebellar unipolar brush cells. *The Journal of Physiology*, 581(2):709–724, 2007.

[207] Rinki Saha, Stephanie Knapp, Darpan Chakraborty, Omer Horovitz, Anne Albrecht, Martin Kriebel, Hanoch Kaphzan, Ingrid Ehrlich, Hansjürgen Volkmer, and Gal Richter-Levin. GABAergic synapses at the axon initial segment of basolateral amygdala projection neurons modulate fear extinction. *Neuropsychopharmacology*, 42(2):473–484, 2017.

[208] Toshiyuki Sawaguchi and Patricia S Goldman-Rakic. The role of D1-dopamine receptor in working memory: Local injections of dopamine antagonists into the prefrontal cortex of rhesus monkeys performing an oculomotor delayed-response task. *Journal of Neurophysiology*, 71(2):515–528, 1994.

[209] Michael A Schwemmer, Adrienne L Fairhall, Sophie Denéve, and Eric T Shea-Brown. Constructing precisely computing networks with biophysical spiking neurons. *Journal of Neuroscience*, 35(28):10112–10134, 2015.

[210] Sugandha Sharma. Neural plausibility of Bayesian inference. Masters thesis, University of Waterloo, 2018.

[211] Jianghong Shi, Bryan Tripp, Eric Shea-Brown, Stefan Mihalas, and Michael A. Buice. Mousenet: A biologically constrained convolutional neural network model for the mouse visual cortex. *PLoS Computational Biology*, 18(9):e1010427, 2022.

[212] James M Shine. The thalamus integrates the macrosystems of the brain to facilitate complex, adaptive brain network dynamics. *Progress in Neurobiology*, 199:101951, 2021.

[213] Demetrio Sierra-Mercado, Nancy Padilla-Coreano, and Gregory J Quirk. Dissociable roles of prelimbic and infralimbic cortices, ventral hippocampus, and basolateral amygdala in the expression and extinction of conditioned fear. *Neuropsychopharmacology*, 36(2):529–538, 2011.

[214] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[215] Adam Sobieszek and Tadeusz Price. Playing games with Ais: The limits of GPT-3 and similar large language models. *Minds and Machines*, pages 1–24, 2022.

[216] Michael W Spratling. A review of predictive coding algorithms. *Brain and Cognition*, 112:92–97, 2017.

[217] Kartik K Sreenivasan, Clayton E Curtis, and Mark D'Esposito. Revisiting the role of persistent neural activity during working memory. *Trends in Cognitive Sciences*, 18(2):82–89, 2014.

[218] Dominic Standage, Hongzhi You, DaHui Wang, and Michael C Dorris. Gain modulation by an urgency signal controls the speed–accuracy trade-off in a network model of a cortical decision circuit. *Frontiers in Computational Neuroscience*, 5, 2011.

[219] Terrence C Stewart, Xuan Choo, and Chris Eliasmith. Dynamic behaviour of a spiking model of action selection in the basal ganglia. In *Proceedings of the 10th International Conference on Cognitive Modeling*, pages 235–40, 2010.

[220] Terrence C Stewart, Yichuan Tang, and Chris Eliasmith. A biologically realistic cleanup memory: Autoassociation in spiking neurons. *Cognitive Systems Research*, 12(2):84–92, 2011.

[221] Mark Steyvers, Michael D Lee, and Eric-Jan Wagenmakers. A bayesian analysis of human decision-making on bandit problems. *Journal of Mathematical Psychology*, 53(3):168–179, 2009.

[222] Andrea Stocco, Christian Lebiere, and John R Anderson. Conditional routing of information to the cortex: A model of the basal ganglia's role in cognitive coordination. *Psychological Review*, 117(2):541, 2010.

[223] Andreas Stöckel. *Harnessing Neural Dynamics as a Computational Resource*. Ph.d. thesis, University of Waterloo, 1 2022.

[224] Andreas Stöckel and Chris Eliasmith. Passive nonlinear dendritic interactions as a general computational resource in functional spiking neural networks. *arXiv preprint arXiv:1904.11713*, 2019.

[225] Andreas Stöckel and Chris Eliasmith. Computational properties of multi-compartment LIF neurons with passive dendrites. *Neuromorphic Computing and Engineering*, 2022.

[226] Andreas Stöckel, Terrence C Stewart, and Chris Eliasmith. Connecting biological detail with neural computation: Application to the cerebellar granule–golgi microcircuit. *Topics in Cognitive Science*, 13(3):515–533, 2021.

[227] D Sulzer and S Rayport. Dale's principle and glutamate corelease from ventral midbrain dopamine neurons. *Amino Acids*, 19(1):45–52, 2000.

[228] Susan M Sunkin, Lydia Ng, Chris Lau, Tim Dolbeare, Terri L Gilbert, Carol L Thompson, Michael Hawrylycz, and Chinh Dang. Allen brain atlas: An integrated spatio-temporal portal for exploring the central nervous system. *Nucleic Acids Research*, 41(D1):D996–D1008, 2012.

[229] David Sussillo and Larry F Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009.

[230] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[231] Shinsuke Suzuki and John P O'Doherty. Breaking human social decision making into multiple components and then putting them together again. *Cortex*, 127:221–230, 2020.

[232] Corinne Teeter, Ramakrishnan Iyer, Vilas Menon, Nathan Gouwens, David Feng, Jim Berg, Aaron Szafer, Nicholas Cain, Hongkui Zeng, Michael Hawrylycz, et al. Generalized leaky integrate-and-fire models classify multiple neuron types. *Nature Communications*, 9(1):1–15, 2018.

[233] Alexandr Ten, Pramod Kaushik, Pierre-Yves Oudeyer, and Jacqueline Gottlieb. Humans monitor learning progress in curiosity-driven exploration. *Nature Communications*, 12(1):1–10, 2021.

[234] Sylvia Terbeck, Julian Savulescu, Laurence P Chesterman, and Philip J Cowen. Noradrenaline effects on social behaviour, intergroup relations, and moral decisions. *Neuroscience and Biobehavioral Reviews*, 66:54–60, 2016.

[235] Dominik Thalmeier, Marvin Uhlmann, Hilbert J Kappen, and Raoul-Martin Memmesheimer. Learning universal computations with spikes. *PLoS Computational Biology*, 12(6):e1004895, 2016.

[236] Isabel Thielmann, Giuliana Spadaro, and Daniel Balliet. Personality and prosocial behavior: A theoretical framework and meta-analysis. *Psychological Bulletin*, 146(1), 2020.

[237] Robert Thomson, Christian Lebiere, John R Anderson, and James Staszewski. A general instance-based learning framework for studying intuitive decision-making in a cognitive architecture. *Journal of Applied Research in Memory and Cognition*, 4(3), 2015.

[238] Mattie Tops, Sascha Russo, Maarten AS Boksem, and Don M Tucker. Serotonin: Modulator of a drive to withdraw. *Brain and Cognition*, 71(3):427–436, 2009.

[239] Sébastien Tremblay, KM Sharika, and Michael L Platt. Social decision-making and the brain: A comparative perspective. *Trends in Cognitive Sciences*, 21(4):265–276, 2017.

[240] Elizabeth Tricomi, Antonio Rangel, Colin F Camerer, and John P O'Doherty. Neural evidence for inequality-averse social preferences. *Nature*, 463(7284):1089–1091, 2010.

[241] Norman Triplett. The dynamogenic factors in pacemaking and competition. *The American Journal of Psychology*, 9(4):507–533, 1898.

[242] Brandon M Turner, Leendert Van Maanen, and Birte U Forstmann. Informing cognitive abstractions through neuroimaging: The neural drift diffusion model. *Psychological Review*, 122(2):312, 2015.

[243] Anita Tusche and Lisa M Bas. Neurocomputational models of altruistic decision-making and social motives: Advances, pitfalls, and future directions. *Wiley Interdisciplinary Reviews: Cognitive Science*, 12(6):e1571, 2021.

[244] Eric van Dijk and Carsten KW De Dreu. Experimental games and social decision making. *Annual Review of Psychology*, pages 415–438, 2021.

[245] Debora Vansteenwegen, Bram Vervliet, Dirk Hermans, Tom Beckers, Frank Baeyens, and Paul Eelen. Stronger renewal in human fear conditioning when tested with an acquisition retrieval cue than with an extinction retrieval cue. *Behaviour Research and Therapy*, 44(12):1717–1725, 2006.

[246] Roberto Vazquez. Izhikevich neuron model and its application in pattern recognition. *Australian Journal of Intelligent Information Processing Systems*, 11(1):35–40, 2010.

[247] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[248] Ioannis Vlachos, Cyril Herry, Andreas Lüthi, Ad Aertsen, and Arvind Kumar. Context-dependent encoding of fear and extinction memories in a large-scale network model of the basal amygdala. *PLoS Computational Biology*, 7(3):e1001104, 2011.

[249] Aaron Russell Voelker. A solution to the dynamics of the prescribed error sensitivity learning rule. Technical report, University of Waterloo, Centre for Theoretical Neuroscience, 2015.

[250] Aaron Russell Voelker. *Dynamical systems in spiking neuromorphic hardware*. Ph.d. thesis, University of Waterloo, 2019.

[251] Aaron Russell Voelker, Eric Crawford, and Chris Eliasmith. Learning large-scale heteroassociative memories in spiking neurons. *Unconventional Computation and Natural Computation*, 7:2014, 2014.

[252] Mei Wang, Weihong Deng, Jiani Hu, Xunqiang Tao, and Yaohai Huang. Racial faces in the wild: Reducing racial bias by information maximization adaptation network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 692–702, 2019.

[253] Weixun Wang, Jianye Hao, Yixi Wang, and Matthew Taylor. Towards cooperation in sequential prisoner's dilemmas: A deep multiagent reinforcement learning approach. *arXiv:1803.00162*, 2018.

[254] Michael L. Waskom. Seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

[255] K Geoffrey White. Forgetting functions. *Animal Learning and Behavior*, 29(3):193–207, 2001.

[256] Hugh R Wilson. Simplified dynamics of human and mammalian neocortical neurons. *Journal of Theoretical Biology*, 200(4):375–388, 1999.

[257] Y Yau, M Dadar, M Taylor, Y Zeighami, LK Fellows, P Cisek, and A Dagher. Neural correlates of evidence and urgency during human perceptual decision-making in dynamically changing conditions. *Cerebral Cortex*, 30(10):5471–5483, 2020.

[258] Matthew A Zelgen. Amygdala modeling with context and motivation using spiking neural networks for robotics applications. Technical report, Wright State University, 2022.

[259] Zhuangdi Zhu, Kaixiang Lin, Anil K Jain, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *arXiv preprint arXiv:2009.07888*, 2020.

APPENDICES

Appendix A

Challenges in Functional Neuroanatomy

In Sec. 1.3, we reviewed the functional neuroanatomy of the human brain. In researching and summarizing the literature on this subject, we repeatedly encountered several challenges that arise when studying the functional organization of the brain. We are not the first to note these challenges, and this is not an exhaustive list, but they are important concerns for researchers who wish to build anatomically-grounded models of cognition.

A.1 Discrete Regions

Dividing the brain into discrete functional regions is conceptually useful but practically difficult. The boundaries between brain areas, such as the cingulate cortex or orbitofrontal cortex, are less obvious than the boundaries between bodily organs, both anatomically and functionally. Our conventions for separating and labelling these regions have evolved significantly over the last few centuries. Early neuroanatomist had limited access to functioning brains, so they classified the brain according to the physical structures that were obvious upon dissection. Neuroanatomy improved as neuroscientists began to classify the cell types and neurotransmitters used in various areas, and as they more thoroughly mapped the connectivity within and between regions. Recently, imaging techniques and electrophysiological studies have further subdivided the larger brain regions (especially the cerebral cortex) into smaller components, which are labelled according to their spatial location in a 3D coordinate system (dorsal vs. ventral, rostral vs. caudal, and lateral vs. medial) or

their relation to particular ridges and folds (gyri and sulci). Current neuroanatomy is thus based on a wide variety of empirical data, but naming conventions and regional boundaries may also reflect historical quirks.

With regards to functional modelling, modern imaging techniques (fMRI) and data analysis tools (applied to datasets containing thousands of individuals) can be used to identify functional differences between volumes of brain tissue. Unfortunately, these regions do not always align with previously established anatomical boundaries. For example, the valuation of potential actions is critical to decision making, but many brain areas (discussed below) seem to be involved in representing these values. Indeed, some of these areas, such as the ventromedial prefrontal cortex, the dorsomedial prefrontal cortex, and the rostral anterior cingulate cortex, are adjacent to one another and are biologically comparable. In cases like these, the boundaries that anatomists have drawn between brain regions may impede modern efforts to functionally subdivide the brain. For these reasons, we assume that the neural representation of high-level concepts (such as action values) occurs within diffuse regions of the brain (such as the prefrontal and cingulate cortices) that only partially align with neuroanamotical standards. We avoid describing the exact spatial boundaries of these regions, but, in order to connect our work with the neuroscience literature and broader modeling community, we still reference the existing neuroanatomical classifications when possible. We hope that the conventions for categorizing brain areas become more flexible and functionally-oriented in the future.

A.2 Ascribing Representation

Ascribing specific representations or computations to brain regions is challenging. While we firmly believe that neural activity reflects the encoding, transformation, and decoding of information in a cognitive system, it is difficult for external observers to infer the exact content of these representations. Most neurons are sensitive to various external and internal features: their activities will change as the properties of those features change. For instance, color-sensitive neurons in visual cortex will spike more frequently as the intensity or frequency of incident light increases, while motor neurons will fire more frequently when the brain requires faster contraction of particular muscles.

The challenge with attributing representations arises when describing the internal representations used by brain regions that lie far from the sensory or motor periphery. Neural activities in these regions correlate with both external features and patterns of neural activity in other brain regions, indicating that the represented information is a high-dimensional mixture of external information and internal evaluations. To know exactly what such a

neuron "represents", we would have to enumerate all the variables that it might be sensitive to. Given the complexity of the brain's inputs, the sophistication of its cognition, and the likelihood that different individuals learn different representations over the course of their lives, it seems unlikely that scientists will ever describe these rich representations with perfect accuracy. In the remainder of this thesis, when we describe neurons as 'representing' a quantity (like "value" or "trust"), we mean that those neurons have varying sensitivities to numerous ineffable features related to that quantity (among others), and recognize that our coarse linguistic descriptions and low-dimensional mathematical descriptions are merely convenient conceptual and modelling abstractions.

A.3 Functional Multiplicity and Redundancy

Another difficulty with functional neuroanatomy is that multiple brain areas appear to do roughly the same thing. We discussed earlier how this reflects the somewhat arbitrary divisions that neuroanatomists draw between regions of the brain. However, there are many cases in which two or more brain areas are unequivocally distinct from one another (e.g., the striatum and the ventromedial prefrontal cortex) but perform similar cognitive operations (e.g., valuation). Why would the brain bother with such redundancy? We see two possibilities. First, redundancy is a failsafe against brain damage: intact brain areas may, through plasticity, cover for functionally-similar areas that have ceased to operate properly. Second, although multiple areas may appear to represent similar information and perform similar cognitive roles, it is quite likely that they do so in subtly different ways, or in slightly different domains. For example, we later discuss below how social learning and decision making appear to involve two distinct networks, one situated primarily in the cortex; and another that includes only subcortical structures. Redundancies like these may reflect the evolution of advanced structures and circuits in humans (and other higher mammals) to complement the simpler functionality of other structures. While evolutionarily older circuits in subcortical areas may mediate simpler forms of social learning (e.g., determining whether to approach or avoid an object given previous experience with associated pain, food, or sex), newer circuits in cortex may mediate more sophisticated forms of social learning (e.g., using high-dimensional representations of social relations to plan goal-directed behavior or predict the behavior of others). Thus, distinct functional systems in the brain may only be "redundant" at a superficial level, and a more thorough understanding of their functional domains may reveal important differences.

A.4 Controlled Experiments

The dynamic, integrated nature of the embodied brain poses yet another challenge for functional neuroanatomy. As in every natural science, cognitive neuroscience develops through an iterated process of experimentation, observation, and theorizing. This method works best when experiments can isolate individual variables (or processes) and control for the rest. Unfortunately, because the brain is always engaged in numerous cognitive processes, it is rarely possible to conduct an experiment in which all neural processes are held constant, save for a single process that relates to the experiment being performed. Whats more, individual differences between brains means that even a tightly-controlled experiment will engage different representations and prompt different patterns of thought and emotion in each individual. In order to draw broad conclusions about social cognition, theorists need access to a wide variety of social experiments that control and manipulate numerous social features in diverse social contexts.

Fortunately, recent meta analyses have identified several features of social experiments that reliably correlate with different neural activities or behaviors. Rhoads et al. [194] used an unsupervised approach to identify three low-level features in social tasks that relate to an individual's prosocial behavior; that is, three factors that strongly influence whether an individual will choose actions that increase the welfare of a beneficiary. These include (a) information about the beneficiary of an individual's action, such whether they are a real person or a computer agent, whether they are personally known to the individual, and whether they have demonstrated a need for resources; (b) the form of interaction between the two individuals, for example whether the beneficiary also make decisions, and whether repeated interaction will occur; and (c) the outcome of the interaction, including the probability or magnitude of reward delivered to both individuals, and whether the action violates social norms. The authors show how these features determine an individual's likelihood of choosing a prosocial action, which were clustered into three categories: cooperative and strategic decisions, where outcomes depend on the decisions of others; equitable and norm-compliant decisions, in which one individual unilaterally decides how to allocate resources, subject to social norms; and altruistic decisions, where an individual unilaterally decides whether to forgo resources and benefit others. For each of these clusters, the authors note which of the aforementioned social features are relevant, identify social experiments that engage this type of behavior, and examine which neuroanatomical areas are activated while performing these experiments.

Similarly, Rusch et al. [205] investigate which dimensions of a social experiment correlate with an individual's tendency to empathize and mentalize, or engage in mental processes that simulate the emotional and/or rational thought processes of other individuals.

They identify two important features. The first is uncertainty and information asymmetry, or the extent to which one individual has greater information or predictive capability (with respect to the task) than another. The authors hypothesize that, if another person has greater knowledge about the (physical or social) environment and demonstrates an enhanced ability to predict it, an individual is motivated to represent that person's emotional state or belief state. Empathizing and mentalizing in this way allows an individual to internalize the mental capabilities of another individual and model the world more effectively, leading to better decisions. However, this is only useful when the environment has a certain degree of uncertainty or asymmetry; if the agent is already certain about how the world will unfold, or already knows as much as another individual, there is no motivation to spend energy empathizing or mentalizing; and if the world is so complex as to appear random, then neither agent can predict it, and there is also no motivation. The second relevant factor for theory of mind is interactivity, which describes the type and extent of involvement between two individuals. Interactivity encompasses social distance, personal relevance, and task dependent consequences; it thus bears many similarities to the second feature identified in [194]. The authors hypothesize that tasks with greater interactivity should prompt more empathy and mentalizing, so an individual can accurately respond in real time. As in the above paper, Rusch et al. classify traditional social experiments according to these dimensions, and show how these features predict an individual's tendency to empathize and mentalize; they further relate this tendency to functional neuroanatomy and to computational models.

These two analyses illustrate the challenges of designing social experiments and demonstrate that seemingly minor experimental parameters may engage significantly different mental representations and processes. We believe that these sorts of meta-analyses help theorists develop unified theories of social cognition that account for a wide range of cognitive interactions and generalize across the idiosyncrasies of individual experiments.

A.5 Vague Terminology

Finally, the terminology used when describing social learning and decision making is often imprecise. While we should not expect every scientist to adopt an identical notion of high-level psychological concepts, it is still important to avoid over-generalization. One relevant example is the term "altruism": some people use this word to refer to a broad class of actions that recognize or benefit others; while other people use the word to describe a narrow set of actions that benefit others at the explicit expense of one's personal wellbeing. We prefer the latter definition, which is more in line with the classification scheme of Rhoads

et al. [194]; prosocial behavior is the wider class of actions, which may be strategic, norm-compliant, or altruistic. In this definition, altruistic actions have no possible personal benefit: they do not prompt reciprocal cooperation, and do not avoid punishment for violating social norms. The only sense in which such altruism may be personally beneficial is the so-called "warm glow", a subjectively-positive sensation associated with selfless acts. As we gain a wider understanding of the rich cognitive operations that underlie prosocial behavior, we will hopefully develop a more refined terminology that identifies and locates the particular computations involved.