UNIVERSITY OF CALIFORNIA SAN DIEGO

Towards understanding diseases of memory and learning: Artificial neural network models of memory formation, manipulation, and disease in the brain

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Neurosciences

by

Ben Tsuda

Committee in charge:

Professor Terrence J. Sejnowski, Chair
Professor Maxim Bazhenov
Professor John T. Serences
Professor Tatyana O. Sharpee
Professor Kay M. Tye

2022

The Dissertation of Ben Tsuda is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

DEDICATION

To my family, colleagues, and friends.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

There are so many components that went into this doctorate without which it would not have been possible.

My family and friends have supported me through thick and thin, and I cannot thank them enough. I am the son of two scientists, and my mother, Eva, and father, Akira, have not only provided me constant inspiration and models of what a life of science should be, but have always supported and guided me, whether it be in discussions of new scientific ideas or of how to cook a proper gulyas stew. My brothers, Taro and Kenta, are an unwavering support, always inspiring me and stoking my fire of enthusiasm. I have always looked to them for guidance.

I thank my wife, Kirsten. She is amazing in so many ways, and I was fortunate to benefit from her sagely guidance and balancing perspective having recently gone through the trials of her own PhD. Her support contributed so much to the entire process of this doctorate.

I thank my son, Matyas, who has changed how I see everything. His energy and curiosity is so boundless it fills everyone else in his presence.

I thank my many close friends who have provided so much joy in their friendship and care over the years of this dissertation and beyond. It turns out that social networks are a critical component to fruitful, fulfilling, productive investigation of neural networks.

I deeply thank my supervisor and mentor, Terry Sejnowski, whose unbounded understanding and breadth and depth of knowledge was invaluable in guiding me at all times, in science and in life. Our regular discussions not only helped guide and clarify my thought, but always left me inspired, eager to tackle the largest unsolved mysteries of the brain.

I thank my lab group — Computational Neurobiology Laboratory or CNL as we are known. CNL is a historic research group and I have been so fortunate to complete my doctorate there among titans of past, present, and future no doubt. Our boisterous, sometimes meandering group meetings were always colorful and exposed us all to so many things we would never have seen otherwise. In particular I thank Robert Kim, Yusi Chen, Stefan Pate, Mariam Ordyan, Nuttida Rungratsameetaweemana, Ramona Marchand, Jorge Aldana, Gerald Pao, Mohammad

Samavat, Margot Wagner, Don Spencer, Tom Bartol, Claudia Lainscsek, Roger Bingham, David Peterson, Leif Gibb, Sara Sameni, Jason Fleischer, Rachel Mendelsohn, Chris Gonzalez, Aaron Sampson, Joel Yancey, Homero Esmeraldo, Lupe Garcia.

I thank my collaborators, Kay Tye, Hava Siegelmann, and Barry Richmond, who have been inspirational and a joy to interact and work with. I have learned enormous amounts from them and they have caused me many beneficial shifts in perspective - one of the ultimate praises in my opinion.

I thank my committee members, Maxim Bazhenov, John Serences, Tatyana Sharpee, and Kay Tye, who have helped me steer through the mist and fog of science. Their guidance and support provided encouragement and motivation throughout.

I also thank my environment — the Salk Institute, the Neurosciences and Medical Scientist Training Program graduate programs of which I am a part, the greater UCSD community, and the San Diego area itself. I cannot imagine a better environment to complete one's doctorate, and I have been deeply shaped not only by the academic endeavours but also by the community and nature around me. As a native of central Massachusetts, it has been dreamlike to wake up and enjoy a coffee during a sunrise stroll on the beach, work with friends and brilliant minds during the day, hop off to soccer training with a competitive club team, then wrap up the day with a sunset beach barbeque.

Chapter 2, in full, is a reprint of the material as it appears in: Ben Tsuda, Kay M. Tye, Hava T. Siegelmann, and Terrence J. Sejnowski. A modeling framework for adaptive lifelong learning with transfer and savings through gating in the prefrontal cortex. *Proceedings of the National Academy of Sciences*, 117 (47), 29872-29882, 2020. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in full, is a reprint of the material as it appears in: Ben Tsuda, Barry J. Richmond, and Terrence J. Sejnowski. Understanding strategy differences between humans and monkeys with recurrent neural networks. *In preparation*, 2022. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in full, is a reprint of the material as it appears in: Ben Tsuda, Stefan C. Pate, Kay M. Tye, Hava T. Siegelmann, and Terrence J. Sejnowski. Neuromodulators generate multiple context-relevant behaviors in a recurrent neural network by shifting activity hypertubes. *In submission*, 2022. The dissertation author was the primary investigator and author of this paper.

VITA

2012	Bachelor of Arts, Harvard University
2013	Master of Philosophy, University of Cambridge
2013–2016	Associate Computational Biologist, Broad Institute of MIT and Harvard
2022	Doctor of Philosophy, University of California San Diego

PUBLICATIONS

Tsuda B, Sejnowski TJ. The hippocampus as a spatiotemporal autoencoder. *In preparation*, 2022.

Tsuda B, Richmond BJ, Sejnowski TJ. Understanding strategy differences in humans and monkeys with recurrent neural networks. *In preparation*, 2022.

Tsuda B, Pate SC, Tye KM, Siegelmann HT, Sejnowski TJ. Neuromodulators generate multiple context-relevant behaviors in a recurrent neural network by shifting activity hypertubes. *In submission*, 2022.

Trujillo CA, Adams JW, Negraes PD, Carromeu C, Tejwani L, Acab A, Tsuda B, Thomas CA, Sodhi N, Fichter KM, Romero S, Zanella F, Sejnowski TJ, Ulrich H, Muotri AR. Pharmacological reversal of synaptic and network pathology in human MECP2-KO neurons and cortical organoids. *EMBO Molecular Medicine*, 2020.

Tsuda B, Tye KM, Siegelmann HT, Sejnowski TJ. A modeling framework for adaptive lifelong learning with transfer and savings through gating in the prefrontal cortex. *Proceedings of the National Academy of Sciences* 117 (47), 29872-29882, 2020.

Majithia AR, Tsuda B, Agostini M, Gnanapradeepan K, Rice R, Peloso G, Kashyap PA, Zhang X, Broekema MF, Patterson N, Duby M, Sharpe T, Kalkhoven R, Rosen ED, Barroso I, Ellard S, Kathiresan S, O'Rahilly S, Chatterjee K, Florez JC, Mikkelsen T, Savage D, Altshuler D. Prospective functional classification of all possible missense variants in PPARG. *Nature Genetics* 48 (12), 1570-1575, 2016.

Busskamp V, Krol J, Nelidova D, Daum J, Szikra T, Tsuda B, Juttner J, Farrow K, Scherf BG, Alvarez CPP, Genoud C, Sothilingam V, Tanimoto N, Stadler M, Seeliger M, Stoffel M, Filipowicz W, Roska B. miRNAs 182 and 183 are necessary to maintain adult cone photoreceptor outer segments and visual function. *Neuron* 83 (3), 586-600, 2014.

Arda HE, Taubert S, MacNeil LT, Conine CC, Tsuda B, Van Gilst M, Sequerra R, Doucette-

Stamm L, Yamamoto KR, Walhout AJ. Functional modularity of nuclear hormone receptors in a Caenorhabditis elegans metabolic gene regulatory network. *Molecular Systems Biology* 6 (1), 367, 2010.

ABSTRACT OF THE DISSERTATION

Towards understanding diseases of memory and learning: Artificial neural network models of memory formation, manipulation, and disease in the brain

by

Ben Tsuda

Doctor of Philosophy in Neurosciences

University of California San Diego, 2022

Professor Terrence J. Sejnowski, Chair

Human memory and learning represent the most complex and miraculous of human abilities and also the most difficult and major challenges of modern medicine. While treatments for diseases from heart to lung have rapidly advanced, basic mechanisms of brain function rooted in memory and learning remain completely unknown, reflected by the dearth of targeted treatments available for neuropsychiatric disorders.

This dissertation seeks to understand the neural processes of memory and learning at three interdependent levels of investigation — systems of circuits, an individual circuit, and molecular modulation of circuits — and how they can become disrupted in disease. To study these processes, I use artificial neural network models.

I first consider a system of circuits (Chapter 2), studying how regions of the brain coordinate to produce coherent and flexible behavior and how disruptions lead to specific clinical syndromes. I focus on the prefrontal cortex where distinct subregions have been characterized, and analyze how simple architectural alterations in circuit interdependencies can give rise to powerful memory and learning features characteristic of humans. I then study how disruptions affect this circuit system, revealing striking parallels to patients who have suffered specific prefrontal lesions.

To understand how an individual circuit learns and develops, I then focus on the internal process of strategy acquisition and evolution within a neural network (Chapter 3). To do this, I investigate a surprising experimental finding that humans and monkeys exhibit major behavioral discrepancies when solving a set of working memory tasks. I find that neural network models follow characteristic strategy progressions, exhibiting both monkey-like and human-like behavior at different stages. Furthermore, neural networks can mimic common problem solving heuristics while operating with unrelated underlying mechanisms. Counterintuitively, I find that some non-optimal heuristics can act as catalytic strategies that accelerate optimal learning.

These investigations raise the question of how highly structured networks of neurons, like brains, are able to store a multitude of versatile behaviors. In Chapter 4, I investigate how a critical class of regulatory molecules in the brain called neuromodulators alter network properties to enable multiplexing of neural circuits. I reveal how neuromodulators may underlie idiosyncratic circuit dose-response properties, suggesting an alternative explanation for the large variability of neuropsychiatric drug sensitivities observed in patients.

Through these three levels of investigation, I aim to contribute to our understanding of how systems, circuits, and molecules help store and manipulate forms of neural memory to generate the complex and dynamic behaviors characteristic of humans and animals. My findings suggest how these processes relate to disease manifestations and begin to suggest possible avenues toward the ultimate goal: treatments for patients.

Chapter 1

Introduction

The brain has been and remains one of the greatest mysteries. The three pound gelatinous mass is the most sophisticated computational device we know of, and despite a constant stream of new and exciting discoveries, many major questions remain unanswered. In one sense this presents a fascinating and exhilarating frontier — a gold rush for discovery, — but in another sense, this lack of knowledge is a source of human suffering derived from our inability to effectively treat many ailments of the brain. Many pharmacologic treatments for psychiatric disorders (e.g. lithium, chlorpromazine, iproniazid) are adapted versions of drugs discovered serendipitously in the 1950s, and have mechanisms of action and side effects that still remain unexplained [Hym13]. This dissertation aims to contribute to the study of fundamental aspects of memory and learning in the brain, in order to support the effort of developing effective treatments.

To understand the properties of the brain good models are required. For millenia animal models have been used to learn about human biology, extending from experiments in nematodes to monkeys. Biological model systems have provided great insight into many aspects of the brain from the molecular and cellular components that make up the system to the connectivity patterns that interconnect regions of the brain. In the past few decades another important model has emerged to study the brain: artificial neural networks (ANNs). ANNs are computational models that consist of simplified neural units, linked to each other through weighted connections.

Mimicking the brain, ANNs are massively parallelized computing devices [RMP86] and have proven to be exceptionally powerful both for function approximation in machine learning applications [Sch+20; Jum+21; He+16; RFB15] and as insightful models of neural computation [Wan+18; KS20; Man+13; Yan+19; Rus+18]. Although simplified compared to biological neural networks, ANNs allow detailed analysis of how parallel distributed processing can occur, providing a unique lens for studying neural processes. The studies in this dissertation employ ANNs to study aspects of neural behavior.

The brain must be studied across levels. In the work presented here, the focus begins with prefrontal cortex (PFC). Yet it quickly becomes evident that understanding the PFC requires some understanding of the tightly connected basal ganglia, as well as the hippocampus, premotor regions, and thalamus, each of which in turn can only be properly understood with some understanding of regions tightly connected to them. While it is often desirable to simplify systems and reduce them to smaller, discrete components that seem more manageable to understand, systems of the brain cannot be fully understood without understanding their relation to other parts of the brain. This holistic approach includes consideration of both dependencies across areas of the brain (PFC, basal ganglia, hippocampus, etc.), and also dependencies across levels of investigation (Figure 1.1). Principles of how the brain works at the circuit level must be understood in relation to how individual neurons behave, how proteins and ions within those neurons interact (deeper levels) and also how those circuits combine into a larger architecture, and how that architecture itself can be differentially activated (higher levels).

This dissertation is a dive through multiple levels of investigation in understanding memory and learning and how they can be modified in disease or with treatment (Figure 1.1B–D). Chapter 2 begins at the level of systems of circuits (Figure 1.1B) with a study on how regions of the brain, specifically within the prefrontal cortex and basal ganglia, can coordinate to give rise to coherent mental schemas in an always-varying world. This study introduces a new question: what does it mean for even one part of the prefrontal cortex, i.e., one circuit, to store a "strategy?" Chapter 3 moves to this deeper level of investigation — a single circuit (Figure 1.1C) — to

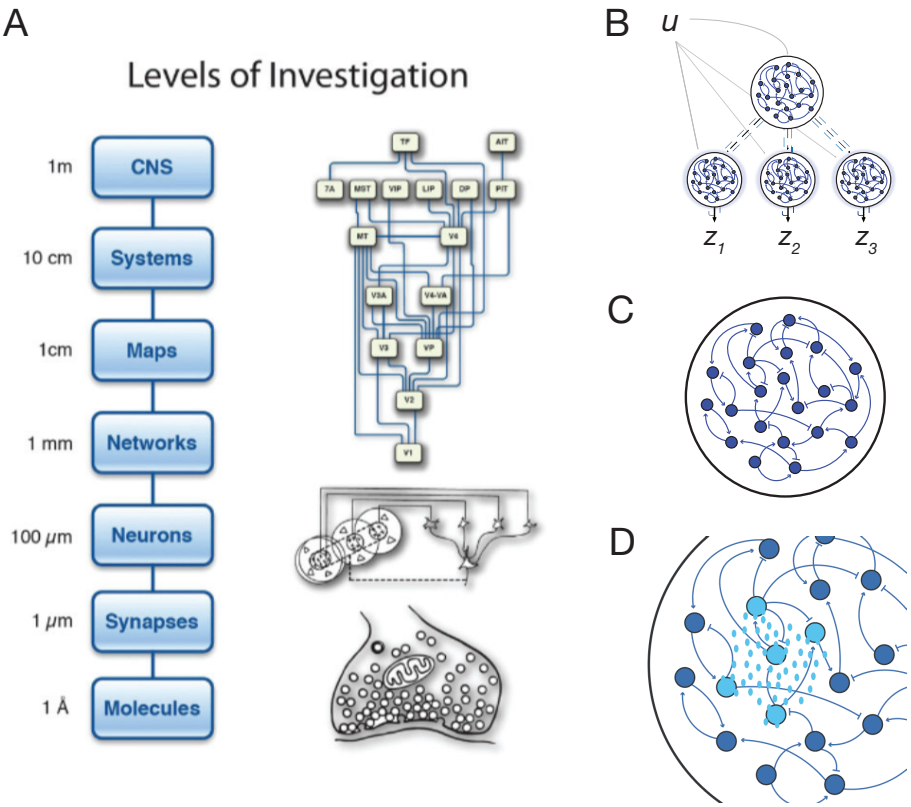


Figure 1.1. Levels of investigation in the brain. (A) Understanding how learning and memory generate flexible behaviors from the brain requires investigation across levels. This dissertation focuses on sub-levels of the network level, drawing on principles from systems and maps from above and synapses and molecules from below. Reprinted from [Sej20]. (B) Chapter 2 considers system-level coordination of subregions across the prefrontal cortex. (C) Chapter 3 considers how a single neural circuit encodes a strategy and its subsequent evolution. (D) Chapter 4 considers how a class of modulatory molecules affects synapses in a way that enables computational flexibility in a neural network.

study the development and evolution of strategies on a series of working memory tasks in which humans and monkeys showed major discrepencies. This study leads to another question: how can multiple strategies be simultaneously stored in a single circuit? Chapter 4 moves still deeper to this level of investigation — molecules acting within circuits (Figure 1.1D), — studying how a vital class of chemical compounds called neuromodulators allow robust and powerful parallelization of a network, enabling storage of multiple behavioral repertoires in a single circuit.

Importantly, these studies aim not only to understand how the brain achieves these properties, but also how they can help inform new treatments for patients. The study of the

structure of subregions within the prefrontal cortex in Chapter 2 includes lesion studies with the model, in which particular areas of the neural network are damaged and resultant behavior is assessed using a widely implemented clinical assessment for prefrontal damage. This experiment provides insight into the underlying structure and deficits that occur in diseases that affect subregions of the prefrontal cortex. The study on strategy evolution in neural networks in Chapter 3 strongly motivates use of our model to study learning impairments and disabilities in future work. As described in more detail below, neuromodulators are central to nearly every neuropsychiatric disease. The study of neuromodulators and learning in Chapter 4 suggests a new explanation for idiosyncratic (and sometimes problematic) drug response profiles of patients and why different modes of treatment (e.g. deep brain stimulation vs pharmacologics) may sometimes act so differently in a single patient.

1.1 Memory and learning and their diseases

1.1.1 Background

Memory can be thought of as information that is retained in some form. In the context of the brain, the term "memory" often causes people to think of declarative memory ("I remember what I ate for breakfast yesterday") or sometimes also procedural memory ("I remember how to ride a bike"), but in fact the basis of all behavior is rooted in memory. All thoughts and actions derive from various forms and manipulations of memory stored in the brain.

Memory in the brain can be exhibited by anything that has a lasting effect on neural computation, and as such, comes in many different forms. The most straightforward is alteration of synaptic strengths between neurons in a network, for example as shown by Bi and Poo's seminal paper [BP98] on spike-timing dependent plasticity. Other forms of memory also play important roles in the brain. Experience-dependent changes in gene expression and neural activity both can alter neural computation. The latter for example is critical for working memory, using exposure to initial stimuli to condition subsequent neural activity used to generate an appropriate

response some time later. Higher order memory mechanisms utilize these components to encode, store, organize, and retrieve neural activity sequences, for example during episodic memory. Many aspects of how and which components combine to generate these higher order memory mechanisms remain unknown.

Learning refers to the process of altering memory stored in the brain to better execute a desired behavior, for example becoming a better swimmer. Similar to memory, learning is multifaceted. Learning can occur by changing synaptic weights between neurons, changing the spatiotemporal pattern of expression of modulatory molecules that transiently affect network parameters (e.g. synaptic weights, activation functions), or changing the dynamical activity pattern of the network, in what has been called meta-learning [Wan+18].

Both memory and the processes of learning can be disrupted or disordered in various ways. Problems can occur with the formation of the structures that hold memory (malformation and neurodevelopmental disorders), the processes of encoding new memory (learning impairments), long-term structural integrity (traumatic lesions, strokes, tumors, atrophy), regulatory processes (various psychiatric disorders), and retention and recall (neurodegenerative diseases). Natural occurrences of these disruptions have provided great insight into various memory and learning processes, and provide persistent and urgent motivation to understand the underlying mechanisms.

In this dissertation, I pursue three lines of investigation, targeting three aspects of memory, learning, and their diseases. The work described in Chapter 2 pertains to problems of information flow between regions in the brain which can be caused for example by traumatic lesions, strokes, or tumors. Disorders of this type can occur due to a problem with a component circuit in the system or due to a problem with transmission between circuits in the system. Chapter 3 focuses on how synaptic memory in a neural circuit can reflect a particular problem-solving strategy, and how the modification of this memory through learning leads to strategy evolution. Problems with these processes are tightly linked to learning impairments and disabilities. Finally Chapter 4 concentrates on how neuromodulatory molecules modify parameters of network transmission to

enable storage of overlapping synaptic memory regimes within a circuit. Nearly every psychiatric illness is linked to neuromodulator abnormalities, and likewise nearly every treatment is aimed at modifying neuromodulatory signalling.

1.1.2 Disorders of neural circuit communication

The brain can grossly be mapped into functional regions. For example in the human cortex occiptal regions are critical for vision, some temporal areas are critical for hearing (auditory cortex), others for declarative memory (medial temporal lobe), others for language (e.g. Wernicke's area), parietal areas are critical for sensation (sensory cortex), and some frontal areas are critical for movement (motor and premotor regions), others for planning and decisions (prefrontal regions). These large subdivisions can often be further subdivided with some functional associations, for example into subregions of the prefrontal cortex (PFC): dorsolateral (dlPFC), ventrolateral, ventromedial (vmPFC), and orbital frontal cortex [Mac+15; Fus01; Wil+10]. Although these divisions and their functional associations are approximately true with supporting evidence from clinical cases and experimental studies, they are estimations and most often, not precise. This imprecision is not simply a shortcoming of measurement or knowledge, but rather reflective of the interconnectedness of all the component parts that make up the brain.

Some disorders can be traced to specific, localized problems with regions or the communication between regions that make up a larger circuit system. These disorders often take the form of focal lesions. The most straightforward examples are traumatic brain injuries, for example in the famous case of Phineas Gage who suffered a projectile penetration from an exploded railway rod that damaged several parts of his prefrontal cortex but left the rest of his brain intact [Mac+15; Har48]. Another example is patient H.M. [SM57] whose memory deficit symptoms following a bilateral medial temporal-lobe resection spurred and continue to spur research into the declarative memory properties of the hippocampus and medial temporal lobe. Focal lesions of this type have been incredibly informative for mapping approximate functionality of brain

regions and their communication [Mac+15].

Lesions causing inter-regional disruption also take other forms. Alzheimer's disease is a progressive neurodegenerative disease that usually begins as a declarative memory deficit that then progresses in severity and scope to eventually cause broader general dementia. This can be thought of as a lesion (a diffuse rather than traumatic, focal lesion) that likely originates in the medial temporal lobe (MTL) and then progressively expands, disrupting communication first through MTL then more broadly. Parkinson's disease, another neurodegenerative disease, with its hallmarks of rigidity, tremor, bradykinesia, and postural instability, originates from a lesion — progressive loss of neurons — of the substantia nigra, a region of the basal ganglia that is a major source of dopamine production. Normal information flow that is dependent on these dopaminergic neurons is disrupted for example in the direct and indirect pathways of the basal ganglia [Wic19].

In Chapter 2, I consider inter-regional disorders of the PFC. Using an artificial neural network system to model the PFC subregions' coordinated activity, I systematically introduce lesions of various types to the model, simulating neural ablations (e.g. traumatic injury) and neurodegeneration (weakening and loss of synaptic connections). I then use these lesioned models to compare resultant behavior to that of patients with prefrontal lesions. I employ a widely used clinical assessment of prefrontal function to reveal parallels in post-lesion behavior between the model and patients, suggesting the neural architecture underlying prefrontal function and how it becomes disrupted in various disorders.

1.1.3 Disorders of learning

Learning disabilities occur in varying degrees, from mild impairment to very severe developmental disorders. They may derive in part from structural abnormalities as in the case of cerebral malformation diseases, and in part from problems with the molecular machinery of learning processes [Rit03]. In many cases the etiology is unknown or unclear. Treatments attempt to ameliorate causative factors when known (e.g. surgical procedures for some neural

malformation disorders) or support patients (e.g. social and educational assistance), but for many learning disabilities no treatments exist, in part due to a lack of understanding of the pathophysiology. Much of current research is aimed at gaining basic understanding of why and how learning disabilities occur at the molecular, cellular, and circuit levels and developing pipelines to discover novel targeted therapeutics (e.g. [Tru+20][1]).

Chapter 3 focuses on the learning processes that occur in a neural network as it tunes to solve a task. The investigation follows how learning leads to formation of a strategy reflected in network activity dynamics that can then evolve through further learning to attain higher proficiency. By revealing some counterintuitive findings about learning heuristics and learning progression, this work lays a foundation for future investigation of how impairments of synaptic plasticity and other learning processes influence how networks store and evolve strategies.

1.1.4 Disorders of neuromodulation

Neuromodulation underlies different modes of behavior driven by different underlying cognitive states, including arousal, hunger, and moods. The behavior of a bear encountering a mysterious mushroom can vary greatly depending on its underlying state of hunger; a starved bear may consume without hesitation while a well-fed bear is more likely to forgo the risk of poisoning (Figure 1.2A). Almost every psychiatric and several neurological diseases have been linked to neuromodulator abnormalities. Clinical findings provide supportive evidence of this link with nearly every pharmacologic treatment for psychiatric illness aimed at targeting a neuromodulator class either directly (e.g. SSRIs) or indirectly [Ash+17]. The four main neuromodulators in the brain are dopamine, serotonin, acetylcholine, and norepinephrine and are produced in subcortical areas and the brainstem (Figure 1.2B), though many other neuromodulators exist including types of neuropeptides [Mer+11].

Dopamine (Figure 1.2B, blue) is a tyrosine-derived neuromodulator produced in the

[1] In parallel to my dissertation work, I coauthored this paper that established an organoid-based drug candidate identification pipeline and applied it to discover new treatments for Rett syndrome.

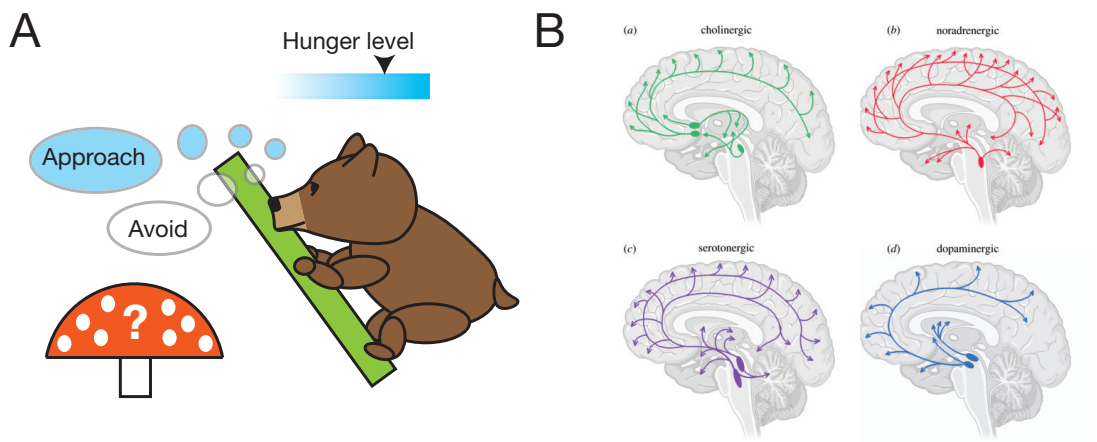


Figure 1.2. Neuromodulatory molecules control cognitive states. (A) The behavior of an organism — here, a bear — depends on the underlying cognitive state. A bear that is in dire need of food is more likely to take the risk of eating a unknown food (mushroom) than a well-fed bear who would avoid the risk. (B) Four major neuromodulators are acetylcholine (green), norepinephrine (red), serotonin (purple), and dopamine (blue). Each is produced in small regions of the subcortex or brainstem and are widely distributed throughout the brain via short and long-range projections. Reprinted from [OWS20].

substantia nigra pars compacta of the basal ganglia and the ventral tegmental area in the midbrain. Dopamine plays a central role in reward information and movement control. Loss of dopaminergic neurons in the substantia nigra pars compacta are central to the pathology of Parkinson's disease and other forms of Parkinsonism [Wic19]. Abnormally and inappropriately elevated dopamine levels are associated with symptoms of psychotic illnesses including schizophrenia [AK17]. This spectrum of dopaminergic levels and their related diseases has been firmly established and supported by evidence from pharmacologic treatments targeting dopamine. Pharmacologics for Parkinson's disease aimed at increasing dopamine levels can result in psychosis, hallucinations, and several other related symptoms. On the other hand, antipsychotic therapeutics aimed at lowering dopamine levels can cause extrapyramidal symptoms including Parkinsonism.

Serotonin (Figure 1.2B, purple) is derived from tryptophan, and is produced in the raphe nuclei in the brainstem. Serotonin is tightly linked to mood regulation and associated disorders, with selective serotonin reuptake inhibitors (SSRIs) being first-line treatment for depression

and anxiety disorders [Fur+19; Tor+19]. Norepinephine (Figure 1.2B, red), like dopamine is tyrosine-derived, and is produced in the locus coeruleus in the brainstem. In addition to its several functions peripherally including in blood pressure regulation, norepinephrine acts centrally to regulate arousal and mood, and serotonin and norepinephrine reuptake inhibitors (SNRIs) are used as close alternatives to SSRIs in treatments of depressive and anxiety disorders [Del+10].

Acetylcholine (Figure 1.2B, green) is synthesized from acetyl-CoA (a derivative of glucose) and choline, and is produced centrally in the nucleus basalis of the basal forebrain and in the brainstem. In addition to its critical role in neuromuscular junctions, it is strongly associated with memory processes in the brain. Not surprisingly, abnormal acetylcholine levels have been linked to Alzheimer's disease, and cholinesterase inhibitors have been developed to treat the memory-related symptoms. Acetylcholine also plays an important role in the direct and indirect pathways of the basal ganglia that are thought to be disrupted in Parkinson's disease. Specifically, the loss of dopaminergic neurons is thought to create an imbalance in dopamine and acetylcholine levels in the striatum, resulting in motor abnormalities [Tat+14].

Table 1.1. Abnormalities of the major neuromodulators associated with neurological and psychiatric diseases [AK17; HP14; YSY14; Tor+19; Tat+14].

	Dopamine	Serotonin	Norepinephrine	Acetylcholine
Parkinson's	×			×
OCD	×	×		
PTSD	×	×	×	
Schizophrenia	×		×	×
ADHD	×		×	×
Depression		×	×	×
Anxiety		×	×	
Alzheimer's				×

Table 1.1 shows a sample of neurological and psychiatric diseases and their links to specific neuromodulator abnormalities. Many of the drugs for these diseases target neuromodulators and have mechanisms of action and side effects that are not well understood. An example we focus on in Chapter 4 is why there is such large dose-response variability among patients, some

of whom exhibit very high sensitivity and others of whom fail to respond at the highest doses of the same drug. We reveal that the answer may be related to another puzzling observation in psychiatric therapeutics: why do some patients who fail pharmacologic treatment respond so well to other modes of treatment like deep brain stimulation?

1.2 Artificial neural network as a model system

1.2.1 Background

The human brain is a massive parallel distributed processing (PDP) device characterized by $\sim 10^{14}$ connections (synapses) between $\sim 10^{11}$ units (neurons). To begin to understand how such a computing device operates, it is critical to understand general principles of parallel distributed processing. Artificial neural networks (ANNs) are models of such PDP devices, built out of connected (via synapses) simple computational units (artificial neurons). As such, they allow unprecedented access to study how large-scale parallel distributed processing devices with neuron-like components work and what they are capable of [RMP86]. In their short history since the McCulloch-Pitts neuron in 1943 and Frank Rosenblatt's perceptron in 1958, myriad surprising emergent phenomena have been reported when ANNs are subjected to various tasks and manipulated in different ways, some of which closely mimic features of the brain.

Many different types of ANNs have been developed. ANNs can be grouped into two large categories based on structure: feedforward architectures in which information flows from start to end of the network without being passed backward, and recurrent neural networks (RNNs) which connect back onto themselves, passing information backward as well as forward through the network. Feedforward networks include traditional convolutional neural networks (CNNs) (e.g. AlexNet, ResNet, U-Net), traditional autoencoders, and transformers; recurrent architectures include sequence-to-sequence networks and recurrent autoencoders. Additionally, networks can be grouped by their method of simulation: rate networks simulate neural firing rates of each neuron, whereas spiking networks explicitly simulate individual action potentials in

neurons. Examples can be found from all these categories that have proven extremely useful in neuroscience research and technological applications [Mio+18; Alz+21; Sch+22]. The work in this dissertation focuses on brain regions and functions most suitably studied with continuous rate recurrent neural networks.

ANNs are made up of very simple units, called neurons. Neurons have weighted connections to other neurons and operate by taking input from upstream connected neurons and outputting a transformed version to downstream connected neurons. Each neuron is governed by a simple equation. For example in a typical rate network, the governing equation for neurons in the network is:

$$\tau \frac{dx}{dt} = -x + Wr + W_{in}u + \varepsilon \tag{1.1}$$

where τ represents the synaptic decay time constants for the neurons, x represents the current variables of the neurons, W is the matrix of connection weights between neurons, r is the output firing rates of the neurons, W_{in} are weights associated with external input u, and ε is added noise. The output firing rate for the neuron often involves a nonlinearity operating on the current variable called the activation function of the neuron, which mimics the relationship between the state of the neural current and consequent firing rate. An example is the standard logistic sigmoid function:

$$r = \frac{1}{1 + e^{-x}} \tag{1.2}$$

The weighted connections between the neurons in network can then be tuned to produce desired output when given input, guided by an objective function (also called a loss function, $\mathscr{L}$) that dictates how the connection weights should be tuned. A common example of an objective function is mean squared error (Equation 1.9) which gives a measure of how far off the network's output is from a desired target. To find out how to tune connection weights, partial derivatives of

the objective function with respect to each weight (θ_i) in the network can be calculated:

$$\frac{\partial \mathscr{L}}{\partial \theta_i} \tag{1.3}$$

which indicates how the loss changes with that weight. Following this gradient, each weight can be tuned in the direction that decreases the loss, often scaled by a scalar, η, called the learning rate. This process, called backpropagation, is then repeated iteratively as more examples of inputs, outputs, and desired target outputs are experienced. The objective function defines the optimization problem that backpropagation attempts to solve through iterative tuning of connection weights with training data.

Somewhat surprisingly (or perhaps not, given the brain as empirical evidence), these networks of artificial neurons have proven to be very powerful computational devices that underlie many major achievements (e.g. AlphaFold [Jum+21]), some even superhuman (in a narrow sense) like the game playing neural network MuZero [Sch+20]. RNNs in particular have been proved to be Turing-complete [SS92], and under certain conditions even super-Turing, capable of solving problems uncomputable by a Turing machine (e.g. the halting problem)[Sie95; CS14]. This computational power in theory and in practice is why ANNs are so popular and useful in machine learning.

As with any model system, there are caveats and limitations of the model in simulating animal and human neural processes. ANN models are vastly simplified approximations of biological neural networks, typically lacking features including spatial constraints (both intracellular and intercellular), metabolic properties, interacting cell types (e.g. glia), action potentials (for rate networks), and detailed molecular interactions (probabilistic neurotransmitter release and binding, individual ions and their channels, etc.). For many simple tasks, objective function construction is straightforward (e.g. labeled picture classification), but extrapolation to plausible objective functions governing human and animal life is less straightforward. Writing down any objective function governing animal learning and behavior would require several unknown

components, and it is unclear how such an objective function would be stored, if at all, and used to calculate gradients in order to optimize parameters of the system [Ric+19]. Furthermore, the learning rules used in animal brains remain unclear.

Nonetheless, ANNs have also proved to be very useful models for understanding principles of animal and human brain function. There are now innumerable examples of this [Man+13; Rus+18; KS20; Bak+21; Nay+18; Tru+20], from layers of a deep CNN predicting activity in the visual stream [Yam+14] to RNNs tightly mimicking experimental animal behaviors and neural activity properties and revealing new learning principles [Wan+18].

1.2.2 Circuit models

Recurrent neural networks

RNNs are defined by unconstrained connections of neurons within a neural population (Figure 1.3A). Precisely, this refers to W in Equation 1.1 being an unconstrained $N \times N$ matrix of real numbers ($W \in \mathbb{R}^{N \times N}$), where N is the number of neurons in the network. Recurrent connectivity is inherently suited for temporally-dependent computation since neural state information can be preserved and manipulated through neural connections that form loops. Information in this way can be retained from history and new information integrated with the old. For this reason, RNNs are particularly useful models for studying principles like working memory and reinforcement learning which rely on temporal dependencies.

Algorithmically, a major difference from feedforward ANNs is that training is done via backpropagation through time. The computation done by an RNN can be reframed as an iterative process that "unrolls" through time (Figure 1.3B). This "unrolling" of the network in time is useful for computing the error-based partial derivatives necessary for weight optimization ($\frac{\partial \mathscr{L}}{\partial \theta_i}$ from Equation 1.3 depends on $\frac{\partial x_t}{\partial \theta_i}$ which depends on $\frac{\partial x_t}{\partial x_{t-1}}, \frac{\partial x_{t-1}}{\partial x_{t-2}}, \frac{\partial x_{t-2}}{\partial x_{t-3}}$, etc.). Because the temporal duration of a task may be many steps, RNNs historically posed challenging for optimization since derivatives had to be backpropagated through several time steps. Similar to very deep feedforward networks this could lead to the "vanishing or exploding gradients"

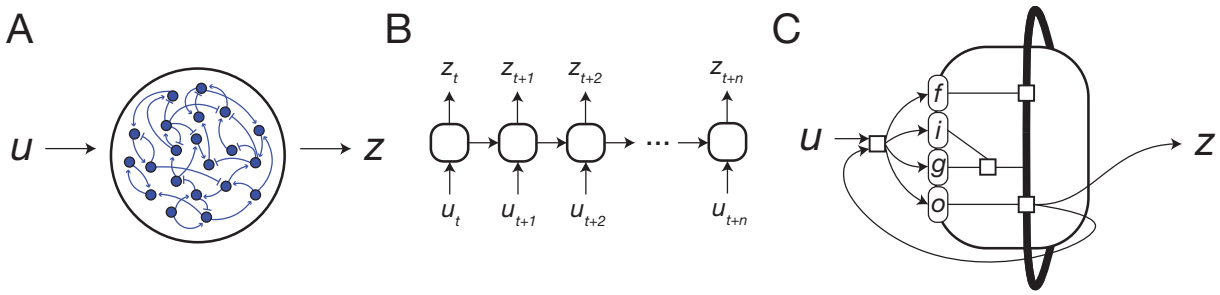


Figure 1.3. RNN circuit models. (A) Standard RNN takes input, u, and produces output, z. Neurons in the network can form weighted connections to any other neuron in the network. (B) An RNN can be "unrolled in time" to depict temporal dependencies of its computations. (C) Long short-term memory cell is a specialized RNN that contains a set of intrinsic gates: a forget gate (f), an input gate (i), and an output gate (o). Input, u, together with output from the prior timestep, z, passes through input weights (g) and conditions the gates to update the state of the network and its output, z. The thick black line represents the state of the network units and boxes represent linear and nonlinear operators.

problem, in which a gradient calculated on the output error when propagated across many time steps (i.e. the partial derivatives calculated across neural units across time steps going back to the beginning of the task) could become very small or very large, hindering training. These challenges have effectively been handled by clever weight initialization methods, improved optimizer algorithms, and altered architectures like long short-term memory networks [Sut+13; KB15; GB10; He+15; HS97].

RNNs are widely used in modeling regions of the brain [Man+13; KS20; Rus+18; Wan+18], particularly working memory-based computations in the prefrontal cortex. Mounting realizations that recurrent connections, even if sparse, play important computational roles throughout the brain have led to incorporation of RNN modelling for regions traditionally thought of as feedforward brain regions like the visual stream [Nay+18]. Each study in this dissertation focuses on recurrently connected brain regions and thus utilizes RNN circuit models.

Long short-term memory networks

Different variations of the RNN have been developed for ease of use and robustness of computation. Most draw on the insight that a limited set of intrinsic gates within the RNN lead

to faster and better learning, especially for tasks with complex temporal dependencies. Among these the most used are long short-term memory cells (LSTMs; Figure 1.3C) and a variation called gated recurrent units (GRUs). Here we introduce the LSTM which is used in Chapters 2 and 3.

The LSTM [HS97] is a recurrent network that includes three intrinsic gates: an input gate, a forget gate, and an output gate. Each gate influences how the units of the RNN encode information. The governing equations for these gates are:

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{1.4}$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{1.5}$$

$$o_t = \sigma(W_{ox}x_t + W_{ho}h_{t-1} + b_o) \tag{1.6}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{1.7}$$

$$h_t = o_t \circ tanh(c_t) \tag{1.8}$$

where f_t, i_t,and o_t are the forget, input, and output gates at time t respectively, σ is the sigmoid activation function, W_{ij} denotes the weights from component i to component j, x_t is the external input at time t, h_t is the ouput of the LSTM at time t, c_t is the state of the LSTM cell at time t, b_f, b_i, and b_o are the biases of the forget, input and output gates respectively, b_c is the bias of the cell states, and $\circ$ denotes the Hadamard product.

As in all ANNs, input into the network passes through input weights (W_{xc} in Equation 1.7 and g in Figure 1.3C). The input gate (i) further modifies the inputs before incorporation into the state of the RNN units (c). The forget gate (f) determines how much of the previous state of the RNN units is passed on to the next state. Finally the output gate (o) modifies the final form of information that is propagated out of the RNN (h).

LSTMs have been used to model areas of the brain involving working memory and recurrent connection loops like those between the PFC, basal ganglia, and thalamus [Wan+18].

The ANNs used in Chapter 2 and 3 to model properties of PFC circuits trained by reinforcement learning utilize LSTMs.

1.2.3 Learning methods

In the training of ANNs, different learning methods can be used. These fall broadly into three categories: supervised, unsupervised, and reinforcement learning. Supervised learning refers to training an ANN with a teacher signal. This can take the form of true labels in a categorization task or a desired output signal in a continuous output value task. A loss (objective) function ($\mathscr{L}$) is then used to calculate an error. For example, the following loss function calculates the mean squared error for a desired output signal, z:

$$\mathscr{L} = \frac{1}{T}\sum_{t=1}^{T}(z_t - o_t)^2 \tag{1.9}$$

where T is the number of timesteps in a trial, z_t is the desired signal and o_t is the ANN output at time t. This error is then used to calculate gradients (partial derivatives of the error with respect to each individual weight in the ANN) for backpropagation weight optimization. Supervised learning is use in Chapter 4.

Unsupervised learning refers to learning that occurs in the absence of a teacher signal, for example clustering. In this case the objective function of the network dictates desired properties of the output without an external teacher signal. An example of unsupervised learning occurs in an autoencoder, which is an ANN that learns to reconstruct whatever input it is given, often performing compression with middle layers being lower dimensional than input and output layers.

Reinforcement learning

Reinforcement learning is a prominent learning mechanism used by humans and animals. Also referred to as "trial and error learning," it occurs through a sequence of experiences in which the agent (e.g. human, animal, or ANN) takes an action and receives feedback from the

environment that carries a positive or negative valence — the reward. Through repeated actions the agent learns associations of environmental states, actions, and rewards. Biologically, these processes likely involve cortical areas, which convey information and encode, organize, and refine memory; the basal ganglia, where reward information is processed and elaborated and motor associations are stored; and dopamine signaling, which conveys a reward prediction error signal [Sej+14; MDS96; SDM97]. Mathematically, the class of problems that reinforcement learning attempts to solve can be posed as a Markov decision processes (MDPs) [SB18]: at each state the agent makes a decision and with some probability gets some reward and ends up in some other state. Using the reward feedback, the agent learns which actions taken in which states lead to positive and negative outcomes. As I show in Chapters 2 and 3 and others have shown in other works and experiments, reinforcement learning is a simple and incredibly powerful mechanism for encoding function approximators in a neural network to solve MDPs.

The core of why reinforcement learning works is captured in a single equation: the Bellman equation, developed by Richard Bellman in the 1950s [SB18].

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma v_\pi(s')] \tag{1.10}$$

where $v_\pi(s)$ is the value of state s under policy π, $\pi(a|s)$ is the probability of taking action a given state s under the policy, s' is the next state, r is the reward, and γ is a temporal discount factor.

This equation reveals a recurrence, namely that the value of the current state s can be written as an average over all temporally adjacent states s'. Importantly, because of this recurrence, the value of state s includes complete information about all possible futures extending from that state (since it is the average of all possible next state's values, which each are the average of all possible next next states, etc.). The goal of reinforcement learning is to find the most accurate value map, or framed another way, the optimal policy π given the value map. This solution is

captured by the Bellman optimality equation:

$$v_*(s) = \max_a \sum_{s',r} p(s',r|s,a)[r + \gamma v_*(s')] \tag{1.11}$$

where $v_*(s)$ is optimal state value function. Reinforcement algorithms aim to closely approximate this solution, for example with a function approximator in the form of a neural network.

The algorithmic implementation of reinforcement learning can take many forms, which depend on the representation of the objective function. Reinforcement learning can be thought of as acting on a representation of the value of states and actions given the state of an environment. A popular implementation of this is Q learning, in which an agent holds a representation of the value of particular actions given the state of the environment (q; the Bellman equations can also be written for q rather than v). Another widely used implementation involves the agent simply holding a representation of the policy π, i.e., what action to take in what state, alleviating the requirement to accurately capture value associations of actions.

Reinforcement learning in humans and animals often exhibits a form of temporal discounting, in which values (i.e. rewards) associated with states more distant temporally (e.g. further into projected future) have less influence than more immediate rewards [BS14]. Such temporal discounting underlies effective planning – actions taken several steps ahead of a final desired outcome enable achievement of that outcome (like diligently working in year 1 to finally earn a diploma in year 4). Temporal discounting is algorithmically implemented as temporal difference learning, introducing a discounting factor, γ, which represents the exponentially decreasing influence of a reward on temporally adjacent states.

Another commonly used method within reinforcement learning is utilization of a reference or baseline value, known in its algorithmic implementation as "actor-critic" [SB18]. Reward feedback from the environment is compared to a stored representation of the value of that state, held independent of the policy representation. This comparison — the "critic" — is used to shape the policy — the "actor" — and takes the form of a temporal difference error. The use of this

baseline value estimation can accelerate learning by constraining the variance of the updates. The actor-critic comparison of reward to expected reward (value estimation) is strongly analogous to dopaminergic reward prediction error signalling in the basal ganglia [JNR02; Sej+14; BNB09].

In Chapters 2 and 3 I use reinforcement learning to train neural network models. I use an actor-critic policy method with temporal difference learning outlined in Mnih et al. 2016 [Mni+16]. The objective function ($\mathscr{L}$) I use is defined by three components: a policy term ($\mathscr{L}_\pi$) and value term ($\mathscr{L}_v$) as discussed above, and an entropy of the policy term ($\mathscr{L}_H$) which simply encourages exploration and prevents premature convergence to a set policy before experiencing the full environment. The gradient of the objective function I use is given by:

$$\nabla\mathscr{L} = \nabla\mathscr{L}_\pi + \nabla\mathscr{L}_v + \nabla\mathscr{L}_H \tag{1.12}$$

$$= \frac{\partial log\pi(a_t|s_t;\theta)}{\partial\theta}\delta_t(s_t;\theta) + \beta_v\delta_t(s_t;\theta)\frac{\partial V}{\partial\theta} + \beta_H\frac{\partial H(\pi(a_t|s_t;\theta))}{\partial\theta} \tag{1.13}$$

where π is the policy, a_t is the action taken at time t, s_t is the state at time t, θ is the network parameters, β_v, β_H are scalar constants, V is the value output of the network, and H is entropy. δ_t is the "advantage estimate" of the critic, which represents the temporal difference error as described above:

$$\delta_t(s_t;\theta) = \left(r_t + \sum_{i=1}^{k-1}\gamma^i r_{t+i} + \gamma^k V(s_{t+k};\theta)\right) - V(s_t;\theta) \tag{1.14}$$

where r_t is the reward at time t, γ is the discount factor, and k is the number of steps until the next end state.

1.3 Levels of investigation

This dissertation takes the approach of studying memory and learning across levels of investigation (Figure 1.1). Chapter 2 considers a system of circuits — how arrangement and

coordination of subregions of the prefrontal cortex support flexible behavior. Chapter 3 zooms in to an individual circuit — how a circuit can adopt a strategy and how that strategy can evolve as a function of network capacity and experience. Chapter 4 zooms in to the molecular level — how a class of molecules called neuromodulators can enable a circuit to flexibly encode multiple behaviors in parallel.

The integration of these three levels of the neural circuit lead to a broader understanding of how parallel distributed computing can work at multiple levels in the brain. Molecules establish unique and accessible parallel modes within a circuit, each mode within a circuit itself can evolve, and combining multiple circuits into a circuit system can coalesce into an integrated flexible higher-order memory device. Along with providing an appreciation for the seamless layering of these levels to generate complex function, study across these levels reveals a better understanding of consequences of disruptions across the whole system such as those that occur in disease. Effective treatments will be those designed taking into account effects that occur at each of these levels.

Bibliography

[Alz+21] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8: 2021, 53.

[Ash+17] Abhishekh H Ashok, Tiago R Marques, Sameer Jauhar, Matthew M Nour, Guy M Goodwin, Allan H Young, and Oliver D Howes. The dopamine hypothesis of bipolar affective disorder: the state of the art and implications for treatment. *Molecular Psychiatry*, 22: 2017, 666–679.

[AK17] Michael C. Avery and Jeffrey L. Krichmar. Neuromodulatory Systems and Their Interactions: A Review of Models, Theories, and Experiments. *Frontiers in Neural Circuits*, 11: Dec. 2017, 108.

[Bak+21] Shahab Bakhtiari, Patrick Mineault, Timothy Lillicrap, Christopher Pack, and Blake Richards. The functional specialization of visual cortex emerges from training parallel pathways with self-supervised predictive learning. *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*. Ed. by M. Ranzato, A. Beygelzimer, K. Nguyen, P.S. Liang, J.W. Vaughan, and Y. Dauphin. 2021.

[BS14] Maria A. Bermudez and Wolfram Schultz. Timing in reward and decision processes. *Philosophical Transactions of the Royal Society B*, 369: 2014, 20120468.

[BP98] Guo-qiang Bi and Mu-ming Poo. Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type. *Journal of Neuroscience*, 18(24): 1998, 10464–10472.

[BNB09] Matthew M. Botvinick, Yael Niv, and Andrew G. Barto. Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *Cognition*, 113: 2009, 262–280.

[CS14] Jeremie Cabessa and Hava T. Siegelmann. The Super-Turing Computational Power of Plastic Recurrent Neural Networks. *International Journal of Neural Systems*, 24(8): 2014, 1450029.

[Del+10] Bernardo Dell'Osso, Massimiliano Buoli, David S. Baldwin, and A. Carlo Altamura. Serotonin norepinephrine reuptake inhibitors (SNRIs) in anxiety disorders: a comprehensive review of their clinical efficacy. *Human Psychopharmacology Clinical & Experimental*, 25: 2010, 17–29.

[Fur+19] Toshi A Furukawa, Andrea Cipriani, Philip J Cowen, Stefan Leucht, Matthias Egger, and Georgia Salanti. Optimal dose of selective serotonin reuptake inhibitors, venlafaxine, and mirtazapine in major depression: a systematic review and dose-response meta-analysis. *Lancet Psychiatry*, 6: July 2019, 601–609.

[GB10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, May 2010, 249–256.

[Har48] John M. Harlow. Passage of an iron rod through the head. *The Boston Medical and Surgical Journal*, 39: 1848, 389–393.

[He+15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *IEEE International Conference on Computer Vision (ICCV 2015)*, 1502: Feb. 2015.

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, 770–778.

[HP14] Michael J Higley and Marina R Picciotto. Neuromodulation by acetylcholine: examples from schizophrenia and depression. *Current Opinion in Neurobiology*, 29: 2014, 88–95.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8): 1997, 1735–1780.

[Hym13] Steven E. Hyman. Psychiatric Drug Development: Diagnosing a Crisis. *Cerebrum*, 5: 2013.

[JNR02] Daphna Joel, Yael Niv, and Eytan Ruppin. Actor–critic models of the basal ganglia: new anatomical and computational perspectives. *Neural Networks*, 15(4): 2002, 535–547.

[Jum+21] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596: 2021, 583–589.

[KS20] Robert Kim and Terrence J. Sejnowski. Strong inhibitory signaling underlies stable temporal dynamics and working memory in spiking neural networks. *Nature Neuroscience*, 24: 2020.

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015.

[Mac+15] Sarah MacPherson, Sergio Sala, Simon Cox, Alessandra Girardi, and Matthew Iveson. *Handbook of Frontal Lobe Assessment*. Oxford, UK: Oxford University Press, 2015. ISBN: 9780199669523.

[Man+13] Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503: 2013, 78–84.

[Mer+11] Adalberto Merighi, Chiara Salio, Francesco Ferrini, and Laura Lossi. Neuromodulatory function of neuropeptides in the normal CNS. *Journal of Chemical Neuroanatomy*, 42(4): 2011, 276–287.

[Mio+18] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T. Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in Bioinformatics*, 19(6): 2018, 1236–1246.

[Mni+16] Volodymyr Mnih, Adria P Badia, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P Lillicrap, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. Ed. by JMLR. Vol. 48. New York, 2016, 1928–1937.

[MDS96] PR Montague, P Dayan, and Terrence J. Sejnowski. A framework for mesencephalic dopamine systems based on predictive Hebbian learning. *Journal of Neuroscience*, 16(5): 1996, 1936–1947.

[Nay+18] Aran Nayebi, Daniel Bear, Jonas Kubilius, Kohitij Kar, Surya Ganguli, David Sussillo, James J. DiCarlo, and Daniel L. Yamins. Task-Driven Convolutional Recurrent Models of the Visual System. *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018.

[OWS20] Claire O'Callaghan, Ishan C. Walpola, and James M. Shine. Neuromodulation of the mind-wandering brain state: the interaction between neuromodulatory tone, sharp wave-ripples and spontaneous thought. *Philosophical Transactions of the Royal Society B*, 376: 2020, 20190699.

[Ric+19] Blake A Richards, Timothy P Lillicrap, Philippe Beaudoin, Yoshua Bengio, Rafal Bogacz, Amelia Christensen, Claudia Clopath, Rui Ponte Costa, Archy de Berker, Surya Ganguli, Colleen J Gillon, Danijar Hafner, Adam Kepecs, Nikolaus Kriegeskorte, Peter Latham, Grace W Lindsay, Kenneth D Miller, Richard Naud, Christopher C Pack, Panayiota Poirazi, Pieter Roelfsema, João Sacramento, Anew Saxe, Benjamin

Scellier, Anna C Schapiro, Walter Senn, Greg Wayne, Daniel Yamins, Friedemann Zenke, Joel Zylberberg, Denis Therien, and Konrad P Kording. A deep learning framework for neuroscience. *Nature Neuroscience*, 22(11): 2019, 1761–1770.

[Rit03] C.D. Rittey. Learning difficulties: What the neurologist needs to know. *Journal of Neurology, Neurosurgery, and Psychiatry*, 74: 2003, i30–i36.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*. Ed. by Nassir Navab, Joachim Hornegger, William M. Wells III, and Alejandro F. Frangi. Vol. 9351. Lecture Notes in Computer Science. Springer, 2015, 234–241.

[RMP86] David E. Rumelhart, James L. McClelland, and PDP Research Group, eds. *Parallel Distributed Processing. Volume 1: Foundations*. Cambridge, MA: MIT Press, 1986.

[Rus+18] Abigail A. Russo, Sean R. Bittner, Sean M. Perkins, Jeffrey S. Seely, Brian M. London, Antonio H. Lara, Andrew Miri, Najja J. Marshall, Adam Kohn, Thomas M. Jessell, Laurence F. Abbott, John P. Cunningham, and Mark M. Churchland. Motor Cortex Embeds Muscle-like Commands in an Untangled Population Response. *Neuron*, 97: 2018, 953–966.

[Sch+20] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy P. Lillicrap, and David Silver. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. *Nature*, 588: 2020, 605–609.

[SDM97] Wolfram Schultz, Peter Dayan, and P. Read Montague. A Neural Substrate of Prediction and Reward. *Science*, 275(5306): 1997, 1593–1599.

[Sch+22] Catherine D. Schuman, Shruti R. Kulkarni, Maryam Parsa, J. Parker Mitchell, Prasanna Date, and Bill Kay. Opportunities for neuromorphic computing algorithms and application. *Nature Computational Science*, 2: 2022, 10–19.

[SM57] William B Scoville and Brenda Milner. Loss of recent memory after bilateral hippocampal lesions. *Journal of Neurology, Neurosurgery, and Psychiatry*, 20: 1957, 11–21.

[Sej+14] Terrence J Sejnowski, Howard Poizner, Gary Lynch, Sergei Gepshtein, and Ralph J Greenspan. Prospective Optimization. *Proceedings of the IEEE*. Vol. 102. 5. IEEE, May 2014, 799–811.

[Sej20] Terrence J. Sejnowski. The unreasonable effectiveness of deep learning in artificial intelligence. *Proceedings of the National Academy of Sciences*, 117(48): 2020, 30033–30038.

[Sie95] Hava T. Siegelmann. Computation Beyond the Turing Limit. *Science*, 268: 1995, 545–548.

[SS92] Hava T. Siegelmann and Eduardo D. Sontag. On the Computational Power of Neural Nets. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 1992, 440–449.

[Sut+13] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the Importance of Initialization and Momentum in Deep Learning. *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. Atlanta, GA, USA: JMLR.org, 2013, III–1139–III–1147.

[SB18] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: an Introduction*. Second. Cambridge, MA: MIT Press, 2018.

[Tat+14] Ada M. Tata, Lucia Velluto, Chiara D'Angelo, and Marcella Reale. Cholinergic System Dysfunction and Neurodegenerative Diseases: Cause or Effect? *CNS & Neurological Disorders - Drug Targets*, 13: 2014, 1294–1303.

[Tor+19] Sebastiano Alfio Torrisi, Gian Marco Leggio, Filippo Drago, and Salvatore Salomone. Therapeutic Challenges of Post-traumatic Stress Disorder: Focus on the Dopaminergic System. *Frontiers in Pharmacology*, 10: 2019, 404.

[Tru+20] Cleber A Trujillo, Jason W Adams, Priscilla D Negraes, Cassiano Carromeu, Leon Tejwani, Allan Acab, Ben Tsuda, Charles A Thomas, Neha Sodhi, Katherine M Fichter, Sarah Romero, Fabian Zanella, Terrence J Sejnowski, Henning Ulrich, and Alysson R Muotri. Pharmacological reversal of synaptic and network pathology in human MECP2-KO neurons and cortical organoids. *EMBO Molecular Medicine*, 13: 2020, e12523.

[Wan+18] Jane X Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Demis Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, 21: June 2018, 860–868.

[Wic19] Thomas Wichmann. Changing Views of the Pathophysiology of Parkinsonism. *Movement Disorders*, 34(8): 2019, 1130–1143.

[Wil+10] Charles R.E. Wilson, David Gaffan, Philip G.F. Browning, and Mark G. Baxter. Functional localization within the prefrontal cortex: missing the forest for the trees? *Trends in Neurosciences*, 33: 2010.

[YSY14] Ken-ichi Yamamoto, Toshikazu Shinba, and Mitsunobu Yoshii. Psychiatric symptoms of noradrenergic dysfunction: A pathophysiological view. *Psychiatry and Clinical Neurosciences*, 68: 2014, 1–20.

[Yam+14] Daniel L. K. Yamins, Ha Hong, Charles F. Cadieu, Ethan A. Solomon, Darren Seibert, and James J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111: 2014, 8619–8624.

[Yan+19] Guangyu Robert Yang, Madhura R Joglekar, H Francis Song, William T Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22: Feb. 2019, 297–306.

Chapter 2

A modeling framework for adaptive lifelong learning with transfer and savings through gating in the prefrontal cortex

In this chapter I consider how a system of circuits in the prefrontal cortex can coordinate to generate flexible, coherent behaviors. I explore how a simple arrangement of neural network architecture with a temporally dynamic component gives rise to a form of inductive transfer learning and preserves memory of prior experiences while continuing to behave flexibly in a changing environment. I apply this ANN model to study how various impairments affect the circuit system and observe interesting parallels with assessments of patients with prefrontal impairments.

The work presented here is reproduced and adapted from: Ben Tsuda, Kay M. Tye, Hava T. Siegelmann, and Terrence J. Sejnowski. A modeling framework for adaptive lifelong learning with transfer and savings through gating in the prefrontal cortex. *Proceedings of the National Academy of Sciences*, 117 (47), 29872-29882, 2020.

2.1 Abstract

The prefrontal cortex encodes and stores numerous, often disparate, schemas and flexibly switches between them. Recent research on artificial neural networks trained by reinforcement learning has made it possible to model fundamental processes underlying schema encoding and

storage. Yet how the brain is able to create new schemas while preserving and utilizing old schemas remains unclear. Here we propose a simple neural network framework that incorporates hierarchical gating to model the prefrontal cortex's ability to flexibly encode and use multiple disparate schemas. We show how gating naturally leads to transfer learning and robust memory savings. We then show how neuropsychological impairments observed in patients with prefrontal damage are mimicked by lesions of our network. Our architecture, which we call DynaMoE, provides a fundamental framework for how the prefrontal cortex may handle the abundance of schemas necessary to navigate the real world.

2.2 Introduction

Humans and animals have evolved the ability to flexibly and dynamically adapt their behavior to suit the relevant task at hand [Mon03]. During a soccer match, at one end of the pitch a player attempts to stop the ball from entering the net. A few moments later at the opposite end of the pitch, the same player now tries to put the ball precisely into the net. To an uninitiated viewer, such apparently contradictory behaviors in nearly identical settings may seem puzzling, yet the ease with which the player switches between these behaviors (keep ball away from net or put ball into net) highlights the ease with which we adapt our behavior to the ever-changing contexts (near own net or opposing team's net) we experience in the world. A bulk of evidence from observations of humans with prefrontal cortical lesions, neuroimaging studies, and animal experiments have indicated the importance of the prefrontal cortex (PFC) and connected regions in encoding, storing, and utilizing such context-dependent behavioral strategies, often referred to as mental schemas [Buc+05; Fus01; GAT19; Mac+15; MMT06]. Yet how the prefrontal and related areas are able to translate series of experiences in the world into coherent mental schemas which can then be used to navigate the world remains unknown.

Research in reinforcement learning has helped provide some insight into how the PFC may transform experiences into operational schemas [Sej+14; SYW17; Wan+18]. In rein-

forcement learning paradigms, an agent learns through trial and error, taking actions in the world and receiving feedback [SB18]. Recent work has demonstrated how recurrent neural networks (RNNs) trained by trial-by-trial reinforcement learning can result in powerful function approximators that mimic the complex behavior of animals in experimental studies [Wan+18].

Although reinforcement learning has provided invaluable insight into mechanisms the PFC may use, it remains unclear how the PFC is able to encode multiple schemas, building on each other, without interference, and persisting so they may be accessed again in the future. The majority of models capable of solving multi-strategy problems require specially curated training regimens, most often by interleaving examples of different problem types [Yan+19]. Models learn successfully due to the balanced presentation of examples in training; if the training regimen is altered—for example, problem types appear in sequence rather than interleaved, as often happens in the world—the unbalanced models fail miserably [Fre99].

Some techniques have been proposed to help models learn and remember more robustly, yet none have established how these processes may occur together in the brain. For example, continual learning techniques (e.g. [Kir+17; ZPG17]) propose selective protection of weights. Yet such techniques heavily bias networks toward internal structures that favor earlier, older experiences over newer ones and are potentially not biologically realistic [Yan+19]. Other models either require explicit storage of past episodes for constant reference [Cha+19; LR17], or an "oracle" to indicate when tasks are "new" [ACT17; Rus+16].

Experimental studies have suggested that areas within the prefrontal cortex and related regions may adopt a gating-like mechanism to control the flow of information in the brain in order to support complex behaviors involving multiple schemas [GB11; Joh+07; RGH18]. Many forms of prefrontal gating have been proposed in the literature to date, including gating of sensory information [Pos05; ON19; Wee+18], gating mechanisms to support working memory [MK08], and gating of task-relevant activity [MMT06; Joh+07; RGH18]. Building off experimental findings, computational models incorporating gating have been developed for action sequences [DCB15] and particularly for working memory [BC00; OF06; FLO01; Rou+05; FB12; Wan+18].

Furthermore, experimental and modeling work has suggested functional divisions within the PFC and neighboring areas [Mac+15; Tye18; SL02; Mil63; DCK14]. Clinical and neuroimaging observations from patients with prefrontal lesions have strongly linked dorsolateral PFC (dlPFC) to set-shifting [Mac+15; SL02; Mil63]. Additionally, some clinical and experimental findings have indicated ventromedial PFC (vmPFC) and anterior cingulate cortex (ACC) involvement in set-shifting [Mac+15; Joh+07; DCK14] while others have not [Mac+15; Mil63]. Notably, although multiple models for working memory mechanisms have been proposed, there is a lack of models designed to investigate schema encoding and shifting, linked to functions distributed across dlPFC, vmPFC, OFC, and ACC among other areas.

We hypothesized that, in addition to gating mechanisms supporting maintenance of working memory, the presence of structural gating could support manipulation and adaptation of multiple task-specific schemas in the prefrontal cortex. Such a network architecture could learn multiple schemas through reinforcement and adapt to new environments without "oracle" supervision, while remaining robust against catastrophic forgetting. To investigate this, we developed a neural network framework for the PFC that mirrors the mixture of experts (MoE) class of models [Jac+91] (Fig. 2.1*A*). MoE networks are used widely across machine learning applications [YWG12; Sha+17] and have been shown to support transfer learning in combination with reinforcement learning [Sin92a; Sin92b; DL17; GSL18] and Bayesian nonparametrics [RG01; WR95; SN96; FFW96; ACT17; KMG06].

Using our network model, we demonstrate how structural gating naturally leads to transfer learning as new scenarios are encountered and schemas are encoded. Furthermore, we show how our network adaptively learns and, due to its architecture, demonstrates robust memory savings for past experiences. We implemented lesions to our model to study how functional components may become disrupted in disease and found that the lesions recapitulated specific neuropsychological impairments observed in patients. Our framework provides a basis for how the PFC and related areas may encode, store, and access multiple schemas.

2.3 Materials and methods

2.3.1 Behavioral task

To demonstrate our framework we used the Wisconsin Card Sorting Task. In this task the subject is asked to sort cards. Each card has symbols with a shape type (triangle, star, cross, circle), a color type (red, green, yellow, blue), and a specific number of symbols (one, two, three, four). During each episode, an unsignaled operating rule is chosen: either shape, color, or number. The subject must discover the rule by trial and error and then sort a given card according to the relevant rule into one of four stacks. The first stack, Stack 1, has 1 red triangle, Stack 2 has 2 green stars, Stack 3 has 3 yellow crosses, and Stack 4 has 4 blue circles (Fig. 2.1*B*). After each attempted card sort, the subject is given feedback as to whether the sort was correct or incorrect. Once the subject has sorted a given number of cards correctly consecutively the operating rule is switched without signal and the subject must discover the new rule through trial and error. For all of our simulations, the operating rule was switched after 3 correct sorts in a row.

At the beginning of each episode a deck of cards containing all 64 possible combinations of shape, color, and number was generated. Cards were randomly drawn from this deck and presented to the subject for sorting, removing each card from the deck after presentation. If all 64 cards from the deck were used before termination of the episode, the deck was regenerated and new cards continued to be drawn in the same manner. An episode was terminated by meeting one of two termination criteria: 1) achieving the given number of correct sorts in a row (3 for our simulations) or 2) reaching the maximum episode length which we set to 200 card draws.

In our sequential scenario training simulations, a particular operating rule was kept constant for the duration of training in that period, either until a given number of sorting episodes was achieved or until performance passed a satisfactory threshold. In the next training period, a new operating rule was held constant and training was repeated in the same manner. As a demonstration, a DynaMoE network was trained in a sequential training protocol with sequential blocks of 1,250 sorting episodes (1 "sorting episode" $\approx$ 12 total WCST episodes across whole

network) of each sort rule type (Fig. 2.2*C*). Each 1,250 sorting episode block was split into two 625 sorting episode subblocks; in the first subblock, the gating network was tuned and in the second both the gating and new expert network were tuned. To evaluate the degree of transfer learning, a moving mean over every 100 sorting episodes was taken for the periods of isolated gate retuning (no expert training), and the minimum value was compared to the minimum value of the initial network before training any expert (baseline without transfer). When the shape sort rule was reintroduced, only the gating network was tuned. The 1,250 sorting episode block training protocol described above was also done with a standard RNN for comparison (*Appendix*, Fig. 2.7). In all line plots of sorts to criterion over training, a moving mean over every 10 sorting episodes was calculated and plotted for readability.

2.3.2 Reinforcement learning training

To train our networks with reinforcement learning, we used the Advantage Actor-Critic algorithm of Mnih et al. [Mni+16], where a full description of the algorithm can be found. Briefly, the objective function for our neural network consists of the gradient of a policy term, an advantage value term, and an entropy regularization term:

$$\begin{aligned}\nabla\mathscr{L} &= \nabla\mathscr{L}_\pi + \nabla\mathscr{L}_v + \nabla\mathscr{L}_H \\ &= \frac{\partial log\pi(a_t|s_t;\theta)}{\partial\theta}\delta_t(s_t;\theta) + \beta_v\delta_t(s_t;\theta)\frac{\partial V}{\partial\theta} + \beta_H\frac{\partial H(\pi(a_t|s_t;\theta))}{\partial\theta}\end{aligned}$$

where π is the policy, a_t is the action taken at time t, s_t is the state at time t, θ is the network parameters, β_v, β_H are hyperparameters for scaling the contribution of the value and entropy terms respectively, V is the value output of the network, and H is the entropy regularization term of the policy. δ_t is the advantage estimate, which represents the temporal difference error:

$$\delta_t(s_t;\theta) = R_t - V(s_t;\theta)$$

where R_t is the discounted reward:

$$R_t = \sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(s_{t+k}; \theta)$$

where k is the number of steps until the next end state. When $\gamma = 0$, $R_t = r_t$.

The advantage equation in this case is equivalent to a temporal-difference error signal enabling temporal difference reinforcement learning.

The parameters of the model were updated during training by gradient descent and back propagation through time after the completion of every 3 episodes. For all simulations we used 12 asynchronous threads for training. In our plots, a single "sorting episode" was defined as the number of total WCST episodes completed while a single thread completed 1 episode, which was roughly equal to 12 episodes for the total network. We used the AdamOptimizer with a learning rate of $1e-03$ to optimize weights. The objective function scaling hyperparameters β_v and β_H were both set to 0.05 for all our simulations.

For feedback as to whether each card sort was correct or incorrect, we gave a reward of +5 if correct and -5 if incorrect. For the WCST, a discount factor of $\gamma = 0$ was used since each card sort was an independent event, based only on the relevant operating rule rather than any prior previous action sequence.

Similar to the implementation by Wang et al. [Wan+18], the input to the networks for each step was given as vector with the current card shape, color, and number, the action taken for the previous time step, a_{t-1}, and the reward given for previous card-sort action, r_{t-1}.

2.3.3 Network architecture

Both our standard RNN and DynaMoE network architectures were composed of LSTMs as implemented by Wang et al. [Wan+18] (for details see *Appendix*). In contrast to "vanilla" RNNs, LSTMs copy their state from each time step to the next by default and utilize a combination of built-in gates to forget, input new information, and output from the states. This RNN

structure allows for robust learning and storage of functional approximators for various tasks as demonstrated by Wang et al. [Wan+18]. The LSTM states and gates are described by the following equations:

$$
\begin{aligned}
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
o_t &= \sigma(W_{ox}x_t + W_{ho}h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
h_t &= o_t \circ tanh(c_t)
\end{aligned}
$$

where f_t, i_t,and o_t are the forget, input, and output gates at time t respectively, σ is the sigmoid activation function, W_{ij} denotes the weights from component i to component j, x_t is the external input at time t, h_t is the ouput of the LSTM at time t, c_t is the state of the LSTM cell at time t, b_f, b_i, and b_o are the biases of the forget, input and output gates respectively, b_c is the bias of the cell states, and $\circ$ denotes the Hadamard product.

For all our simulations described in the paper we used a standard RNN of 105 units and a DynaMoE network with a 98 unit gating network and 19 unit experts. We chose these network sizes because they provided ample capacity to learn the WCST scenarios and shared the same number of total trainable network parameters (47,145) which enabled the direct comparisons between standard RNN and DynaMoE networks.

In DynaMoE networks, if the gating network could not solve a scenario using its current experts, a new expert was brought online. In this case, first the gating network was retuned with the current experts and an additional randomly initialized expert of the same size. If performance did not achieve the desired performance criterion, the gating network and the new expert were then trained simultaneously. The gating network LSTM learned a functional approximator mapping from inputs to experts, and the experts learned functional approximators mapping from

inputs to actions in their input domain of expertise which was determined by the gating network's mapping function.

2.3.4 DynaMoE seeded with pretrained experts transfer simulations

For our demonstration of DynaMoE networks' transfer learning property, we performed a simulation with pretrained experts. We trained one expert on only shape sorting until the expert network achieved near perfect "expert performance," defined in this simulation as an average sorts to criterion of <4 in the last 100 episodes of a single asynchronous thread (minimum sorts to criterion is 3). We repeated the same with a second expert network trained on only color sorting. We then created a DynaMoE network with these two pretrained expert networks and a third randomly initialized expert network, and trained the gating network only on the number sorting rule for 7,500 sorting episodes to ensure convergent decision behavior. "Expert performance" as defined above was not achieved during this stage of training (Fig. 2.3*B, Inset*). Network weights were then fixed and behavior and performance of the network was evaluated. To evaluate behavior of the network, 1,000 test episodes were performed in the number rule and the proportion of decisions to use each expert network (the decision rate) was measured in subsets of the number rule described in the the Results section ("shape-match-number," "color-match-number,""no-match-number;" Fig. 2.3*C*). From this parent network, we then ran 10 independent training runs in parallel in which the gating network and the randomly initialized expert network were trained simultaneously on the number sorting rule until the "expert performance" criteria was achieved. To evaluate the decision rate of the gating network for each of the 10 independent training runs, 1,000 test episodes were performed and the mean and standard deviation of the decision rates were calculated in the same subsets of the number rule (Fig. 2.3*F* and *Appendix*, Fig. 2.8).

2.3.5 Organic transfer simulations

For our demonstration of the DynaMoE network's implementation of transfer learning without any pretraining, we independently trained 10 DynaMoE networks from scratch in the following manner. We began with a randomly initialized gating network with a single randomly initialized expert network. The gating network was then trained alone on the shape sort scenario of the WCST for 1,250 sorting episodes. The gate and single expert network were then trained simultaneously until "expert performance," defined for this simulation as an average sorts to criterion of <4 in the last 100 episodes of a single asynchronous thread. A new randomly initialized expert network was then added, and the gating network was trained for 7,500 sorting episodes in the color scenario to allow full convergence of decision behavior. The gate and new expert were then trained simultaneously until "expert performance" was achieved. This was repeated finally for the number scenario and a third expert network. To evaluate transfer, for each sort rule we retuned the gating network until expert performance was achieved. The gating network was then tested for 1,000 episodes in the given sort scenario and the relative expert network use was measured as described in the Data analysis section below (Fig. 2.4 and *Appendix*, Fig. 2.9).

2.3.6 Robust memory savings simulations

To demonstrate the DynaMoE network's robust memory savings, we independently trained 10 DynaMoE networks and standard RNNs with the same number of trainable parameters (47,145) in an identical presentation of scenarios. First, the randomly initialized networks trained on 1,250 sorting episodes of the shape sort scenario to ensure convergent performance. This was followed by 1,250 sorting episodes of the color sort scenario, followed by 1,250 sorting episodes of the number sort scenario (same as for networks in Fig. 2.2*C* and *Appendix*, Fig. 2.7). For the DynaMoE network each block of 1,250 sorting episodes with a sort rule was broken into 2 sub-blocks of 625 sorting episodes—in the first 625 sorting episodes, the DynaMoE network

did the first stage of training in which only the gating network is tuned, and for the second 625, the second stage of training was done in which both the gating and new expert networks are tuned simultaneously. After this sequential scenario training, for each standard RNN and the DynaMoE network, we ran 10 independent retrainings on the first scenario encountered: the shape scenario. For the DynaMoE network only the gating network was retuned. To measure how quickly the networks could recover performance in the previously learned rule, the networks were tuned until they reached a performance criteria of average sorts to criterion <10 cards for the last 10 episodes of a single asynchronous thread. The number of sorting episodes required to achieve this performance were measured, as well as the distance traveled in weight space during relearning/remembering the shape scenario (Fig. 2.5 and *Appendix*, Fig. 2.10). For additional methods on comparing network remembering to initial learning see *Appendix*.

2.3.7 Classic WCST simulations with untrained and pretrained networks

To simulate performance on the classic WCST in which the different sorting rule episodes are interleaved randomly, 5 different networks were created. The first network was a standard RNN with randomly initialized weights (*Appendix*, Fig. 2.12*A*). The second network was a standard RNN that was pretrained sequentially on first the shape rule, followed by the color rule, followed by the number rule (*Appendix*, Fig. 2.12*B*). For each rule type, the network was trained until "expert performance," defined as average sorts to criterion <4 over last 100 episodes of single asynchronous thread before switching rules. The third network was a DynaMoE network with three untrained expert networks with randomly initialized weights (Fig. 2.1*C* and *Appendix*, Fig. 2.12*C*). The fourth network was a DynaMoE network seeded with three pretrained expert networks—one pretrained on shape sorting, one on color sorting, and one on number sorting (Fig. 2.1*D* and *Appendix*, Fig. 2.12*D*). Each of these pretrained experts had been trained on the given rule until reaching "expert performance." The fifth network was a DynaMoE network that was pretrained sequentially on first the shape rule, followed by the color rule, followed by the

number rule (Fig. 2.1*E* and *Appendix*, Fig. 2.12*E*). The network started with a gating network and a single expert network with randomly initialized weights. The gating and expert networks were trained simultaneously on the shape rule until "expert performance" was reached. The rule was then switched to the color rule and a new expert network with random weights was added. The gating network was trained for a maximum of 250 sorting episodes and then the new expert was brought online and trained until "expert performance." The same was then repeated for the number rule.

Each network was then trained on the classic WCST, in which rules are randomly interleaved (rules switch after every episode; see full description in Behavioral task section of *Materials and Methods*). The center column of *Appendix*, Fig. 2.12 shows performance of each network over 2,500 sorting episodes of training. Networks with pretraining (*Appendix*, Figs. 2.12*B,D,E*), were also trained for 2,500 sorting episodes on the shape rule (the first rule experienced) to compare each network's ability to "remember" a previously learned rule.

2.3.8 Lesion studies

To perform the lesions studies, we first trained a DynaMoE network identical to the network in *Appendix*, Fig. 2.12*E* as described above. We then implemented one of the following lesions: L0- no lesion; L1- ablation of the reward feedback input to the network; L2- ablation of the action feedback input; L3- both L1 and L2 simultaneously; L4- ablation of varying amounts of the synaptic connections of the "forget gate" component of the LSTM, ranging from 10–100% and denoted by the subscript (e.g. $L4_{0.9}$ has 90% of the synaptic connections ablated) (*Appendix*, Fig. 2.13); L5- ablation of varying amounts of output from the RNN; L6- ablation of synaptic connections to the units used to determine which expert network to use; L8- ablation of the synaptic connects to the unit used to estimate value; and L7- both L6 and L8 simultaneously. For two of the lesions types (L4,L7) we show the full severity spectrum as an example in *Appendix*, Fig. 2.13.

After implementing the lesion, we then tested the full DynaMoE network on the classic

interleaved WCST. We ran 1,000 test episodes and then performed analysis on performance as described in the Results section. For each lesion type, we randomly ablated at each level of severity 10 times and analyzed average behavior, since lesions of specific connections or units within a given region may have differential effects. Errors per episode in Fig. 2.6 was the total number of errors in a sort rule episode. Perseveration proportion was calculated as the proportion of increase in total error due to change in perseveration errors. We defined perseveration errors as incorrect card sorts immediately following the inevitable error trial after a rule change, which would have been correct according to the previous rule [Gre93]. The inevitable error trial refers to the first trial which receives feedback that a sort according to the previously correct rule is now incorrect, signaling a rule change has occurred. We note that there are more complex methods for scoring perseverations on the WCST largely due to the ambiguity introduced by cards that can sort to multiple stacks [GB48; Mil63; FHF91; Gre93]. Practically, we categorized an error as a perseveration if it immediately followed an inevitable error trial, another perseveration error, or an unbroken streak of perseveration errors and correct sorts with ambiguity that included the previous sort rule. Using the same criteria, we also tested the effects of the lesions with the subset of cards excluding ambiguous cards, as in the Modified Card Sorting Task (MCST or MWCST) [Nel76; Mac+15] and found qualitatively similar results (*Appendix*, Fig. 2.14).

2.4 Results

To demonstrate the properties of our framework we chose the Wisconsin Card Sorting Task (WCST), a neuropsychological assessment of PFC function commonly used in clinic [Buc+05; Mac+15; Mit+05; GB48]. In the WCST, a subject is required to sequentially sort cards according to one of three possible sorting rules: shape, color, number (Fig. 2.1*B*; see *Materials and Methods* for full description). The sorting rule is not explicitly given, but rather must be discovered through trial and error. After each attempted card sort, the subject receives feedback as to whether the sort was correct or incorrect. After a set number of correct card

sorts in a row, the sort rule is changed without signal, requiring the subject to adapt behavior accordingly [Mac+15; GB48]. Performance can be measured by the number of attempted card sorts until the episode termination criterion is achieved ("sorts to criterion;" 3 correct sorts in a row in our simulations), with fewer attempted sorts representing superior performance.

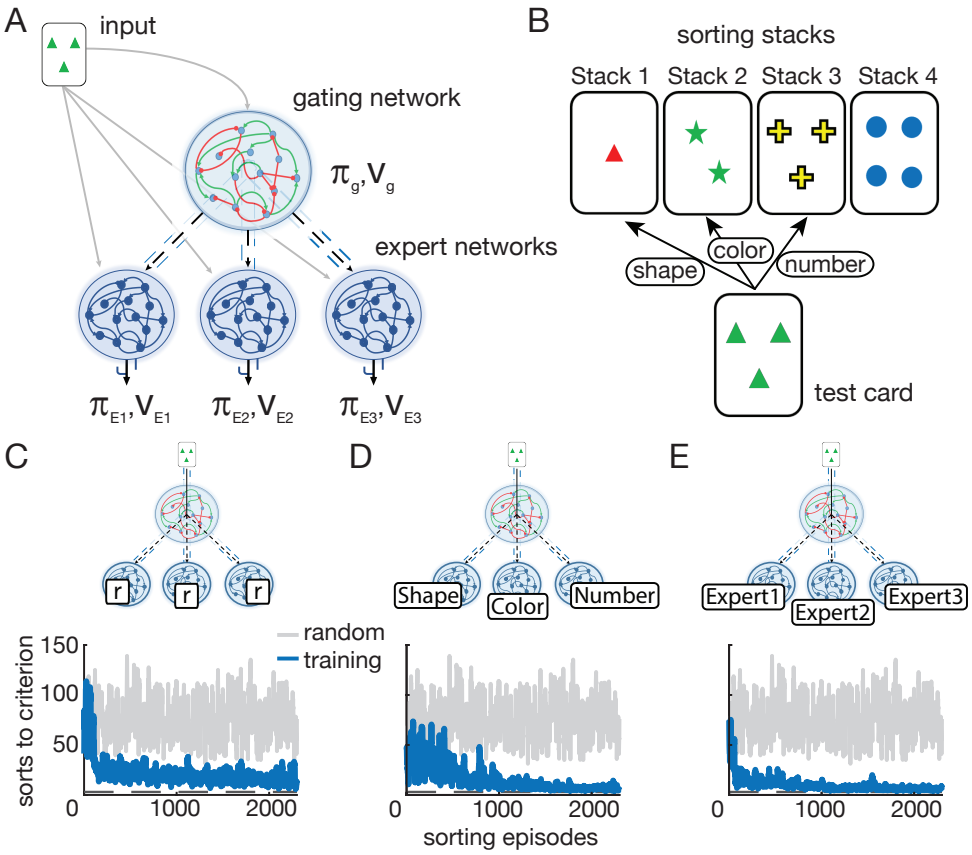


Figure 2.1. DynaMoE network structure and the WCST. (*A*) The DynaMoE network is in the MoE family of networks. A gating network takes input and outputs a decision of which expert network to use (π_g) and a value estimate (v_g). The chosen expert network (e.g. E1) takes input and outputs an action to take (π_{E1})—for the WCST, in which stack to place the current card—and a value estimate (v_{E1}). (*B*) The WCST. The subject must sort the presented test card to one of four stacks by matching the relevant sort rule. The subject continues to attempt sorting test cards until achieving the termination criterion (correctly sorting a given number of cards consecutively). (*C–E*) MoE networks on the classic WCST. (*C*) MoE network with 3 experts achieves good performance quickly and slowly improves further over time. (*D*) MoE network with pretrained experts on the sort rules also learns quickly reaching near perfect performance faster. (*E*) DynaMoE network pretrained sequentially on the sort rules learns rapidly and reaches near perfect performance fastest. In all plots, blue traces are from networks during training and grey traces are random behavior for reference. Grey dotted line is the minimum sorts to criterion.

The WCST requires the PFC's abilities to encode, store, and access multiple schemas. The task requires a recognition of "rule scenarios" (a form of "set learning") and flexible adaptation through reinforcement signals to shift with changing rules. Patients with prefrontal damage often have difficulty with this task, with some stereotypically making perseveration errors, indicating an inability to switch rules when given reinforcement [GAT19; Mac+15].

Although many models are able to solve the classic WCST (Fig. 2.1*C–D* and *Appendix*, Fig. 2.12), we sought to use the WCST to help uncover the mechanisms by which the PFC is able to learn and remember multiple schemas in the absence of curated training or supervision. The framework we develop can be generalized to many similar tasks.

2.4.1 The model: dynamic mixture of experts (DynaMoE)

Our neural network architecture combines RNNs used previously to model the function of PFC [Wan+18] with the MoE architecture [Jac+91], and introduces two new features that enable flexible lifelong learning of disparate schemas: a new learning process and repeated focal retuning. Our MoE design uses two specialized networks: a gating network that receives input from the external environment and outputs a decision of which expert network to use; and expert networks that take external input and output an action decision—the card stack to sort the current card in the WCST (Fig. 2.1*A*). To capture the complex dynamics of the PFC, we modeled both the gating network and expert networks as RNNs (long short term memory networks (LSTMs) in our implementation). While other architectures have been used in MoE networks, recent work by Wang et al. [Wan+18] demonstrated the ability of RNNs to reliably store biologically realistic function approximators when trained by reinforcement learning that mimic animal behaviors.

Using this network architecture, we first introduce a new learning process (Fig. 2.2*B*). Our neural network begins as a gating network with a single expert network (Fig. 2.2*A*). As it gathers experience in the world, it learns in series of 2-step tunings. When the neural network experiences a scenario (e.g. a series of card sorts in the WCST), it first tunes its gating network to attempt to solve the problem by optimally delegating to expert networks, much as a traditional

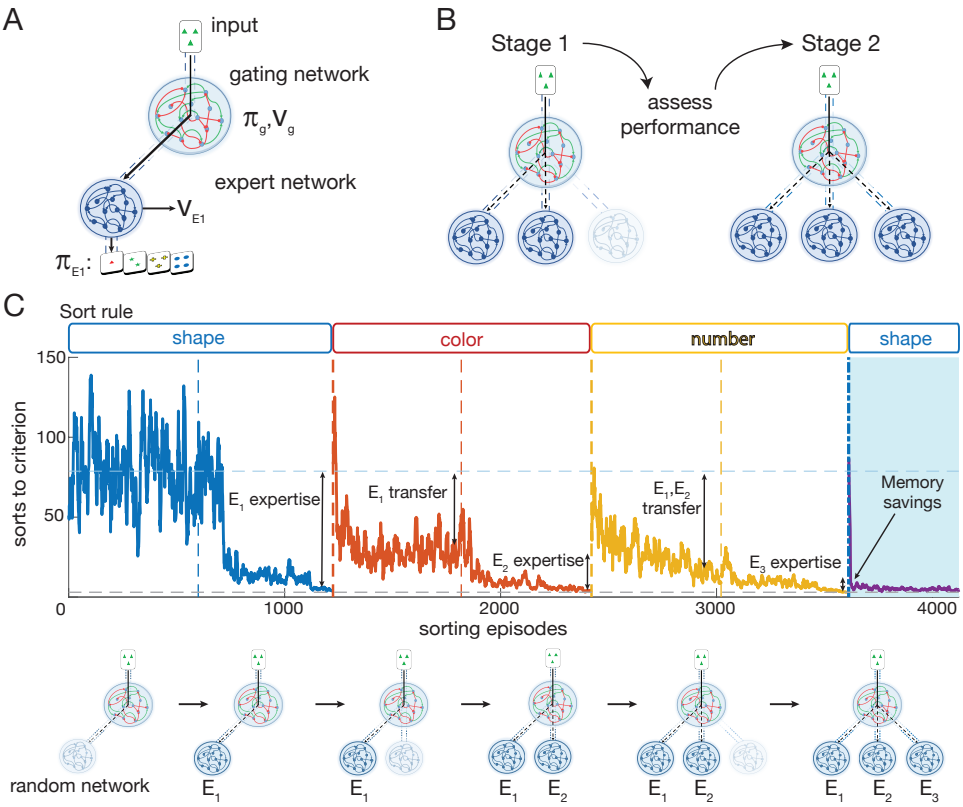


Figure 2.2. Training of a DynaMoE network. (*A*) DynaMoE begins with a gating network and a single expert network, E_1. Both the gating and expert networks train by reinforcement learning, outputing a predicted value (v_g and v_{E1}) and policy (π_g and π_{E1}). (*B*) DynaMoE's 2-step learning process. In Stage 1 the gating network retunes to attempt to solve the task at hand with current experts; if performance is unsatisfactory, the network adds an additional expert in Stage 2 which preferentially trains on tasks that could not be solved by other experts. (*C*) A sample training trajectory of a DynaMoE network presented with sequential periods of sorting rules in the WCST. A randomly initialized DynaMoE begins in the shape sorting scenario. First the gating network is tuned alone. In the second step of learning, the first expert network, E_1, is trained (second half of the blue curve). The sort rule is switched to color (red curve) and the same 2-step training process is repeated; followed by the number sort rule (yellow). The improved performance between the first and second stages of training in each sort rule scenario results from expert training. The improved performance from gate retuning results from transfer learning from past experts and increased network capacity. The purple curve shows how DynaMoE rapidly "remembers" past experience due to robust memory savings. The schematic below the graph shows the progression of the DynaMoE network as it experiences the scenarios. Each stage of training above was done for 625 sorting episodes to display convergent learning behavior.

MoE model would. If some combination of expert actions results in satisfactory performance, no further learning is necessary. If, however, the experiences are sufficiently novel such that

no combination of the current expert networks' outputs can solve the task fully, the network then brings online a latent untrained expert (Fig. 2.2*B*–*C*). The new expert is trained along with the gating network, resulting in a new expert that handles those problems that could not be solved with previous experts. Importantly, this training procedure is agnostic to the order of training scenarios presented and does not require any supervision. Instead, given only the desired performance criteria (e.g. level of accuracy) and limit of training duration per step (how long to try to solve with current experts), our neural network dynamically tunes and grows to fit the needs of any scenario it encounters (Fig. 2.2*C*). The learning curves in Fig. 2.2*C* reveal two prominent features. First, the speed of learning is successively faster for the second (55.8% improvement in performance on color sorting with only gate retuning; "E_1 transfer" in Fig. 2.2*C*) and third (77.5% improvement in performance on number sorting with only gate retuning; "E_1,E_2 transfer" in Fig. 2.2*C*) scenarios, which is a form of transfer learning. Second, after learning all three scenarios, relearning the first scenario (shape sorting) was rapid, a form of memory savings.

The second new feature is repeated retuning of the gating network. Training standard neural networks on new tasks leads to overwriting of network parameters resulting in catastrophic forgetting [Fre99] (*Appendix*, Fig. 2.7). By decomposing a single network into a hierarchy of gating and expert networks, we are able to separate the memory of the neural network into the "decision strategy" (gate), which maps between inputs and experts, and the "action strategies" (experts), which map from input to actions. The hierarchical separation enables repurposing expertise through combinatorial use of previously acquired experts and a natural means to confine memory overwriting to a small portion of the neural network that can be easily recovered through repeated retuning. Experimental evidence suggests areas of the brain similarly support different levels of plasticity, with regions higher in hierarchical structures exhibiting increased plasticity [HB19]. The resulting network exhibits memory savings [Nel85] that remain robust to new learning and lead to rapid "remembering" rather than relearning from scratch (compare purple curve in blue shaded region in Fig. 2.2*C* and *Appendix*, Fig. 2.7).

We found that the implementation of these two new features in a hierarchical MoE com-

posed of RNNs results in an architecture that organically learns by reinforcement relative to past experiences and preserves memory savings of past experiences, reminiscent of prefrontal cortex. Importantly, when presented with the classic interleaved WCST, our network (Fig. 2.1*E*) learns just as fast or faster than standard RNNs and traditional MoE networks (Fig. 2.1*C–D* and *Appendix*, Fig. 2.12). We next sought to understand how our dynamic architecture enabled the observed transfer and savings.

2.4.2 Transfer learning: DynaMoE seeded with pretrained experts

To probe how the DynaMoE network implements transfer learning, we first created an easily interpretable scenario in which two expert networks were separately pretrained on specific rule sets of the WCST; one on shape sorting (E_{shape}) and another on color sorting (E_{color}) (Fig. 2.3*A*). We then seeded a DynaMoE network with the two pretrained experts and a randomly initialized untrained expert, and introduced it to the third rule, number sorting, and studied its behavior.

Reflexively, one may speculate that a DynaMoE network with a shape sorting expert, a color sorting expert, and a random network would perform no better on number sorting than with only a random network, since number sorting is seemingly independent of shape or color. Somewhat surprisingly, we found this was not the case. After tuning the gating network, the DynaMoE network with pretrained experts performed drastically better than without them, nearly reaching perfect performance (Fig. 2.3*B*). We found that the gating network learned to identify cards for which the shape or color sort matched the correct number sort, and allocate them to the corresponding expert. For example, a card with 1 blue triangle would be sorted to Stack 1 in both the shape (triangle) and number (one) scenarios ("shape-match-number"). Similarly, some cards, for example the card with 1 red circle, would be sorted to Stack 1 in both the color (red) and number (one) scenarios ("color-match-number"). The gating network learned to map these cards to E_{shape} and E_{color} to perform correct card sorts in the number rule (Fig. 2.3*C*). Only cards for which the number sort did not match the shape or color sort ("no-match-number") were

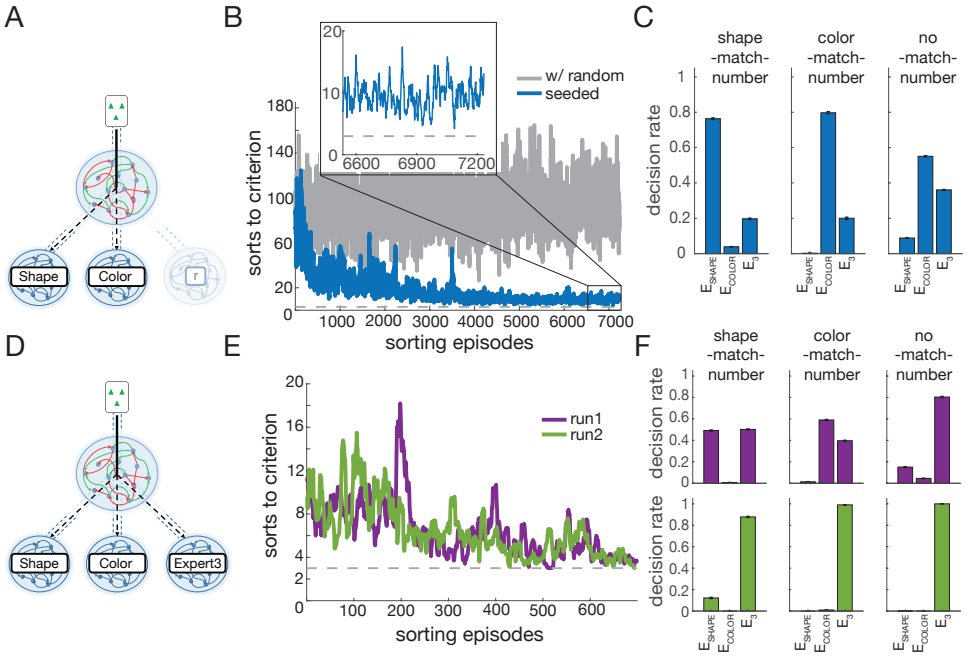


Figure 2.3. Transfer learning with a seeded DynaMoE network. (*A*) A DynaMoE network seeded with pretrained shape and color experts and a randomly initialized untrained network. (*B*) The DynaMoE network from *A* achieves near perfect performance in number sorting when only the gating network is trained (blue) in contrast to a network with only an untrained expert (grey). *Inset* shows that performance of the seeded network does not reach the minimum sorts to criterion (grey dash) without training the third expert network. (*C*) The proportion of cards allocated to each expert network after the training in *B* in three different subsets of the number sort rule: shape-match-number, color-match-number, and no-match-number. (*D*) Seeded DynaMoE network with trained Expert3 network. (*E*) Performance (measured by decrease in sorts to criterion) of two example training runs from the same initial network (*A*–*C*) that result in different end behavior (see *F*). The performance of both runs improves from DynaMoE with an untrained expert (*B, Inset*) and are indistinguishable from each other. (*F*) (*Top*) Proportion of experts used in same subsets of number sort rule as in *C* for an example run (run 1). A varying decision rate for experts is used depending on the scenario subset. (*Bottom*) Proportion of experts used for a second example run (run 2). The new expert (E_3) is used regardless of subset of number sort rule. See *Appendix*, Fig. 2.8 for all 10 runs. Error bars are standard deviation over 1,000 test episodes after training. Absence of bar indicates zero selections of the given expert during testing.

unsolvable with either E_{shape} or E_{color}—for these cards, the gating network used a mixture of the shape, color, and untrained expert networks (Fig. 2.3*C* rightmost panel), since no expert could reliably sort these cards correctly. The network had learned to exploit a hidden intrinsic symmetry between the features in the task to enhance performance.

Consequently, when the new expert network was brought online and trained (Fig. 2.3*D*), the gating network allocated a large proportion of "no-match-number" cards to the new expert (E_3) (Fig. 2.3*F*, rightmost panels). E_3's expertise thus became number sorting cards that do not match shape or color sorts. Interestingly, this demonstrates a form of transfer learning. The gating network learned to use existent experts to find partial solutions for new problems, leaving unsolvable parts of the problem to be disproportionately allocated to the new expert in the second step of training. New experts thus learn relative to old experts, building on prior knowledge and experience.

In practice, the expertise of E_3 varied between number sorting predominantly "no-match-number" cards and all cards. This likely reflects a trade-off between the complexity of mapping functions the gating and expert networks must learn. In the number sorting scenario, the gating network can learn to map each card type to the appropriate expert or the simpler function of mapping all cards to E_3; E_3 in turn learns to number sort only "no-match-number" cards or all cards. This highlights a trade-off that occurs in biological systems like the brain. We may be able to solve a new problem by piecing together numerous tiny bits of completely disparate strategies, but as complexity of the mapping function increases, at some point it becomes more efficient to simply learn a separate strategy for the new problem, allocating dedicated memory for it.

We found that in the first stage of training, tuning of the gating network consistently led to a mapping function that allocated the vast majority of "shape-match-number" cards to E_{shape} and "color-match-number" cards to E_{color} (Fig. 2.3*C*). "No-match-number" cards were allocated between all three experts. After the second stage of training in which both the gating network and E_3 are trained, we found that the "no-match-number" cards were almost entirely allocated to E_3 as expected (Fig. 2.3*F*, rightmost panels). We found that usage of experts for "shape-match-number" and "color-match-number" cards varied across different training runs (Fig. 2.3*F* and *Appendix*, Fig. 2.8). To see how often training led to different expert network decision rates, we ran the second stage of training 10 times from the same initial network that had gone through the first stage of training. Usage of the relevant pretrained expert (e.g. E_{shape}

for "shape-match-number" cards) ranged from as much as 65% to as low as 1%, representing end behavior in which E_{shape} and E_{color} continued to be used or in which E_3 was used almost exclusively (run1 and run2 in Fig. 2.3*F*, respectively). The non-relevant expert (e.g. E_{color} for "shape-match-number" cards) was rarely ever used (0–5%). This shows that while DynaMoE networks support pure transfer, the degree of transfer learning implemented depends on network capacity, learning efficiency, and the stochastic nature of learning. All networks achieved the same near perfect performance stop criteria within similar numbers of sorting episodes (Fig. 2.3*E*; see *Materials and Methods*).

2.4.3 Transfer learning: organic case

To probe how a DynaMoE network naturally implements the transfer learning described in the previous section, we trained 10 DynaMoE networks independently from scratch through sequential experiences of the different rules of the WCST. Each network began with an untrained gating network and expert network (E_1). The DynaMoE networks were then trained on shape followed by color and then number sorting, adding a new expert in each new sort rule scenario (see *Materials and Methods*).

As expected from the result in the previous section, we found that the expert networks were not pure rule sorters, but rather had learned an expertise in a mixture of rule types relative to the other experts. For each sort rule scenario, one expert network was used preferentially (E_1 for the first rule experienced, E_2 for the second, etc.), which we refer to as the "dominant expert network" for that sort rule scenario. To quantify the degree of transfer learning utilized, we measured the usage of all 3 expert networks in the different sorting scenarios. For each sort rule scenario, the gating network was retuned until "expert performance" was once again attained. We then measured the usage of each of the non-dominant expert networks with respect to usage of the dominant expert network. Although the magnitude of relative usage varied between independent runs, a consistent pattern emerged. In the shape sort scenario—the first scenario encountered with only E_1—E_2 and E_3 were used very little or never (Fig. 2.4*A, Left*). For the

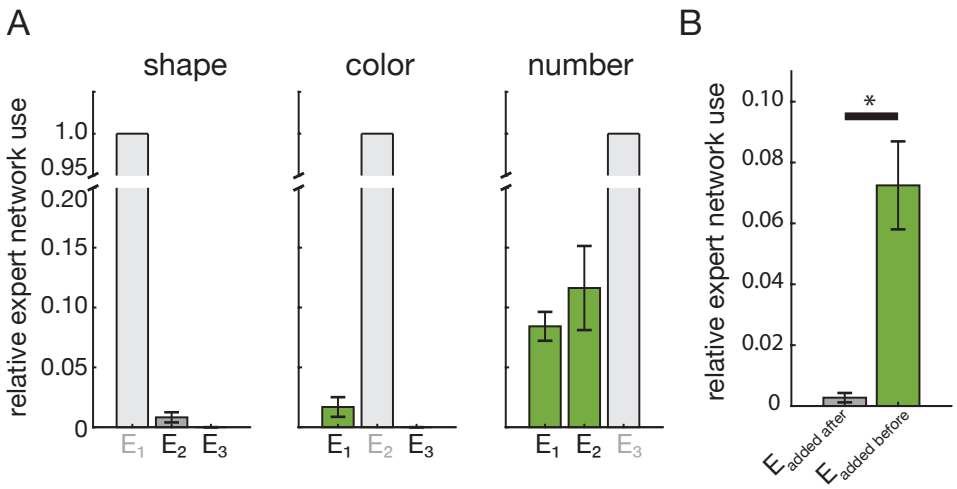


Figure 2.4. Transfer learning in an unseeded DynaMoE network. (*A*) The relative use of each expert network in each sort rule normalized to the dominant expert for the sort rule from 10 independent DynaMoE networks trained in a sequential training regimen (see *Materials and Methods*). The greyed out expert network label with lightest grey bar of value 1 indicates the dominant expert network for each sort rule. Darker grey bars indicate usage of experts that were not present during initial training of the given sort rule (e.g. E_2 and E_3 for the shape rule). Green bars indicate experts that were present during initial training of the given sort rule (e.g. E_1 and E_2 for the number rule). Absence of a bar indicates the given expert was never used. Error bar is SEM over 10 independent runs (see *Appendix*, Fig. 2.9). (*B*) Aggregated bar plot from *A* grouped by whether the expert was added before or after initial training on the rule. Use of experts present during initial training of a rule indicates transfer learning (green bar), while use of experts not present during initial training indicates non-transfer usage (grey bar). The usage of experts added before was significantly higher ($p = 1.16e-05$, Student's t-test) than that of experts added after initial training on a rule. Error bar is SEM.

second scenario encountered—color sort scenario—E_1 was used a small amount, and E_3 was never used (Fig. 2.4*A, Center*). Finally, for the third scenario—number sort scenario—E_1 and E_2 were used a small but significant portion of the time (Fig. 2.4*A, Right*).

This trend of increased usage of experts that were present during the learning of a rule compared to experts added afterward strongly indicates transfer learning as the DynaMoE network encountered new scenarios (Fig. 2.4*B*). Newly added experts predominantly trained on examples that the other experts could not solve. Thus when the gating network was retuned to solve a scenario later, it continued to use the previously added experts. In contrast, if an expert was added after the learning of a scenario, all the knowledge to solve the scenario was already

contained in the existent experts, so the expert added after learning was rarely used. This shows that new experts were trained relative to knowledge contained by existent experts. Furthermore, while the aggregated expert use percentages clearly show the presence of transfer learning, they mute the degree of transfer learning adopted by some individual networks (*Appendix*, Fig. 2.9).

2.4.4 Robust memory savings

A critical feature of the PFC is the ability to retain knowledge from past experiences in the form of learned connectivity patterns [EGM12]. Many neural network models suffer from castastrophic forgetting [Fre99], overwriting information from previous experiences. Put in terms of network parameters, when such networks retune weights to solve new problems, they move to an optimal point in weight space for the current task which can be far away from the optimal space for previous tasks.

In contrast, DynaMoE networks, like the PFC, maintain near optimal configuration for previously experienced scenarios, exhibiting "memory savings" [Nel85]. The hierarchical architecture of DynaMoE networks confines memory loss to a small flexible portion of the network: the gating network. If a scenario has been encountered before, retuning the gating network to optimal configuration is rapid, requiring only a small number of reinforcement episodes (Fig. 2.5*A–B* and *Appendix*, Fig. 2.11). Retuning the gating network requires much less movement in weight space compared to standard RNNs, since tuning is confined to only the gating network. This is in stark contrast to standard neural networks which can require complete retraining (Fig. 2.5*C–D* and *Appendix*, Fig. 2.10).

To measure the memory savings of DynaMoE networks, we sequentially trained networks with identical presentations of, first shape, then color, then number sorting scenarios (see *Materials and Methods*). We then tested how many sorting episodes of reinforcement were required for the network to regain expertise in the first sorting rule it experienced (shape). As Fig. 2.5*A–B* shows, DynaMoE networks required 78% fewer episodes to regain expertise than standard RNNs ($p = 2.49e-11$, Student's t-test). The number of episodes required to remember

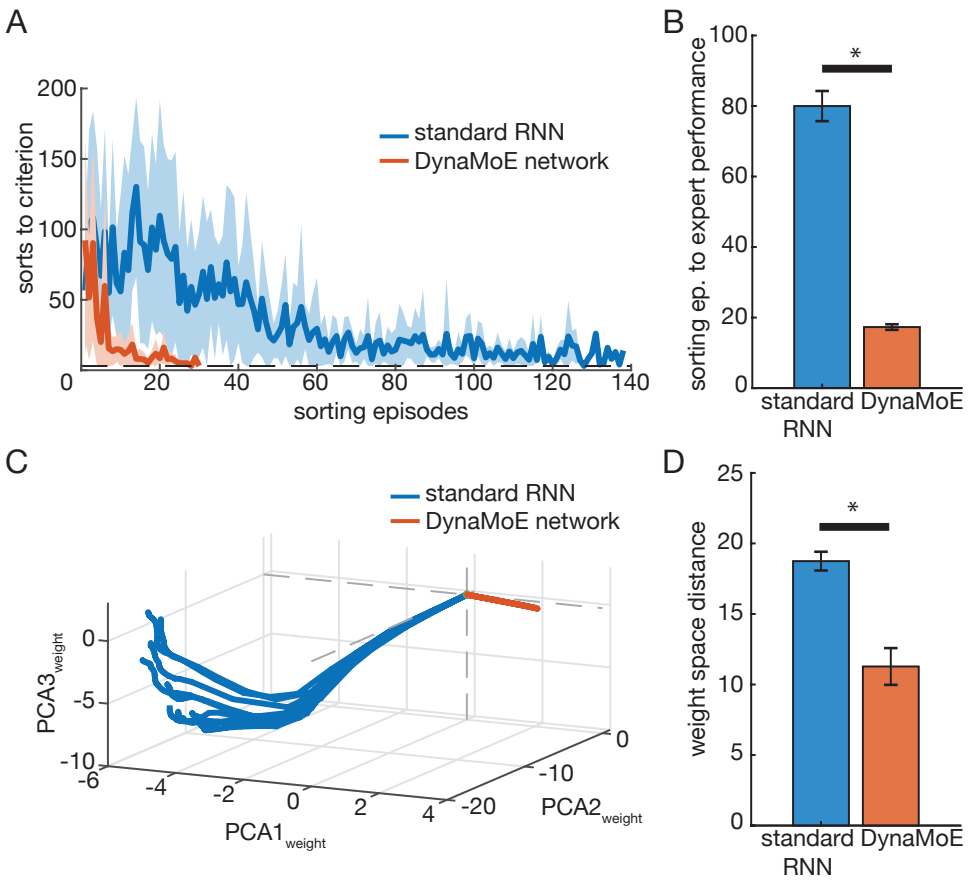


Figure 2.5. Robust memory savings of DynaMoE. (*A*) Example of performance over sorting episodes of retraining of standard RNN (blue) and DynaMoE network (orange) on a previously encountered task. Shading indicates standard deviation over 10 independent retraining runs. (*B*) Average number of sorting episodes required until expert performance for standard RNN and DynaMoE networks over 10 independently trained networks of each type. DynaMoE networks require 78% fewer sorting episodes to remember ($p = 2.49e-11$, Student's *t*-test). (*C*) Visualization of top three principal components of weight space for 10 relearning/remembering trajectories of an example standard RNN and a DynaMoE network trained sequentially. (*D*) Euclidean distance between networks before and after remembering previously learned rule in full weight space (average of 10 independently trained networks of each type). DynaMoE network moves 40% less in weight space compared to the standard RNN ($p = 7.43e-05$, Student's *t*-test). Error bars are SEM.

was drastically fewer than when it first learned the rule, whereas standard RNNs improved only slightly compared to when they first learned the rule (*Appendix*, Figs. 2.7, 2.11, 2.12). While standard RNNs nearly completely overwrote the information learned through initial training, DynaMoE networks preserved their memory and only required brief reinforcement for the gating

network to remember how to allocate cards to experts.

To measure the weight changes required to regain optimal performance, we measured the distance in weight space each of the networks traversed when remembering the shape sort rule after sequential training. DynaMoE networks traversed 40% less distance in weight space to reach optimal performance compared to standard RNNs ($p = 7.43e-05$, Student's t-test; Fig. 2.5*C–D* and *Appendix*, Fig. 2.10). Even after sequential training, DynaMoE networks remain relatively close in weight space to the optimal performance configurations on all previously experienced tasks. In contrast, standard RNNs moved far from their initial optimal point in weight space for the shape scenario, resulting in movement of nearly equal distance when relearning the shape scenario as when initially learned (*Appendix*, Fig. 2.10).

2.4.5 Lesions of DynaMoE cause PFC lesion-like impairments

The DynaMoE framework provides an opportunity to understand how disruptions to specific functional aspects of the PFC and related areas can lead to different neuropsychological impairments observed in clinical cases. Numerous clinical and neuroimaging studies have indicated regional specialization within the PFC, yet evidence from human studies is invariably messy, involving overlapping brain regions and varying degrees of impairment in different aspects of tasks [Buc+05; Mac+15; Dim+99]. Our framework enables targeted disruption of specific functional components of our network that may help clarify the underlying organization of the human PFC. The WCST has served as a standard clinical assessment to evaluate PFC impairment [Mac+15] making it an ideal task with which to analyze functional consequences of various lesion types.

To assess how lesions of our network architecture could result in behavioral impairments, we damaged specific regions of the gating network of our architecture. Importantly, in our lesion studies the expert networks were unperturbed, leaving available the action strategies to perfectly perform the task. This characteristic is often seen in patients with prefrontal damage: although they have difficulty with the full WCST, if explicitly told which sort rule to use, patients are

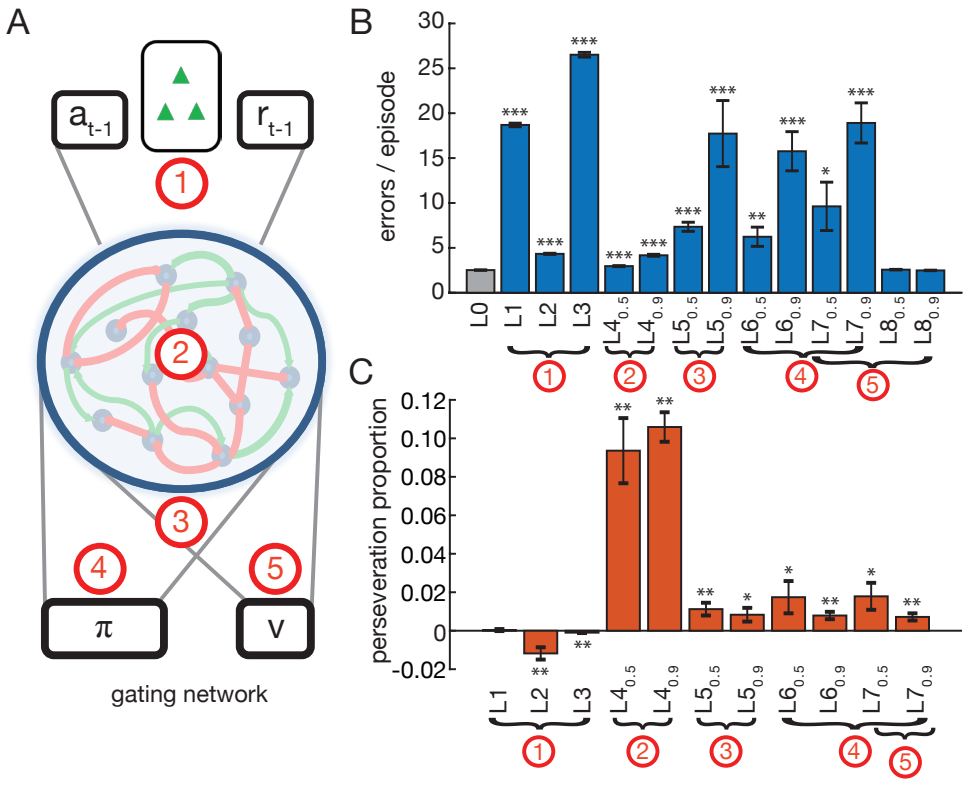


Figure 2.6. Different lesion-induced error modes of DynaMoE gating networks. (*A*) Map of lesioned regions in DynaMoE's gating network. Three lesions of input (region 1) were done (L1–L3), one lesion of the network dynamics (region 2—L4), one lesion of network output (region 3—L5), one lesion of decision determination (region 4—L6–L7), one lesion of value determination (region 5—L7–L8). (*B*) Average number of errors per episode for each lesion type. L1 is ablation of reward feedback from previous trial; L2 is ablation of action feedback from previous trial, L3 is simultaneous L1 and L2 lesions; $L4_{0.5}$ is ablation of 50% of connections to forget gate of network and $L4_{0.9}$ is ablation of 90%; L5 is ablation of output units; L6 is ablation of connections to decision units (π); L8 is ablation of connections to value unit (v); L7 is simultaneous L6 and L8. Asterix indicates significant difference from no lesion (L0) (Student's *t*-test *:$p < 0.05$; *:$p < 0.01$; ***:$p < 0.001$) (*C*) Proportion of increase in errors that were perseveration errors for lesions that caused significant increase in errors. Asterix indicates confidence interval (CI) excluding zero (*: 95% CI; **: 99% CI). All error bars are SEM.

often fully capable [Mac+15; RSM02]. We first trained DynaMoE networks on each rule type, and then the classic interleaved WCST (*Appendix*, Fig. 2.12*E*; see *Materials and Methods*). We then lesioned the gating network and performed testing on the classic WCST to assess changes in performance and behavior.

Lesions were targeted to five different regions within the gating network (Fig. 2.6*A*):

inputs to the network (red region 1 in Fig. 2.6*A*)—ablation of reward feedback (L1), action feedback (L2), or both (L3); internal network dynamics (region 2)—ablation of varying numbers of synaptic connections to the "forget" gate of the LSTM ($L4_{0.5}$ and $L4_{0.9}$ were 50% and 90% ablations of synaptic connections, respectively; *Appendix*, Fig. 2.13*A–B* for full range); output of the network (region 3; L5); and areas downstream of the network—ablation of synaptic connections to the units that determined which expert network to use (region 4; L6), to the unit that estimated value (region 5; L8), or both (L7; *Appendix*, Fig. 2.13*C–D* for full range). Since lesions could potentially have differential effects depending on the specific disruptions incurred, we performed each lesion 10 times and measured the average effect.

We found that different lesion types resulted in different degrees of impairment, ranging from no change in error rate (e.g. $L8_{0.9}$, $p = 0.3664$, Student's t-test) to 10.5 fold more errors (L3, $p = 2.1268e-25$, Student's t-test) than before the lesion (Fig. 2.6*B* and *Appendix*, Table 2.1). Since perseverative errors are a signature of some prefrontal lesions, particularly associated with dlPFC impairment, we measured the proportion of the increase in error rate that was due to perseverative errors ("perseveration proportion"; a negative value indicates a decrease in perseverative errors relative to no lesion). Fig. 2.6*C* shows the variability in perseveration proportion for the lesions that caused a significant increase in error rate. To ensure that the error profiles we observed were not an artifact due to the presence of ambiguous cards for which sorting by multiple rules could result in the same action decision, we also tested the lesioned networks with the card deck from the Modified WCST (MWCST) [Nel76; Mac+15], in which all ambiguous cards are removed, and found qualitatively similar results (*Appendix*, Fig. 2.14).

The neuropsychological impairment profiles defined by increase in total error and perseveration proportion reveal different lesion-specific error modes that mirror the different error modes observed from patients across the range of prefrontal lesions (Fig. 2.6 and *Appendix*, Table 2.1). Overall, lesions grouped qualitatively into three categories: lesions that caused often substantially increased total error rate (1.72–10.51x), a small proportion of which were perseverative errors (-1.18–1.79%) (regions 1,3,4; L1-L3,L5-L7); lesions that caused a small but

significant increase in total errors (1.18–1.65x), a large proportion of which were perseverative errors (9.36–10.59%) (region 2; L4); and lesions that caused no change in error rate (region 5; L8).

Our lesion results provide a roadmap with which to interpret and understand the variety of error modes observed in human patients with prefrontal damage due to trauma or disease. While the PFC as a whole has been definitively linked to set-shifting and cognitive flexibility, localization of functional components to specific subregions remains unclear. Lesions throughout the prefrontal areas have been associated with impairments observed in the WCST, ranging from no change in error rate to large increases in perseverative and non-perseverative error rates similar to the range of behavioral outcomes resulting from our lesions [Mac+15]. Canonically, though with mixed evidence, impairment of the dlPFC is associated with increased error rate on the WCST, particularly perseveration errors. Our lesion study indicates this behavioral phenotype maybe due to impairment of gating network dynamics, suggesting the dlPFC may contribute to a gating-like mechanism within the PFC. Interestingly, the lesions of components inside the gating network's recurrent connections that caused a specific increase in perseverations only weakly increased the total error rate. In contrast, lesions to input components led to a large increase in total error while perseverations increased relatively less. These contrasting neuropsychological impairments highlight a double dissociation of neural components underlying perseveration errors and total errors, a characteristic also observed in patients.

2.5 Discussion

In this paper we propose a new framework for how the PFC may encode, store, and access multiple schemas from experiences in the world. Like the PFC, the DynaMoE neural network is agnostic to training regimen and does not require "oracle" supervision. We showed how the hierarchical architecture of DynaMoE naturally leads to progressive learning, building on past knowledge. We then demonstrated how DynaMoE networks reliably store memory

savings for past experiences, requiring only brief gate retuning to remember. Finally, we showed how lesions to specific functional components of the DynaMoE network result in different error modes in the WCST, analogous to the error modes described of patients with different forms and severity of prefrontal damage.

The parallels seen between the DynaMoE network and the PFC and related areas encourages investigation into the extent to which these two systems recapitulate each other. Perhaps most poignantly, this comparison puts forth the hypothesis that the PFC may incorporate a gating system that is tuned to optimally combine knowledge from past experiences to handle problems as they are encountered. Some studies have provided evidence for such a functional architecture in the brain [Yao+09], and prefrontal cortical areas in particular [GB11; MMT06; Joh+07; RGH18]. In our model, each "unit" within our recurrent networks represents a population of neurons, and as such, inputs to the network were represented by populations of neurons, each of which receives multiple inputs and responds with mixed selectivity as has been observed in PFC [Rig+13; Man+13]. Our model also suggests that the neural representations of different contextual "rules" can distributed across multiple overlapping subpopulations. In this way, our model adopts a hierarchy of distributed representations in which overlapping neural subpopulations support distributed representations of both lower level sensory information (inputs) and higher level abstract information (context-dependent rules). Experimental investigations that compare the diverse neural population activities of DynaMoE networks to that in relevant prefrontal cortical areas will be fruitful in supporting or refuting our framework. Studies in non-human primates with a WCST analog [MMT06] and related task [Joh+07] have reported specific neural activity patterns associated with set-shifting, suggesting the possibility of direct comparisons to our model; a topic we are currently exploring.

Our model's feature of adaptive growth by adding new expert networks represents the recruitment of neural subpopulations for new learning. Both the assessment of performance and recruitment of new subpopulation bring up interesting parallels in neurophysiological studies (Fig. 2.2b). Error-related negativity and positivity signals are well-known phenomena that are thought

to relate to self-assessment of performance, preceding a switch to an alternative strategy [Wes12]. Research on neuromodulation, neuropeptides, and metaplasticity provide evidence for spatially and temporally regulated plasticity differences in neural subpopulations [MT02; LCD14; RH18; Abr08; VC17], potentially supporting a dynamic, regionally selective learning system like DynaMoE. Elaborating DynaMoE to explicitly model these processes will help untangle how these processes may interact to support behavior and learning.

The DynaMoE model suggests functional relationships underlying organization in the PFC and related areas. The gating network in the model is reflective of the set-shifting functions that are closely linked to activity in dlPFC. Our lesions studies support this association, showing a relative increase in perseveration errors with disruption of the gating network internal dynamics, similar to dlPFC lesions association with increased perseveration errors in patients [Mil63; SL02] and increased dlPFC activity observed in patients during rule shifts [Buc+05; Mac+15; Mit+05]. However, the gating network in our model likely incorporates elements that are distributed anatomically, including processing of error signals attributed to ACC [Joh+07; OH12]. Other elements of our model may correspond to adjunct prefrontal regions—value tracking functions attributed to OFC — mOFC in particular [Lop+17] — may correspond to the value units in both the gating and expert networks [RM14]. The expert networks in our model correspond to functional elements spanning regions of PFC (perhaps downstream of dlPFC) to premotor cortex, culminating in an action decision sent to the motor system to execute in the environment. The different expert networks then represent different effector pathways originating from the gating network signal and extending to the premotor cortex. These "cognitive maps" representing different context-dependent behavioral strategies are characteristic of areas of OFC [Sch+16], perhaps lOFC in particular [Lop+17]. Both the gating network and the effector expert networks also encapsulate cortical-thalamic-basal ganglia loops, routing and processing sensory information and reward information to and from areas of PFC and ACC, nuclei of the thalamus, and areas of the striatum and other basal ganglia regions [Sej+14]. Further comparison between individual functional elements in our model (e.g. expert networks and OFC) will be

helpful in studying the relationship between subregions of the PFC.

Our model also can be integrated with the anterior-posterior cascade model of the frontal regions as proposed by Kochelin and Summerfield [KS07]. From this perspective, the gating network of our model corresponds to the more abstract contextual information in the anterior regions of lateral PFC, while the expert networks correspond to more immediate context sensory processing, taking external input and mapping to an output action. Our model layers into the anterior-posterior cascade scheme, providing a mechanism for encoding and flexible usage with transfer and savings.

Several computational models have provided strong support for a hierarchical organization in the PFC [FB12; AB15; KS07; Bot07]. Combining our model with insights from other models will likely yield further insights into the architecture and functions of the prefrontal areas. Together with Kochelin and Summerfield's cascade model [KS07], Botvinick's model of Fuster's hierarchy [Bot07] suggests that added layers to a hierarchical structure lead to organizing principles that map to higher levels of abstraction and context. Likewise in our model, the gating network sits above the expert networks, processing the higher level context to choose which expert to use. Understanding how the processing of higher and lower levels of the task self-organize within the hierarchical structure, i.e., how the division of labor between the levels is determined, is a fascinating area of future research we are pursuing. Integrating the HER model's [AB15] hierarchical propagation of errors into DynaMoE is also a promising direction, particularly in understanding the roles and relationship of dlPFC and ACC.

Our model also relates closely to previous computational models investigating the emergence of hierarchical rule-like representations [Rou+05; DCK14]. The model proposed by Donoso et al. [DCK14] to study the medial (vmPFC-pgACC, dACC, ventral striatum) and lateral (FPC, mid-LPC) tracks contribution to strategy inference shares features with DynaMoE in storing multiple strategies and allowing combinatorial use. Our model further posits a part of PFC, likely within dlPFC and perhaps spanning to parts of ACC, supports nonlinear and dynamically evolving combinations of stored strategies, enabling more powerful transfer learning and

reducing memory requirements. To gain a deeper understanding of how multi-schema inference is done in the prefrontal regions, further fMRI studies elaborating on the paradigm used by Donoso et al. [DCK14] will be important to compare features of their model to those of the DynaMoE model.

Our lesion studies motivate further investigation of functional specialization in the PFC through comparison of our framework and clinical, experimental, and neuroimaging studies [Buc+05; Mac+15]. Clinical and experimental studies have yielded unclear and sometimes contradictory findings due to the anatomical inseparability of PFC functions [Mac+15]. Although studies have most clearly linked perseverative abnormalities to dlPFC, similar abnormalities can be observed in compulsive behaviors like alcohol consumption, which depend on strategies generated by the mPFC that predict individual behavioral patterns [Sic+19]. Our model provides full access to the underlying structure enabling targeted studies to use as a reference for interpreting human and animal studies. Further comparison of our framework with in-depth phenotypic analyses across various tasks may help us understand the functional organization of the PFC and the consequences of disruptions due to trauma and disease.

Our lesion analysis also motivates future studies on adaptation to lesions. In the present study we focused on lesions after learning was complete, since most clinical case reports describe testing of patients after acute injury. In clinic, it is also important to understand how patients may cope and adapt after a lesion has occurred. The DynaMoE framework may be useful for studying the effects lesions on learning and adaptation.

The DynaMoE framework also has interesting implications for areas of machine learning. It combines the advantages of prior models that leverage transfer learning in the MoE architecture with reinforcement learning [Sin92a; Sin92b; DL17; GSL18] and Bayesian nonparametric MoE models [RG01; WR95; SN96; ACT17; FFW96; KMG06]. DynaMoE's organic, unsupervised implementation of transfer may be useful for intractable problems that may be handled piecewise in a way that may be non-obvious to an "oracle" supervisor. By letting the model learn how to grow and structure itself, our framework puts the burden of optimally solving complex

problems on the algorithm. This may significantly improve progress by removing the need for careful curation of training data and training regimen.

The form of transfer learning demonstrated by our dynamic architecture—acquiring new knowledge (new expert) based on indirect knowledge of what other parts of the network (old experts) know—has not been reported before to our knowledge. This form of transfer learning is reminiscent of "learning by analogy," a learning skill humans are very good at, but machines continue to struggle with [Hil+19; KV18]. Through our framework, this dynamic form of transfer could be extended to much larger networks, utilizing a myriad of experts. Such a framework could be useful both as a model of the brain and for machine learning applications.

Finally, our framework provides a new method for lifelong learning and memory. Major challenges persist in developing methods that do not get overloaded, but also scale well to lifelong problems [Par+19]. Similar to "grow-when-required" algorithms, our network adds capacity when necessary. However, our network also leverages already acquired knowledge to help solve new problems, reducing demand for growth. This feature supports scalability, which both the brain and machine learning methods must support given their limited resources. Elaborating and adapting DynaMoE to more complex tasks and incorporating other techniques such has simultaneous combinatorial use of experts will lead to exciting steps forward in lifelong learning.

Chapter 2, in full, is a reprint of the material as it appears in: Ben Tsuda, Kay M. Tye, Hava T. Siegelmann, and Terrence J. Sejnowski. A modeling framework for adaptive lifelong learning with transfer and savings through gating in the prefrontal cortex. *Proceedings of the National Academy of Sciences*, 117 (47), 29872-29882, 2020. The dissertation author was the primary investigator and author of this paper.

2.6 Appendix

2.6.1 Behavioral task example

As an example of the WCST, a subject could be given a sample card with 3 green triangles. If the operating rule is shape, the card should be placed in Stack 1 (matching the triangle shape of the 1 red triangle). If the operating rule is color, the card should be placed in Stack 2 (matching the green color of the 2 green stars). If the operating rule is number, the card should be placed in Stack 3 (matching the number of the 3 yellow crosses)(Fig. 2.1*B*).

2.6.2 Details of network architectures

Our "standard RNN" architecture consists of a single LSTM network. As input, the RNN takes the card from the current time step and the action and reward from previous time step. The RNN sends output to a policy layer and a value layer. The policy layer implements a softmax transformation over the possible actions to determine the action to take. The value layer outputs a single number value estimate used to calculate the advantage term for the objective function.

In our DynaMoE architecture, the gating network consists of a single LSTM network. The gating network takes as input the card from the current time step and the action and reward from the previous time step. It sends output to two parallel layers, one implementing a softmax transformation over the possible expert networks to determine which expert network to use, and the other estimating the value of action to be taken. Each expert network consists of a single LSTM network that takes the same input as the gating network, and sends output to two parallel layers. One of these layers implements a softmax transformation over all possible actions to determine the action to take, and the other gives a value estimate of that action. The gating network and the expert networks are each optimized by their own objective function in the form described in the Reinforcement learning training section of *Materials and Methods*.

2.6.3 Memory savings: comparison of network training required for "remembering" vs initial learning

To compare how many sorting episodes were required for each network type to re-learn/remember the shape scenario after the sequential rule training versus learn the scenario for the first time from scratch (initial learning), we trained 10 randomly initialized standard RNNs and 10 randomly initialized DynaMoE networks with a single expert network until they achieved the performance criteria of average sorts to criterion <10 cards for the last 10 episodes of a single asynchronous thread. We then compared the average number of sorting episodes required to achieve this criteria to the number required for relearning/remembering as described in the Results section (*Appendix*, Fig. 2.11).

2.6.4 Additional data analysis

For the organic transfer learning (without pretraining) analysis, after sequential training, the gating network was retuned for each rule type. The "dominant expert network" was then defined as the expert network that received majority of the decisions after retuning in that scenario. We found the newly added expert was always the "dominant expert network:" first expert for the shape rule, second expert for the color rule, and third expert for the number rule. Of the 10 independent training runs, the minimum usage of the first expert in the shape scenario was 96.45% and maximum of other expert usage was 3.55%; minimum usage of second expert in color scenario was 93.19% and maximum of other expert usage was 6.81%; minimum usage of third expert in number scenario was 67.96% and maximum of other expert usage was 24.34%. To calculate the relative expert network use shown in Fig. 2.4 and *Appendix*, Fig. 2.9 we normalized the non-dominant expert usages to the dominant expert usage in each scenario. The aggregate relative usage of the expert networks in Fig. 2.4 was calculated by combining the relative usage of all non-dominant experts that were present during initial learning of a rule (first expert for color and number training; second expert for number training) to represent experts present during initial training (green bar), and the relative usage of all non-dominant experts that were not

present during initial learning of a rule (second and third experts for the shape training; third expert for the color training) to represent experts not present during initial training (grey bar).

Two analyses were done to assess the networks' memory savings. In the first, we measured the number of sorting episodes required to achieve expert performance in the first sort rule (shape sort) after going through the sequential rule scenario training described above. We ran this simulation 10 times for both the standard RNN and DynaMoE network, and compared the results with the Student's *t*-test. We also used the Student's *t*-test to compare the number of sorting episodes each network required to relearn/remember the shape scenario to the number of sorting episodes required to learn the shape scenario for the first time (*Appendix*, Fig. 2.11).

In the second analysis, we measured the change in weights during retraining of both the standard RNN and the DynaMoE network on the first sort rule (shape) after sequential training. We did this for all 10 repetitions of the retraining. To visualize the movement in weight space for the retraining runs of a network, we performed PCA on the weights for all retraining runs of both networks together. We then took the top three principal components, translated the starting point in this 3 PCA weight space for all networks to the same origin point, and plotted the retraining trajectory of each repetition for each network type (Fig. 2.5*C*). For 10 independent full training and retraining runs, we also measured the Euclidean distance between the start and end points in full weight space and compared these using the Student's *t*-test (Fig. 2.5*D*). For the comparison of movement in weight space of the two networks throughout training and remembering, distances traveled in weight space for 10 runs of each type of network were measured throughout the training and remembering periods (*Appendix*, Fig. 2.10).

2.6.5 Code availability

The code used to generate the DynaMoE model used in this work is available at:

`https://github.com/tsudacode/DynaMoE`

2.6.6 Supplementary Figures

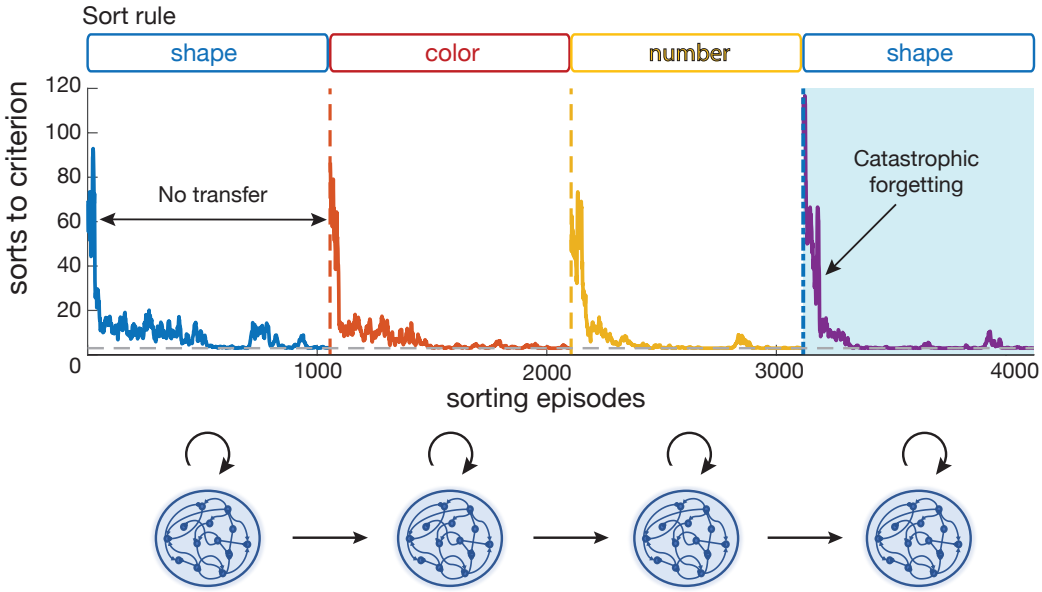


Figure 2.7. Sequential training of a standard recurrent neural network. For each task, the network was trained for 1,250 sorting episodes to display performance behavior over time (lower sorts to criterion is better performance in the WCST). During learning, all network parameters are tuned (*Bottom* schematic). When trained on a sequences of tasks, the performance trajectory of the network appears nearly identical for every new task, demonstrating the lack of transfer learning for solving new tasks. The nearly identical performance trajectory for relearning a previously learned rule—the shape rule shown in the leftmost white panel and the rightmost light blue—demonstrates the memorylessness of standard RNNs, also called catastrophic forgetting.

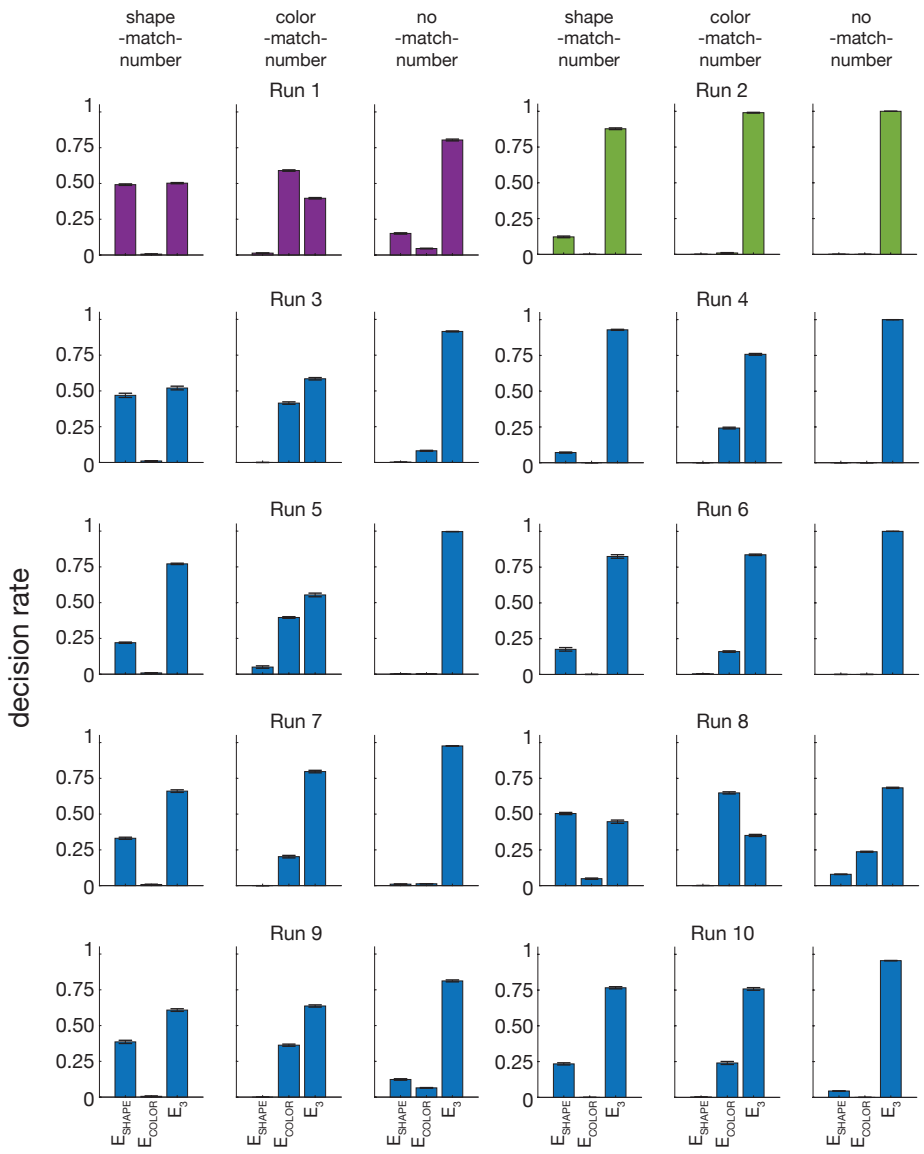


Figure 2.8. Decision rate in subsets of number sort rule after 10 independent training runs from the same seeded DynaMoE network. For each training run, the gating network and a new expert network (E_3) were simultaneously trained on the number sort rule until "expert performance" (see Methods). Weights were then fixed and 1,000 test sorts were done in the number sort rule. The proportion of decisions by the gating network to use each expert in subsets of the number rule were measured. The purple and green colored runs are the same as depicted in Fig. 2.3*E–F*. Error bars show standard deviation over 1,000 test trials. Different training runs resulted in different decision rates in subsets of the number sort rule, showing how expert performance could be achieved by different expert usage strategies.

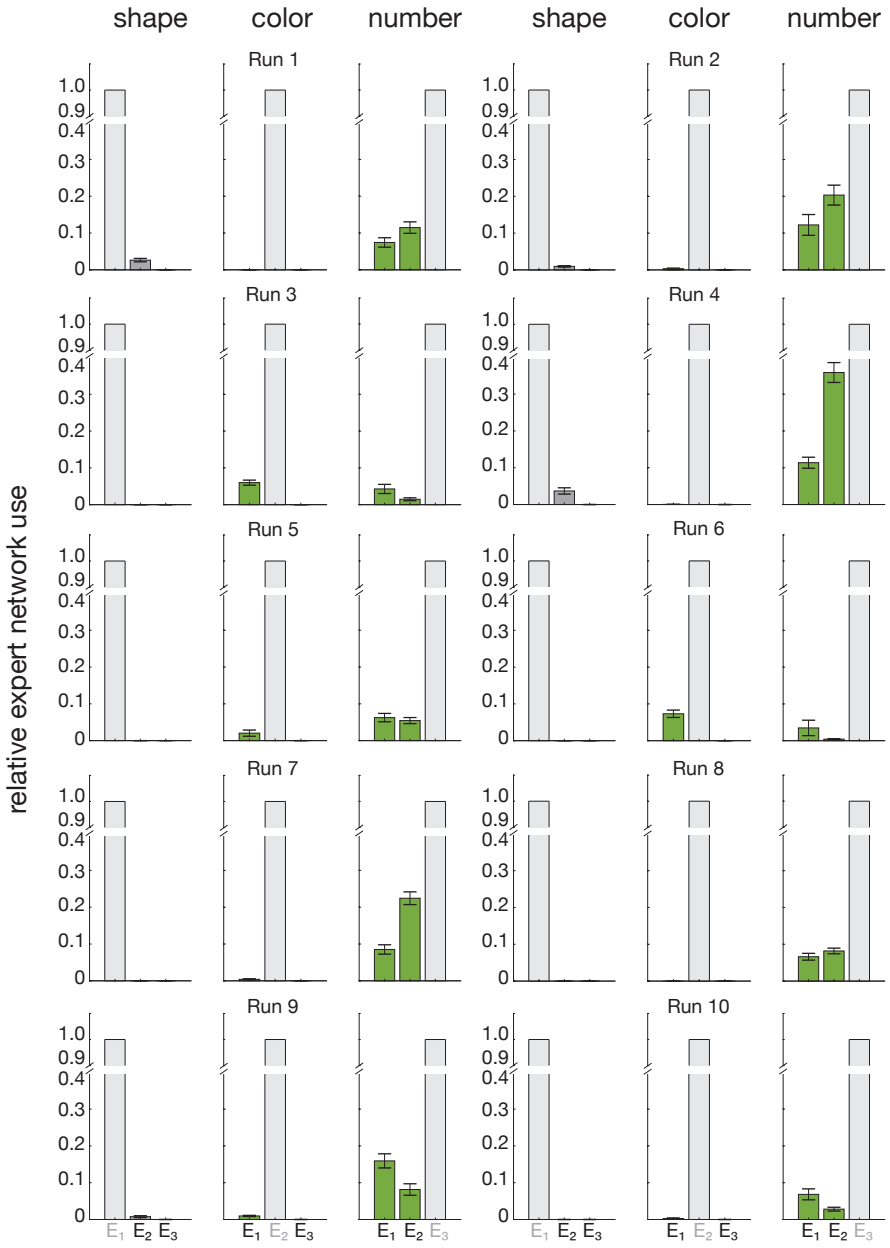


Figure 2.9. Relative expert network usage in each sort rule for 10 independently trained DynaMoE networks. For each network, after sequential training, the gating network was retuned for each sort rule until expert performance. Weights were then fixed and 1,000 test sorts were done in the same sort rule. Proportion of cards allocated to each expert were normalized to the dominant expert in that sort rule and plotted (see Methods). Color schemes and notations are the same as Fig. 2.4. Error bars show standard deviation over 1,000 test trials. Independent training runs led to variable levels of expert network usage, while clearly showing a consistent pattern of using experts added before initial training (green bars) more than those added after (grey bars) (Fig. 2.4).

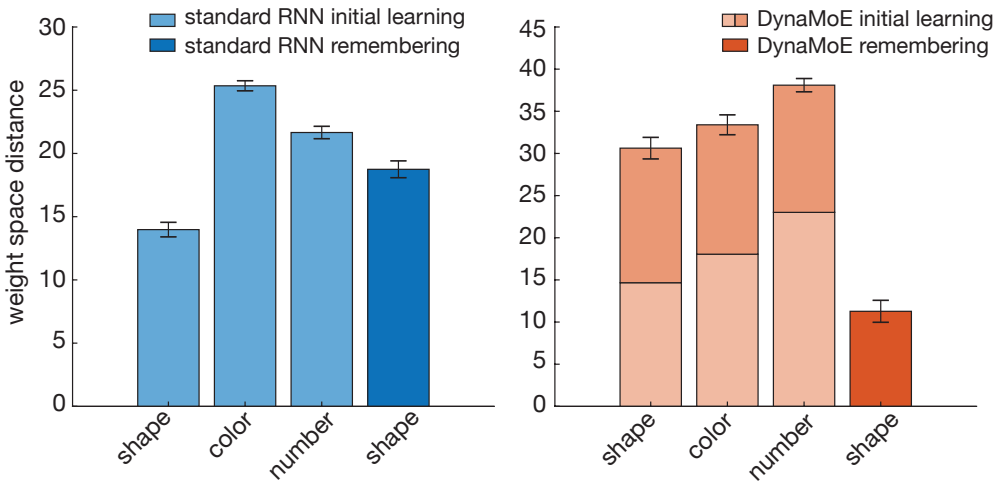


Figure 2.10. Comparison of networks during initial learning and relearning of a previously learned task in weight space. Bar graphs show movement of networks from Fig. 2.5 in weight space during sequential training of sort rules in WCST (initial learning of shape rule is leftmost bar of each bar graph). For initial learning of each rule (light blue and light and medium orange bars), networks were trained on 1,250 sorting episodes to allow full convergence to optimal performance (see Fig. 2.2 and Fig. 2.7). For relearning (dark bars), networks were trained until they reached expert performance criteria (see Methods). (*Left*), Standard RNN. Distances moved in weight space during initial learning of each rule (light blue) and "remembering" of the shape rule (dark blue) are similar. (*Right*), DynaMoE network. In stacked bars, lightest orange indicates stage 1 of training for a task (gating network only), medium orange indicates stage 2 of training (gating and new expert network). Distance moved in weight space during initial learning of each rule is similar. Distance moved during "remembering" shape rule (dark orange) is much less than initial learning of each of the rules (stacked bars). Error bars are SEM for 10 independent training runs of each type of network.

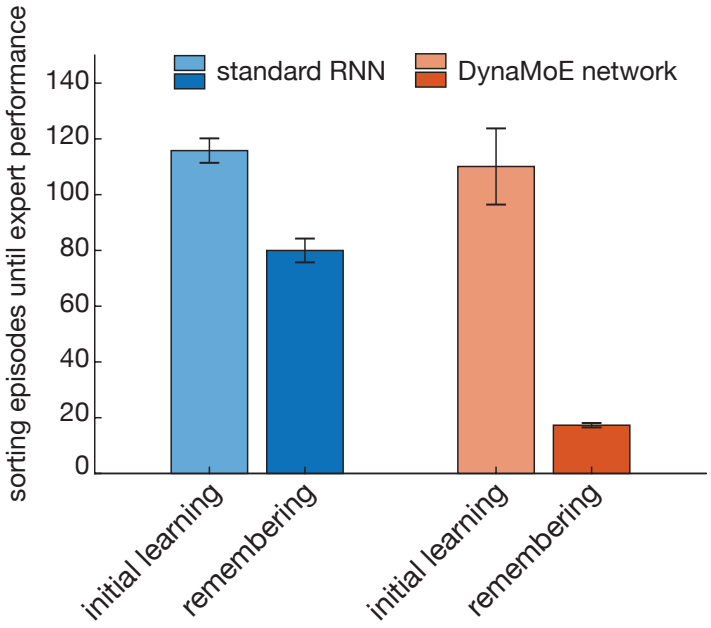


Figure 2.11. Comparison of number of sorting episodes required to attain expert performance during initial learning compared to during remembering of shape rule after sequential training. For a standard RNN, 31% fewer sorting episodes ($p = 1.54e-05$) were required when remembering compared to initially learning a rule, while for the DynaMoE network 84% fewer sorting episodes ($p = 2.37e-06$) were required for remembering compared to initially learning a rule. Number of sorting episodes required to initially learn the rule were not significantly different between the standard RNN and the DynaMoE network (light colored bars; $p = 0.70$). Error bars are SEM over 10 independent simulations for each condition.

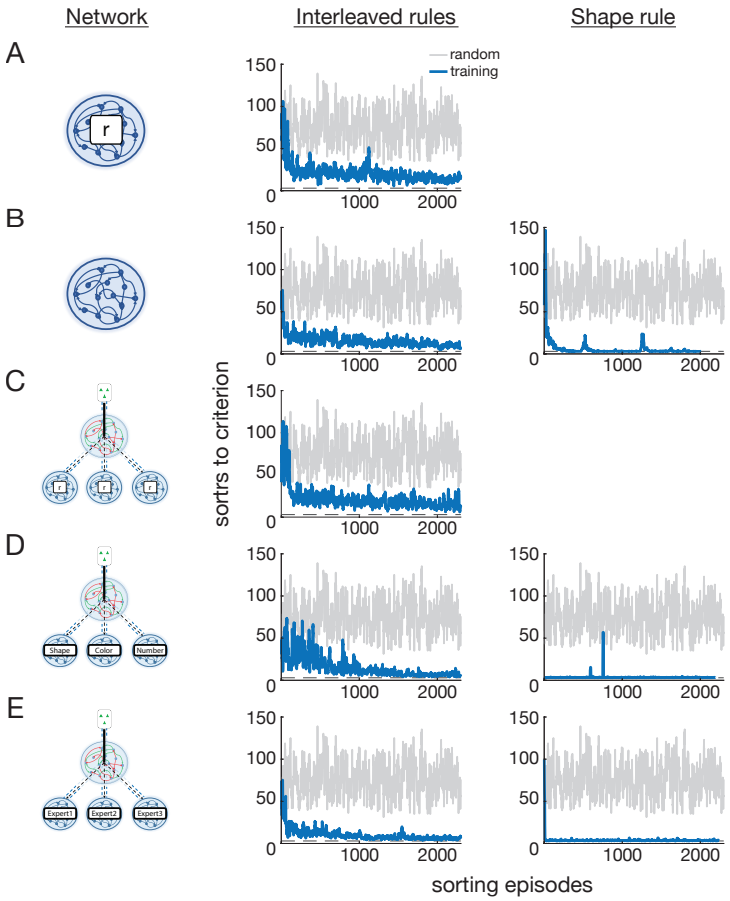


Figure 2.12. Performance of standard RNN (*A–B*), MoE (*C–D*), and DynaMoE networks (*E*) on the classic interleaved rules WCST and on a single rule after random initialization or pretraining. The first column depicts the network type, the second column shows performance on the classic interleaved rule scenario, and the third column shows performance on the shape rule for networks with pretraining. (*A*) A randomly initialized standard RNN rapidly learns the interleaved WCST (mean sorts to criterion (STC) over 10 sorting episodes (SrtEp) <50 by about 50 SrtEp of training). (*B*) A standard RNN pretrained sequentially on the shape, then color, then number rules (see Methods) learns the interleaved WCST faster (mean STC <50 by SrtEp 21) and exhibits catastrophic forgetting in the single previously experienced rule. (*C–E*) Networks from Fig. 2.1*C–E*. (*C*) MoE network with 3 randomly initialized expert networks (mean STC <50 by SrtEp 109). (*D*) MoE network seeded with pretrained rule experts (mean STC <50 from the first SrtEp). Within a single training SrtEp, this network performs near perfect on the single rule scenario. (*E*) DynaMoE network trained sequentially in a manner similar to Fig. 2.2*C* (see Methods) (mean STC <50 by SrtEp 21) converges to the best performance. It more rapidly achieves near perfect performance on the single rule scenario than the standard RNN, due to its robust memory savings. All simulations were done for 2,500 SrtEp to show convergent learning behavior. In all plots, blue traces are from networks during training and grey traces are random behavior for reference.

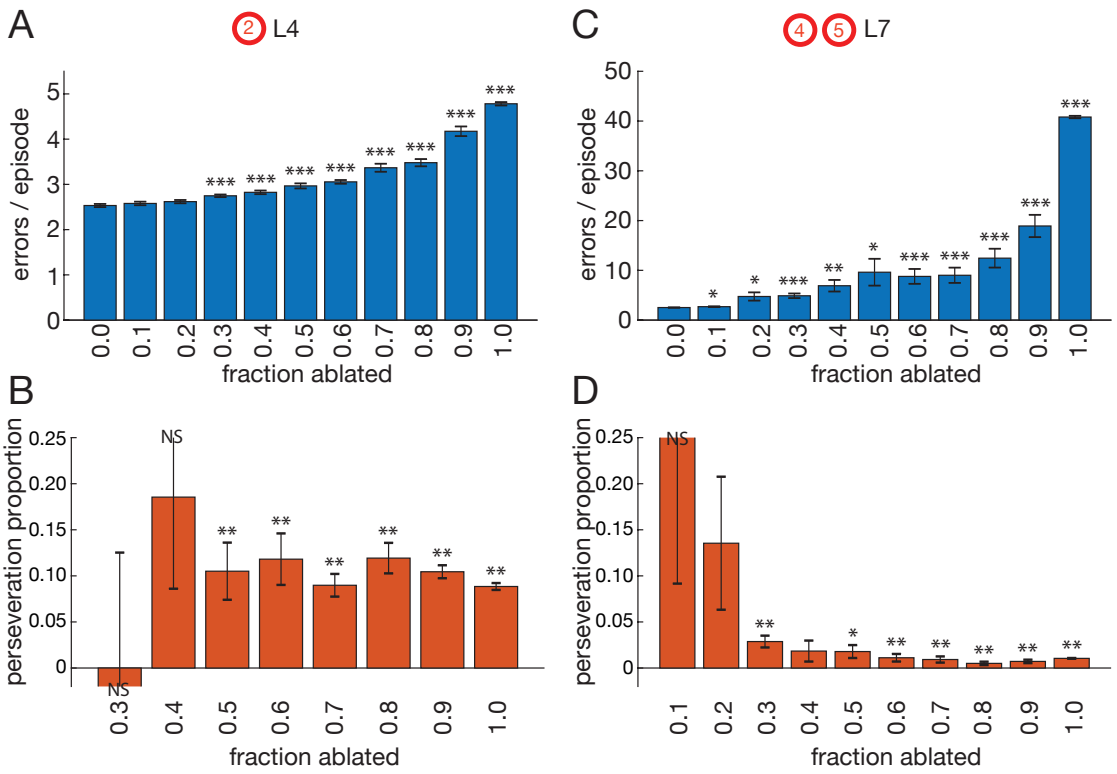


Figure 2.13. Examples of full spectrum of severity for specific lesions from Fig. 2.6. (*A–B*) L4 - lesions of the internal network dynamics (synaptic connections to the "forget" gate in the LSTM) of DynaMoE's gating network (region 2 in Fig. 2.6) (*A*) Average total errors per episode after L4 lesions of varying severity. (*B*) Average perseveration proportion for lesions that caused significant increase in average total error. (*C–D*) L7 - lesions of the synaptic connections to action and value units (region 4 and 5 in Fig. 2.6). (*C*) Average total errors per episode after L7 lesions of vary severity. (*D*) Average perseveration proportion for lesions that caused significant increase in average total error. Asterixes indicating significance follow same key as Fig. 2.6. For all plots, error bars are SEM.

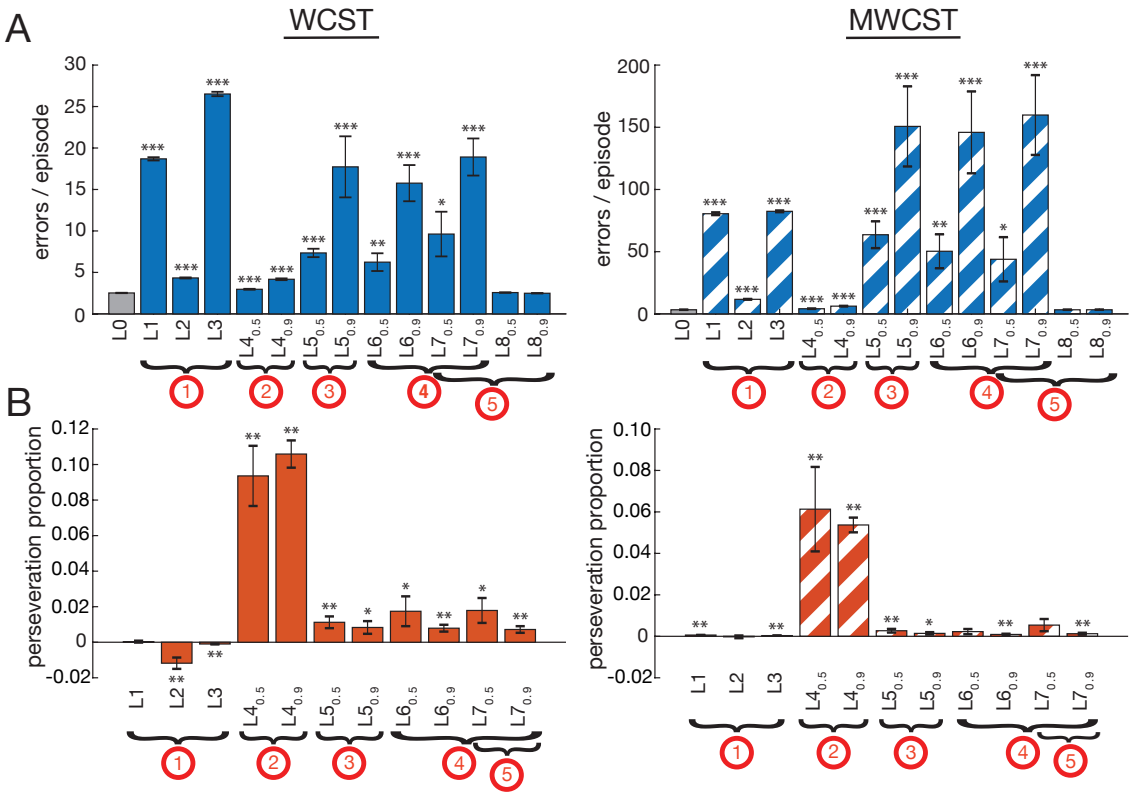


Figure 2.14. Comparison of lesions to DynaMoE's gating network on the WCST and Modified Wisconsin Card Sorting Task (MWCST). In the MWCST all ambiguous cards (which could be sorted to the same stack for 2+ different rules) are removed, leaving only the 24 unambiguous cards. (*A*) Total errors per sorting episode with the WCST (*Left*, solid blue) and MWCST (*Right*, striped blue). (*B*) Perseveration proportion with the WCST (*Left*, solid orange) and MWCST (*Right*, striped orange). The WCST error patterns are the same as in Fig. 2.6. The MWCST show a similar pattern of errors across lesion types. Asterixes indicating significance follow same key as Fig. 2.6. For all plots, error bars are SEM.

Table 2.1. DynaMoE lesion-induced impairments on the WCST. Lesions of the gating network of DynaMoE qualitatively separated into 3 categories based on error rate and perseveration proportion (Fig. 2.6): (red) lesions that caused significant increase in error rate with small or not significant perseveration proportion; (blue) lesions that caused small but significant increase in error rate with large perseveration proportion; and (grey) lesions that did not significantly change the error rate. The lower end of all ranges is 50% impairment and the high end is 90% impairment for the given lesion type. Lesions labels are same as Fig. 2.6

lesion	1 L1	1 L2	1 L3	2 L4	3 L5	4 L6	4 L7	5 L8
error increase (fold)	7.41	1.72	10.51	1.18 - 1.65	2.91 - 7.03	2.47 - 6.25	3.81 - 7.50	NS
perseveration proportion (%)	NS	-1.18	-0.10	9.36 - 10.59	0.83 - 1.12	0.79 - 1.74	0.72 - 1.79	NA

Bibliography

[Abr08] Wickliffe C. Abraham. Metaplasticity: tuning synapses and networks for plasticity. *Nature Reviews Neuroscience*, 9(387): 2008.

[AB15] William H. Alexander and Joshua W. Brown. Hierarchical Error Representation: A Computational Model of Anterior Cingulate and Dorsolateral Prefrontal Cortex. *Neural Computation*, 27: 2015, 1–57.

[ACT17] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert Gate: Lifelong Learning with a Network of Experts. *Computer Vision and Pattern Recognition Conference (CVPR)*. 2017, 7120–7129.

[Bot07] Matthew M. Botvinick. Multilevel structure in behaviour and in the brain: a model of Fuster's hierarchy. *Philosophical Transactions of the Royal Society B*, 362: 2007, 1615–1626.

[BC00] Todd S. Braver and Jonathan D. Cohen. *On the control of control: The role of dopamine in regulating prefrontal function and working memory*. Attention and Performance XVIII. London, UK: MIT Press, 2000.

[Buc+05] Bradley R. Buchsbaum, Stephanie Greer, Wei-Li Chang, and Karen Faith Berman. Meta-Analysis of Neuroimaging Studies of the Wisconsin Card-Sorting Task and Component Processes. *Human Brain Mapping*, 25: 2005, 35–45.

[Cha+19] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient Lifelong Learning with A-GEM. *International Conference on Learning Representations (ICLR)*. 2019.

[DCB15] Theresa M. Desrochers, Christopher H. Chatham, and David Badre. The Necessity of Rostrolateral Prefrontal Cortex for Higher-Level Sequential Behavior. *Neuron*, 87: 2015, 1357–1368.

[Dim+99] Mariana Dimitrov, Michael Phipps, Theodore P. Zahn, and Jordan Grafman. A Thoroughly Modern Gage. *Neurocase*, 5: 1999, 345–354.

[DL17] Mihai S. Dobre and Alex Lascarides. Combining a Mixture of Experts with Transfer Learning in Complex Games. *Association for the Advancement of Artificial Intelligence*. 2017.

[DCK14] Mael Donoso, Anne G E Collins, and Etienne Koechlin. Foundations of human reasoning in the prefrontal cortex. *Science*, 344: 2014, 1481–1486.

[EGM12] David R. Euston, Aaron J. Gruber, and Bruce L. McNaughton. The Role of Medial Prefrontal Cortex in Memory and Decision Making. *Neuron*, 76(6): 2012, 1057–1070.

[FHF91] Laura A. Flashman, Michael D. Horner, and David Freides. Note on Scoring Perseveration on the Wisconsin Card Sorting Test. *Clinical Neuropsychologist*, 5(2): 1991, 190–194.

[FB12] Michael J. Frank and David Badre. Mechanisms of Hierarchical Reinforcement Learning in Corticostriatal Circuits 1: Computational Analysis. *Cerebral Cortex*, 22: 2012, 509–526.

[FLO01] Michael J. Frank, Bryan Loughry, and Randall C. O'Reilly. Interactions between frontal cortex and basal ganglia in working memory: A computational model. *Cognitive, Affective, & Behavioral Neuroscience*, 1: 2001, 137–160.

[Fre99] Robert French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4): 1999, 128–135.

[FFW96] Jürgen Fritsch, Michael Finke, and Alex Waibel. Adaptively growing hierarchical mixtures of experts. *Advances in Neural Information Processing Systems*. 1996, 459–465.

[Fus01] Joaquin M Fuster. The Prefrontal Cortex - An Update: Time Is of the Essence. *Neuron*, 30: 2001, 319–333.

[GSL18] Michael Gimelfarb, Scott Sanner, and Chi-Guhn Lee. Reinforcement Learning with Multiple Experts: A Bayesian Model Combination Approach. *Advances in Neural Information Processing Systems 31 NeurIPS*. 2018.

[GB11] Thomas Gisiger and Mounir Boukadoum. Mechanisms gating the flow of information in the cortex: what they might look like and what their uses may be. *Frontiers in Computational Neuroscience*, 5: 2011.

[GAT19] Jan Gläscher, Ralph Adolphs, and Daniel Tranel. Model-based lesion mapping of cognitive control using the Wisconsin Card Sorting Test. *Nature Communications*, 10: 2019.

[GB48] David A Grant and Esta Berg. A behavioral analysis of degree of reinforcement and ease of shifting to new responses in a Weigl-type card-sorting problem. *Journal of Experimental Psychology*, 38(4): 1948, 404–411.

[Gre93] Kevin W. Greve. Can Perseverative Responses on the Wisconsin Card Sorting Test be Scored Accurately? *Archives of Clinical Neuropsychology*, 8: 1993, 511–517.

[HB19] Koen V. Haak and Christian F. Beckmann. Plasticity versus stability across the human cortical visual connectome. *Nature Communications*, 10: 2019.

[Hil+19] Felix Hill, Adam Santoro, David G.T. Barrett, Ari S. Morcos, and Timothy Lillicrap. Learning to Make Analogies by Contrasting Abstract Relational Structure. *International Conference on Learning Representations (ICLR)*. 2019.

[Jac+91] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive Mixture of Local Experts. *Neural Computation*, 3: 1991, 79–87.

[Joh+07] Kevin Johnston, Helen M Levin, Michael J Koval, and Stefan Everling. Top-Down Control-Signal Dynamics in Anterior Cingulate and Prefrontal Cortex Neurons following Task Switching. *Neuron*, 53: 2007, 453–462.

[KV18] Ying-Fang Kao and Ragupathy Venkatachalam. Human and Machine Learning. *Computational Economics*: 2018.

[KMG06] Mehdi Khamassi, Louis-Emmanuel Martinet, and Agnès Guillot. Combining Self-organizing Maps with Mixtures of Experts: Application to an Actor-Critic Model of Reinforcement Learning in the Basal Ganglia. *From Animals to Animats 9*. Ed. by Stefano Nolfi, Gianluca Baldassarre, Raffaele Calabretta, John C. T. Hallam, Davide Marocco, Jean-Arcady Meyer, Orazio Miglino, and Domenico Parisi. Vol. 4095. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, 394–405. ISBN: 978-3-540-38615-5.

[Kir+17] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13): 2017, 3521–3526.

[KS07] Etienne Koechlin and Christopher Summerfield. An information theoretical approach to prefrontal executive function. *TRENDS in Cognitive Sciences*, 11: 2007, 229–235.

[LCD14] Graciano Leal, Diogo Comprido, and Carlos B. Duarte. BDNF-induced Local Protein Synthesis and Synaptic Plasticity. *Neuropharmacology*, 76: 2014, 639–56.

[Lop+17] Nina Lopatina, Brian F. Sadacca, Michael A. McDannald, Clay V. Styer, Jacob F. Peterson, Joseph F. Cheer, and Geoffrey Schoenbaum. Ensembles in medial and lateral orbitofrontal cortex construct cognitive maps emphasizing different features of the behavioral landscape. *Behavioral Neuroscience*, 131(3): 2017, 201–212.

[LR17] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continuum learning. *Neural Information Processing Systems (NIPS)*. 2017.

[Mac+15] Sarah MacPherson, Sergio Sala, Simon Cox, Alessandra Girardi, and Matthew Iveson. *Handbook of Frontal Lobe Assessment*. Oxford, UK: Oxford University Press, 2015. ISBN: 9780199669523.

[MMT06] Farshad A. Mansouri, Kenji Matsumoto, and Keiji Tanaka. Prefrontal Cell Activities Related to Monkeys' Success and Failure in Adapting to Rule Changes in a Wisconsin Card Sorting Test Analog. *Journal of Neuroscience*, 26: 2006, 2745–2756.

[Man+13] Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503: 2013, 78–84.

[MT02] Eve Marder and Vatsala Thirumalai. Cellular, synaptic and network effects of neuromodulation. *Neural Networks*, 15: 2002, 479–493.

[MK08] Fiona McNab and Torkel Klingberg. Prefrontal cortex and basal ganglia control access to working memory. *Nature Neuroscience*, 11: 2008, 103–107.

[Mil63] Brenda Milner. Effects of different brain lesions on card sorting. *Archives of neurology*, 9: 1963, 100–110.

[Mit+05] Maura Mitrushina, Kyle B. Boone, Jill Razani, and Louis F. D'Elia. *Handbook of Normative Data for Neuropsychological Assessment*. New York, New York: Oxford University Press, 2005. ISBN: 0195169301.

[Mni+16] Volodymyr Mnih, Adria P Badia, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P Lillicrap, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. Ed. by JMLR. Vol. 48. New York, 2016, 1928–1937.

[Mon03] Stephen Monsell. Task switching. *TRENDS in Cognitive Sciences*, 7(3): Mar. 2003, 134–140.

[Nel76] Hazel E. Nelson. A modified card sorting test sensitive to frontal lobe defects. *Cortex*, 12: 1976, 313–324.

[Nel85] Thomas O Nelson. Ebbinghaus's Contribution to the Measurement of Retention: Savings During Relearning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 11(3): 1985, 472–479.

[OF06] Randall C. O'Reilly and Michael J. Frank. Making Working Memory Work: A Computational Model of Learning in the Prefrontal Cortex and Basal Ganglia. *Neural Computation*, 18: 2006, 283–328.

[OH12] Catherine Orr and Robert Hester. Error-related anterior cingulate cortex activity and the prediction of conscious error awareness. *Frontiers in Human Neuroscience*, 6: 2012.

[ON19] Torben Ott and Andreas Nieder. Dopamine and Cognitive Control in Prefrontal Cortex. *Trends in Cognitive Sciences*, 23: 2019, 213–234.

[Par+19] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113: 2019, 54–71.

[Pos05] Bradley R. Postle. Delay-period activity in prefrontal cortex: one function is sensory gating. *Journal of Cognitive Neuroscience*, 17: 2005, 1679–1690.

[RG01] Carl Edward Rasmussen and Zoubin Ghahramani. Infinite Mixtures of Gaussian Process Experts. *Advances in Neural Information Processing Systems 14, NIPS*. 2001, 881–888.

[RSM02] K. Richard Ridderinkhof, Mark M. Span, and Maurits W. van der Molen. Perseverative Behavior and Adaptive Control in Older Adults: Performance Monitoring, Rule Induction, and Set Shifting. *Brain and Cognition*, 49: 2002, 382–401.

[Rig+13] Mattia Rigotti, Omri Barak, Melissa R. Warden, Xiao-Jing Wang, Nathaniel D. Daw, Earl K. Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497: 2013, 585–590.

[RGH18] Rajeev V. Rikhye, Aditya Gilra, and Michael M. Halassa. Thalamic regulation of switching between cortical representations enables cognitive flexibility. *Nature Neuroscience*, 21: 2018, 1753–1763.

[RH18] Pieter R. Roelfsema and Anthony Holtmaat. Control of synaptic plasticity in deep cortical networks. *Nature Reviews Neuroscience*, 19: 2018.

[Rou+05] Nicolas P. Rougier, David C. Noelle, Todd S. Braver, Jonathan D. Cohen, and Randall C. O'Reilly. Prefrontal cortex and flexible cognitive control: Rules without symbols. *Proceedings of the National Academy of Sciences*, 102: 2005, 7338–7343.

[RM14] Peter H. Rudebeck and Elisabeth A. Murray. The orbitofrontal oracle: cortical mechanisms for the prediction and evaluation of specific behavioral outcomes. *Neuron*, 84: 2014, 1143–1156.

[Rus+16] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive Neural Networks. *arXiv:1606.04671*: 2016.

[SN96] Kazumi Saito and Ryohei Nakano. A constructive learning algorithm for an HME. *Proceedings of International Conference on Neural Networks (ICNN'96)*. Vol. 2. 1996, 1268–1273.

[Sch+16] Nicolas W. Schuck, Ming Bo Cai, Robert C. Wilson, and Yael Niv. Human Orbitofrontal Cortex Represents a Cognitive Map of State Space. *Neuron*, 91(6): 2016, 1402–1412.

[Sej+14] Terrence J Sejnowski, Howard Poizner, Gary Lynch, Sergei Gepshtein, and Ralph J Greenspan. Prospective Optimization. *Proceedings of the IEEE*. Vol. 102. 5. IEEE, May 2014, 799–811.

[Sha+17] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *5th International Conference on Learning Representations, ICLR*. 2017.

[Sic+19] Cody A. Siciliano, Habiba Noamany, Chia-Jung Chang, Alex R. Brown, Xinhong Chen, Daniel Leible, Jennifer J. Lee, Joyce Wang, Amanda N. Vernon, Caitlin M. Vander Weele, Eyal Y. Kimchi, Myriam Heiman, and Kay M. Tye. A cortical-brainstem circuit predicts and governs compulsive alcohol drinking. *Science*, 366: 2019, 1008–1012.

[Sin92a] Satinder P. Singh. The efficient learning of multiple task sequences. *Advances in Neural Information Processing Systems 4 NIPS*. 1992, 251–258.

[Sin92b] Satinder P. Singh. Transfer of learning by composing solutions of elemental sequential tasks. *Machine Learning*, 8: 1992, 323–340.

[SYW17] H Francis Song, Guangyu R Yang, and Xiao-Jing Wang. Reward-based training of recurrent neural networks for cognitive and value-based tasks. *eLife*, 6: 2017.

[SL02] Donald T Stuss and Brian Levine. Adult Clinical Neuropsychology: Lessons From Studies of the Frontal Lobes. *Annual Review of Psychology*, 53: 2002, 401–433.

[SB18] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: an Introduction*. Second. Cambridge, MA: MIT Press, 2018.

[Tye18] Kay M. Tye. Neural Circuit Motifs in Valence Processing. *Neuron*, 100: 2018, 436–452.

[VC17] Roby Velez and Jeff Clune. Diffusion-based neuromodulation can eliminate catastrophic forgetting in simple neural networks. *PLoS ONE*: 2017.

[Wan+18] Jane X Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Demis Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, 21: June 2018, 860–868.

[WR95] S. R. Waterhouse and A. J. Robinson. Pruning and growing hierachical mixtures of experts. *Fourth International Conference on Artificial Neural Networks*. 1995, 341–346.

[Wee+18] Caitlin M. Vander Weele, Cody A. Siciliano, Gillian A. Matthews, Praneeth Namburi, Ehsan M. Izadmehr, Isabella C. Espinel, Edward H. Nieh, Evelien H. S. Schut, Nancy Padilla-Coreano, Anthony Burgos-Robles, Chia-Jung Chang, Eyal Y. Kimchi, Anna Beyeler, Romy Wichmann, Craig P. Wildes, and Kay M. Tye. Dopamine enhances signal-to-noise ratio in cortical-brainstem encoding of aversive stimuli. *Nature*, 563: 2018, 397–401.

[Wes12] Jan R Wessel. Error awareness and the error-related negativity: evaluating the first decade of evidence. *Frontiers in Human Neuroscience*, 6: 2012.

[Yan+19] Guangyu Robert Yang, Madhura R Joglekar, H Francis Song, William T Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22: Feb. 2019, 297–306.

[Yao+09] Bangpeng Yao, Dirk Walther, Diane Beck, and Li Fei-fei. Hierarchical Mixture of Classification Experts Uncovers Interactions between Brain Regions. *Neural Information Processing Systems (NIPS)*. 2009.

[YWG12] Seniha Yuksel, Joseph Wilson, and Paul Gader. Twenty Years of Mixture of Experts. *IEEE Transactions on Neural Networks and Learning Systems*, 23: 2012, 1177–1193. DOI: 10.1109/TNNLS.2012.2200299.

[ZPG17] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning with intelligent synapses. *Proceedings of International Conference on Machine Learning (ICML)*. 2017.

Chapter 3

Understanding strategy differences in humans and monkeys with recurrent neural networks

This chapter investigates how a single neural circuit can come to encode a "strategy" and how can that encoded strategy can change throughout learning. To explore this, I follow an intriguing finding that drew attention to differences in strategies used by humans and monkeys even when task performance was similar [Wit+16]. In addition to calling into question commonly made extrapolations from monkey to human behaviors, the study suggested potential underlying neural circuit differences. In this chapter, I use ANNs to systematically assess what properties of a neural network lead to particular strategy adoption and show how this evolves with experience and is influenced by network capacity. My investigation reveals two surprising twists that reorient our understanding of strategy adoption and learning in neural circuits.

The work presented here is reproduced and adapted from: Ben Tsuda, Barry J. Richmond, Terrence J. Sejnowski. Understanding strategy differences in humans and monkeys with recurrent neural networks. *In preparation*, 2022.

3.1 Abstract

Animal models are used to understand principles of human biology. Within cognitive neuroscience, non-human primates (monkeys) are considered the premier model for decision-making

behaviors where direct manipulation experiments are still possible. Some prominent studies have brought to light major discrepancies between monkey and human cognition, highlighting the problematic nature of unverified extrapolation from monkey to human. Here, we use a parallel model system — artificial neural networks (ANNs) — to investigate a well-established discrepancy identified between monkeys and humans on a working memory task: monkeys appear to use a recency-based strategy to solve the same tasks for which humans use a selective strategy. We find that ANNs trained on the task exhibit a progression of strategy from random behavior (untrained) to recency-like behavior (partially trained) and finally to selective behavior (further trained), suggesting monkeys and humans may occupy different points in the same overall learning progression. To our surprise, we reveal that what appears to be recency-like behavior in the ANN, is in fact an emergent non-recency-based property of the organization of the neural network's solution space through training. We find that explicit encouragement of recency behavior has a dual effect, not only causing an accentuated recency-like behavior, but speeding up the learning process altogether, resulting in an efficient shaping mechanism to achieve the optimal strategy. Our results suggest an explanation for the discrepency observed between monkeys and humans and reveal that what can appear to be a recency-based strategy in some cases may not be recency at all.

3.2 Introduction

Animal models have been used to help understand human biology for millennia [ECF13]. Among these, studies with non-human primates (monkeys) are considered the gold-standard for studying higher order behaviors and decision in an experimental system [Fen+20; BS19; RC19; Ver+15; Lut14]. As for all animal models, non-human primate studies rely on the similarity between monkeys and humans, yet assumption of such similarities without validation can be problematic [Lee+19; Wit+16; Elm+11; PP07; Các+03]. Wittig et al. brought to light one such case, revealing major problems with inferring human behaviors from findings in non-

human primates. In a series of papers [WR14; Wit+16], they showed that while humans and monkeys both proficiently solved a number of working memory tasks, closer examination of behaviors revealed significant discrepancies in problem solving strategies, only apparent through measurement of error patterns. In particular, for several tasks monkeys used what appeared to be a recency-based strategy while humans used a target-selective strategy.

The results from Wittig et al. suggested an alternative strategy used by monkeys, though the underlying neural computations were left unknown. Wittig et al. speculated that the difference may originate from use of language, understanding of the task rules, or superior working memory capacity and selectivity [Wit+16].

Artificial neural networks (ANNs) are a parallel model system particularly suited to investigating the latter two possibilities: strategy differences resulting from task learning or neural network capacity. ANNs are powerful models to understand neural computations due to their ability to capture important aspects of neural population dynamics while providing full access to all components of the network for analysis. For these reasons, ANNs have been established as another useful model system and have been successfully used numerous times to reveal the network computations underlying observed neurocognitive phenomena [Man+13; Wan+18; Rus+18; KS20].

Using recurrent neural networks (RNNs), we show that the most apparent explanation is once again likely not the full story, peeling back another layer of the onion. We show that RNNs trained on a working memory task can exhibit human-like or monkey-like strategy behaviors, and in fact progress through these strategies with increasing experience. A deeper dive into the mechanism underlying the RNNs' strategy progression reveals that what appears to be a recency-based strategy is not recency-related, but rather stems from the evolution of the RNN's solution space, which, depending on the RNN's experience, can mimic recency-like or target-selective behavior. We then consider the role of a true recency-based strategy on task learning and performance.

3.3 Materials and methods

3.3.1 Working memory tasks from Wittig et al. 2016

Our analyses were done with the working memory tasks used by Wittig et al. (see [Wit+16; WR14] where a full description can be found). In brief, there were three tasks: Same-Different, in which a response indicates if the second cue matches the first cue; Match-First, in which a response indicates if the last cue matches the first cue in a cue sequence; and Match-Any, in which a response indicates if the last cue matches any of the previous cues in the current trial. Each task had variations depending on the set size (number of different cues that could be selected) and trial size (number of steps in a single trial), leading to a total of 8 task variations. We computationally implemented each of these task variations and trained and tested all network configurations on each task.

Cues were encoded as a single value ranging from [1,maximum cue] (e.g. for Match-First, set size 5, this was [1..5]). For timesteps during which no cue was presented, [0] was presented in place of any cue. Trials were separated by a timestep with no cue presented.

Rewards were given after the RNN's decision of "match" or "no match" following presentation of the final cue in a trial. Correct trials were rewarded with +5 and incorrect trials with -5. At all other timepoints reward feedback was 0.

In the recency-based reward scheme for Match-First, two modifications were made to encourage recency-based behavior. X-A-X-A trials in which the RNN responded "match" were rewarded +3 and X-X-A-A trials in which the RNN responded "match" were rewarded +1 (in original reward scheme reward for both these would be -5). All other rewards remained unchanged.

3.3.2 Recurrent neural networks models

Our RNN models consisted of LSTMs of different sizes ($n \in \{5, 10, 25, 50, 100, 200, 500\}$ neural units). Unlike "vanilla" RNNs, LSTMs incorporate multiple intrinsic gates that influence

the RNN's state dynamics with respect to retaining information, incorporating new information, and outputting information. This gating structure enables robust learning and behaviors mimicking animal and human models on multiple tasks [Wan+18; Tsu+20]. The LSTM states and gates are described by the following equations:

$$
\begin{aligned}
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
o_t &= \sigma(W_{ox}x_t + W_{ho}h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
h_t &= o_t \circ tanh(c_t)
\end{aligned}
$$

where f_t, i_t,and o_t are the forget, input, and output gates at time t respectively, σ is the sigmoid activation function, W_{ij} denotes the weights from component i to component j, x_t is the external input at time t, h_t is the ouput of the LSTM at time t, c_t is the state of the LSTM cell at time t, b_f, b_i, and b_o are the biases of the forget, input and output gates respectively, b_c is the bias of the cell states, and $\circ$ denotes the Hadamard product.

3.3.3 Reinforcement learning

We trained our RNNs by reinforcement learning, similar to how humans and animals learn many tasks such as the working memory tasks from Wittig et al. 2016. Similar to [Tsu+20], we used an Advantage Actor-Critic algorithm (see Mnih et al. [Mni+16] for a full description of the algorithm). In brief, the objective function consists of the gradient of a policy term, an advantage value term, and an entropy regularization term:

$$
\begin{aligned}
\nabla\mathscr{L} &= \nabla\mathscr{L}_\pi + \nabla\mathscr{L}_v + \nabla\mathscr{L}_H \\
&= \frac{\partial log\pi(a_t|s_t;\theta)}{\partial\theta}\delta_t(s_t;\theta) + \beta_v\delta_t(s_t;\theta)\frac{\partial V}{\partial\theta} + \beta_H\frac{\partial H(\pi(a_t|s_t;\theta))}{\partial\theta}
\end{aligned}
$$

where π is the policy, a_t is the action taken at time t, s_t is the state at time t, θ is the network parameters, β_v, β_H are hyperparameters for scaling the contribution of the value and entropy terms respectively (we set $\beta_v = \beta_H = 0.05$ for all our simulations), V is the value output of the network, and H is the entropy regularization term of the policy. δ_t is the advantage estimate, which represents the temporal difference error:

$$\delta_t(s_t;\theta) = R_t - V(s_t;\theta)$$

where R_t is the discounted reward:

$$R_t = \sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(s_{t+k};\theta)$$

where k is the number of steps until the next end state. When $\gamma = 0$, $R_t = r_t$.

The advantage equation in this case is equivalent to a temporal-difference error signal enabling temporal difference reinforcement learning.

The parameters of the model were updated during training by gradient descent and back propagation through time using AdamOptimizer with learning rate $1e-03$. Training occurred after every 3 trials. For all tasks, we set $\gamma = 0$ since actions at time points prior to the decision timepoint had no bearing on outcome.

Similar to the implementation of others [Wan+18; Tsu+20], input to networks at each time point was given as vector with the current cue, the action taken at the previous time point, a_{t-1}, and the reward received at the previous timepoint, r_{t-1}.

RNNs were trained for 50,000 trials (n_{ep}) and tested throughout after $n_{ep} \in \{0, 5, 10, 20, 30, 40, 50\}$ x 10^3 trials. To compare 25 unit RNNs on Match-First, set size 5 with the original and recency-based reward schemes, we trained RNNs with either reward scheme and tested every 5 x 10^3 trials. To compare RNN activity solution spaces across reward schemes we selected training

points with similar performance (and close to that of monkeys on the task) which was 20k for the original (77.6% correct) and 12k for the recency-based reward scheme (72.4% correct).

3.3.4 Serial position curves

Following Wittig et al. 2016 we measured serial position curves for RNNs. Serial position curves give the fraction of yes (i.e., "match") responses when a cue at a given position matches the target cue (last cue in a trial). For example, for Match-First, trial size 4, if the target cue was cue 1, the serial position curve would give the fraction of yes/no responses when cue 1 was in the first position (-3 test recency position in Figure 3 of Wittig et al. 2016; yes response in this case was a correct match), fraction of yes/no responses when cue 1 was in the second position (-2 test recency position), and same for third position (-1 test recency position). These fractions (e.g. -2 and -3 test recency positions for Match-First, trial size 4) then can be fit with a least-squares linear fit as was done in Figure 3 of Wittig et al. 2016 [Wit+16]. The linear fit can be used to calculate a residual that describes the deviation of the observed "hit rate" (yes/no fraction at -3 test rececny position) from the linear fit; this is what we refer to as linfit_res. The slope of the linear fit indicates the presence or absence of a recency-like behavior; this is what we refer to as the linear fit slope.

3.3.5 Sorting directions

Sorting directions in Fig. 3.5 were computed by calculating the centroid of each point cloud in the space of the first three principal components identified by the cue presented at a given timestep in the trial. A least-squares error line was fit to these centroids to give the sorting direction. This was calculated for network states after being presented with the first 3 cues in the sequence for Match-First on all correct trials from 5000 test trials at each level of training (50k training episodes for Fig. 3.5A, and 20k for Fig. 3.5B). The first sorting direction was calculated with respect to the first cue presented (different colors in Fig. 3.5A,B leftmost colored PC plots). The second and third sorting directions were calculated using the point cloud in which the first

cue was cue 1 (i.e., within the blue point cloud in the leftmost colored PC plots in Fig. 3.5).

3.4 Results

3.4.1 Modeling the contribution of network size and experience

We investigated the contribution of network size (i.e., capacity) and training experience (i.e., task understanding) using the same working memory tasks introduced by Wittig et al. 2016 [Wit+16](see *Materials and Methods*) to reveal strategy differences between monkeys and humans (Fig. 3.1A). ANNs allow systematic characterization of network behavior as a function of both network size and amount of experience, i.e. training. We varied RNNs across a size spectrum ($n \in \{5, 10, 25, 50, 100, 200, 500\}$ neural units) and training experience spectrum ($n_{ep} \in \{0, 5, 10, 20, 30, 40, 50\}$ x 10^3 trial episodes) (Fig. 3.1B).

We found that for some tasks (e.g. Same-Different) networks across the size spectrum tested could gain proficiency, though different sized networks required different amounts of training. Across all tasks, smaller networks generally required more experience to gain comparable proficiency (Supp. Fig. 3.7). Following the metrics used by Wittig et al., we measured "serial position curves" [Wit+16] for networks with each size and experience combination. To assess the degree of target-selectivity of each network configuration, we measured the linear fit residual (linfit_res), following the linear fit protocol of Wittig et al. (see *Materials and Methods*). We found that the linfit_res increased more quickly with experience for larger networks, correlating very tightly with performance (Fig. 3.1C,D and *Appendix*, Fig. 3.8).

3.4.2 RNNs progress from random to recency to selective strategy with experience

Serial position curve linear fit residuals indicate the degree of target selectivity, but they cannot distinguish between different forms of non-selectivity. To illustrate this point, consider two scenarios: 1) a network begins untrained with random behavior, gradually increases selectivity with no other response trends, and finally achieves high target-selectivity (*Appendix*,

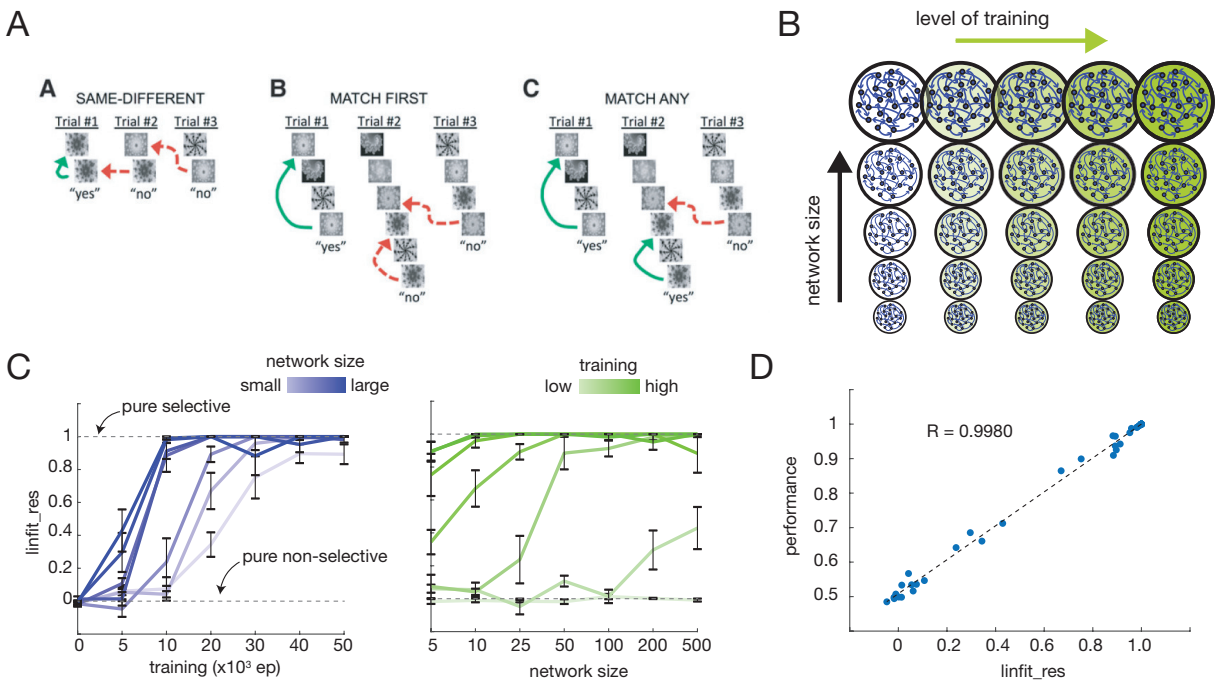


Figure 3.1. Task performance and target-selectivity increases as a function of both network size and training experience. (A) Working memory tasks from Wittig et al. 2016. Reprinted from [Wit+16]. (B) Ashby chart schematic of modeling strategy to characterize contribution of network size and training experience to task performance and strategy. (C) Linear fit residuals of serial position curves across network sizes and training experience, here shown for Same-Different (set size 2). Linfit_res mirrors performance progression, with larger networks' linfit_res increasing more rapidly with increasing training experience. (D) Linfit_res and performance are tightly correlated across network configurations.

Fig. 3.9A); 2) a network begins untrained, adopts a recency-like response profile, then achieves high target-selectivity (Fig. 3.2A and *Appendix*, Fig. 3.9B). In both these cases, the linear fit residual is small while the network has a random behavior or a recency-based behavior and may follow very similar linear fit residual progressions. To distinguish between these possibilities, we introduced another measure of serial position curves: linear fit slope.

Linear fit slope distinguishes between the two scenarios above: in the first (random $\rightarrow$ low target-selective $\rightarrow$ high target-selective), linear fit slope will remain around zero (*Appendix*, Fig. 3.9A), whereas in the second (random $\rightarrow$ recency-based $\rightarrow$ high target-selective), linear fit slope will follow a predictable deflection during recency-based behavior (Fig. 3.2A and *Appendix*, Fig. 3.9B). Importantly, the predicted recency-based behavior provides a strong prior

on the direction of the slope deflection, decreasing the probability of erroneously attributing a spurious slope deflection to a strategy difference.

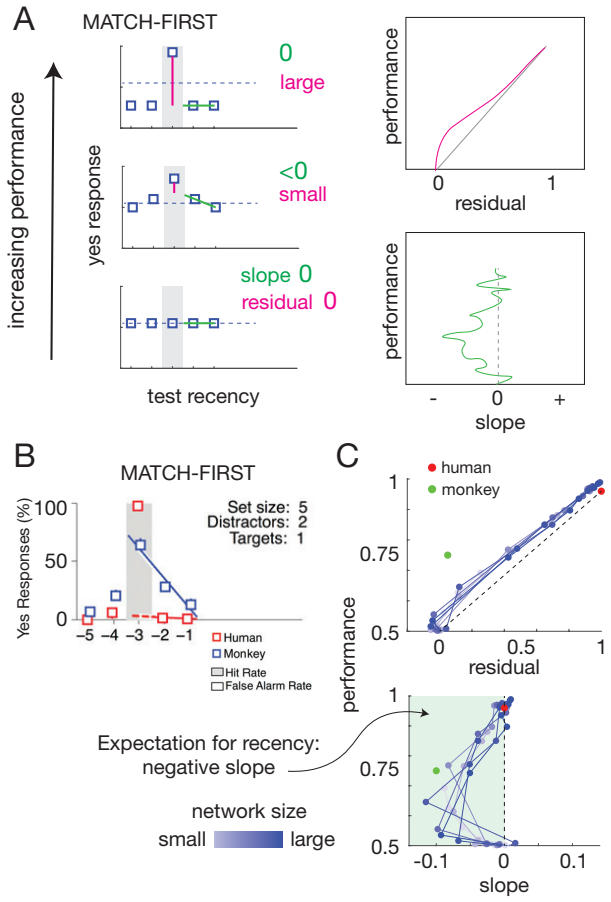


Figure 3.2. RNNs progress from random to recency-like to target-selective behavior over training. (A) Serial position curve exhibits specific characteristics depending on strategy. Left panels, Random behavior would be characterized by a near-zero linear fit residual and slope; recency-based strategy would continue to have a small residual, but a slope of a particular direction — for Match-First, a negative slope; target-selective strategy would have a large residual and near-zero slope. Right panels, Progression through these strategies would be characterized by the residual and slope vs performance schematic plots. (B) Serial position curve with linear fits from Wittig et al. 2016. Humans exhibit a target-selective behavior with large residual and near-zero slope. Monkeys exhibit a recency-based behavior with small residual and expected negative slope. Reprinted and adapted from [Wit+16]. (C) RNNs across the size spectrum (small to large is light to dark blue) exhibited a progression from random (zero residual, zero slope) to recency-like (small residual, negative slope) to target-selective behavior (high residual, zero slope). Points representing human and monkey residuals and slopes are shown in red and green, respectively.

Using both measures — linear fit residual and slope, — we assessed the networks on the

working memory tasks. For some tasks, a clear progression emerged, following that predicted by the random → recency-based → high target-selectivity scenario (Fig. 3.2C), while other tasks appeared to follow the alternative progression (random → low target-selective → high target-selective) (Supp. Fig. 3.10). Furthermore, in some cases networks across all sizes tested exhibited the recency-based progression (Fig. 3.2C), while in others there appeared to be a size dependency, with smaller networks more likely to progress through recency-based behavior (Supp. Fig. 3.10). Three possibilities may underlie these discrepencies. First, different network configurations may truly follow different behavioral progressions. Second, there appeared to be a trend where task scenarios involving larger cue deck sizes (100 cues vs 5 cues) showed less recency-based progression. A simple reason for this may be related to sample size. Serial position curves rely on occurrences of "yes" responses at each position for each cue. We found evidence of such undersampling, revealing similar recency-based progression when sample size was increased proportional to cue deck size (*Appendix*, Fig. 3.11A). Third, in some cases networks — particularly larger networks — may progress through different strategies rapidly during training, such that if the sampling rate during training is too coarse, deflections of the residual or slope curves will be missed. We found evidence of this form of undersampling as well (*Appendix*, Fig. 3.11B).

While we had hypothesized that network size (capacity) and experience (learning) may contribute to strategy adoption and progression, it was nevertheless surprising that networks appeared to progress through a recency-based behavior given no recency-based reward scheme was used in the reinforcement learning structure of the tasks (see *Materials and Methods*). To investigate why the networks appeared to transition through a recency-based intermediate strategy, we considered a network configuration that exhibited a clear progression from random to recency-based to high target-selectivity, approaching intersection with the data points from human and monkey from Wittig et al. (red and green points in Fig. 3.2C, respectively): the 25 neural unit RNN trained on the Match-First task with small cue deck size (5 cues) and 4-step trial size (2 distractors) (Fig. 3.2B,C), which we focus on for the remaining analyses performed.

In this task, trials take on the form A-B-C-D where a Match occurs when D and A are the same cue and no-match occurs otherwise (Fig. 3.3A). Recency-based behavior was characterized by a higher X-A-X-A error "yes" response rate than X-X-A-A (where X is any cue other than A; this relative difference causes the negative slope deflection in Fig. 3.2C).

3.4.3 Recency or something else?

At intermediate training experience, RNNs' serial position curves exhibited the same residual and slope profiles characteristic of monkeys in Wittig et al. 2016. Our RNN models allowed systematic interrogation of the network behaviors, specifically the hypothesis that this behavior is reflective of a recency-based strategy; a hypothesis that has concrete predictions which can be validated or refuted in our model. Like monkeys, RNNs emergently exhibit this behavior, without an explicit recency-based reward scheme. A straightforward measure of whether a true recency-based strategy is being used is comparison of different cues sequences in which lead cue is varied. A recency-based strategy not only predicts more X-A-X-A match errors than X-X-A-A in Match-First, but also predicts that all X-A-X-A errors are equally likely, regardless of the other cues' (X) identities; 2-**1**-3-**1** and 5-**1**-3-**1** match responses should equally contribute to recency-based errors, where **1**-X-X-**1** is the correct match, and numbers indicate specific cues (i.e., "1" refers to cue 1).

To assess this, we measured the cue-specific match error responses of RNNs exhibiting "recency-strategy" behavior. A recency-based strategy would predict a flat rate of match error responses across all lead cue identities. For example, 2-1-X-1, 3-1-X-1, 4-1-X-1, and 5-1-X-1 would each make up 25% of X-1-X-1 match errors. Interestingly, this was not the case. Lead cue-specific responses varied strongly, ranging from 89% to 0%. This refutes a simple recency-based strategy. Furthermore, match error rates exhibited a ramping progression around the corresponding mistaken match cue (Fig. 3.3), suggesting an underlying lead cue-specific structure to the match error rates. This was true in both the higher (X-A-X-A) and the lower (X-X-A-A) match error categories (Fig. 3.3 dark and light blue bars, respectively). Although

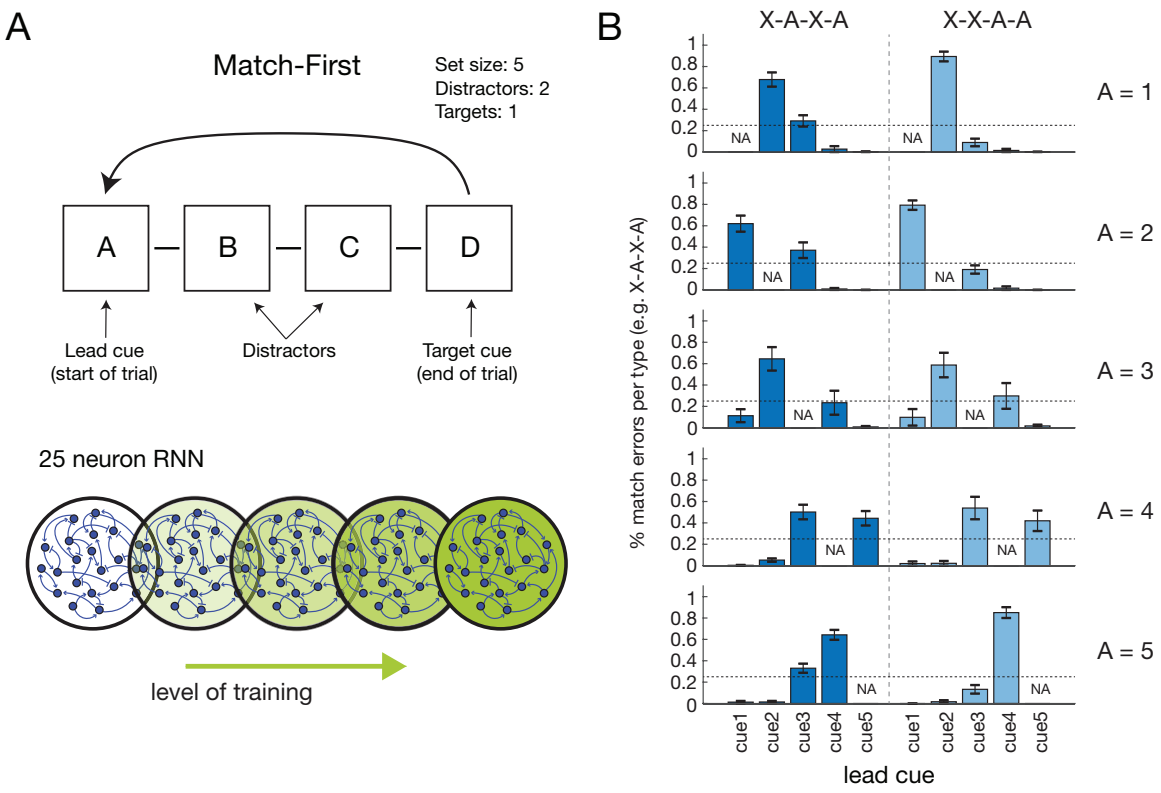


Figure 3.3. Stimulus-specific breakdown of match errors suggests not mere recency. (A) In a Match-First trial, a sequence of cues is presented and the agent must respond "yes" or "no" match by comparing the lead (cue A) and target (cue D) cues in the sequence. For a set size of 5, there are 5 unique cues which we refer to as cue1 through cue5 (1–5, for short). We focus our analysis on the 25 neuron RNN which exhibited behavioral variation mimicking monkeys and humans over training on Match-First. (B) In the cue sequence labels, the number/letter represents the cue identity, A represents the cue of interest, and X represents any cue other than A. E.g. in the top row A=1; dark blue bars indicate X-1-X-1 cue sequences (2-1-X-1, 3-1-X-1, 4-1-X-1, 5-1-X-1 from left to right) and light blue bars indicate X-X-1-1 cue sequences (2-X-1-1, 3-X-1-1, etc). A identities for each row is given on the right. For RNN trained on 20k trials and exhibiting recency behavior, both dark and light bars show uneven distribution of match errors across different lead cues. This refutes a simple recency-based strategy which would predict parity across all lead cues, i.e., 25% (horizontal dotted line in each graph) for all dark blue bars and 25% for all light blue bars. Error bars are SEM.

course-grain analysis suggested a recency-based strategy, the strong variation in response rate by lead cue identity reflects an underlying non-recency behavior derived from the formation and refinement of the RNNs' solution spaces.

3.4.4 Formation of the RNN solution space with experience

To understand the RNN's solution space, we first considered the population activity of the network over the course of individual trials. Untrained RNNs' activity trajectories showed no discernible pattern based on cue order. In contrast, after reaching target-selective performance (50k training episodes), activity trajectories suggested a mechanism of encoding cues. The relevant cues (first and last) appeared to cause a directionally similar deflection in 3-PC space, while the two distractor cues caused a deflection along a different and roughly consistent direction. This suggested the possibility that RNNs learn to ignore the distractor cues by controlling activity flow in an independent direction from the activity flow of the relevant cues. Visualization of the activity trajectories during recency-like behavior (20k training episodes) showed a pattern more organized than that of untrained RNNs, but not achieving the apparent directional independence for distractor cues seen after 50k training episodes (*Appendix*, Fig. 3.12).

To better understand the structure of the RNN's solution space, we considered the final activity state of the RNN after being presented with all four cues (first cue, 2 distractors, target cue). We found a precise structure that emerges through training. Untrained RNNs adopt random behavior, for example always responding "match" leading to final state activity reflective only of stimulus-dependent effects (i.e., similarity-based influence on network state, Fig. 3.4A and B, 0k). After some training when the recency-like behavior emerges (20k training episodes), activity states begin to show clustering reflective of "match" and "no match" regions (Fig. 3.4A, 20k) suggesting some organization by target cue and first cue (Fig. 3.4B, 20k). Finally, when target-selectivity is achieved (50k training episodes) a highly precise solution space map underlies network behavior, with tightly defined activity spaces demarcating particular match and no match combinations (1-X-X-1 vs 2-X-X-1 vs 2-X-X-2, etc.) that follows a clear organization based on first and target cues (colored and shaded point clouds in Fig. 3.4A,B, 50k).

Further dissection of the RNN activity solution space revealed that the arrangement of these network activity point clouds depends on cue-encoding, i.e., the final network state for

1-X-X-1 is closer to 2-1-X-1 than to 5-1-X-1 (blue, light red, and dark red points in Fig. 3.4C, respectively), something that would not be predicted by a recency-based strategy. This also occurred for X-X-1-1 activity states (light vs dark green points in Fig. 3.4C), as well as for the other target cues (Fig. 3.4D shows for target cue 3). Less separated activity states after 20k training episodes resulted in more errors than after 50k training episodes, though the same structure remained (Fig. 3.4D, 20k vs 50k). This closer activity state proximity (resulting in higher activity region overlap and consequent erroneous network readout) of particular cue sequences explains why the proportion of X-1-X-1 match errors was much higher for 2-1-X-1 (68%) than for 5-1-X-1 (0.3%) (Fig. 3.3, dark blue bars, first row), and were not close to the 25% parity predicted by a recency-based strategy.

3.4.5 Alignment of activity space sorting directions underlies recency-like phenotype

Although this insight into solution space structure explained the lead cue-mediated inequality of match-error response profile, it did not help explain why networks consistently exhibited a recency-like response behavior. Given that final network state was organized by cues that were presented, we investigated how sequential cue presentation in a trial was encoded by the network. To do this, we considered the state of each network after being presented with the first 3 cues (lead cue and 2 distractor cues) and assessed whether network activity could be separated by cue history.

We found that indeed network state was neatly organized by cue history, with each successive cue causing a directional separation in activity space. We characterized this "sorting direction" for each cue (see *Materials and Methods*) and found that in the space spanned by the first three PCs (accounting for $>$88% of variance in activity), the angles of the sorting directions approached orthogonality (for an example network: $\theta_{12} = 81^\circ$ between sorting directions 1 and 2; $\theta_{23} = 62^\circ$ between directions 2 and 3, and $\theta_{13} = 112^\circ$ between directions 1 and 3; Fig. 3.5A).

The same network with less experience had sorting angles less close to orthogonality

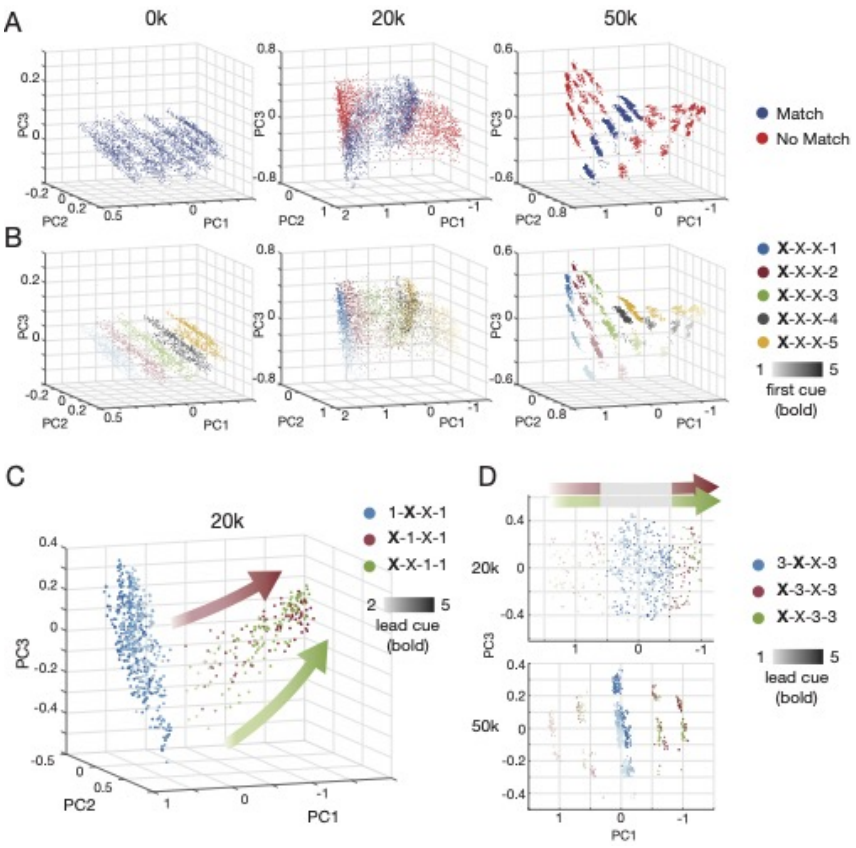


Figure 3.4. RNN solution space geometry explains uneven match error profiles. (A) PCA of network activity after presentation of all cues in a trial for trials answered correctly, colored by match (blue) or no match (red) identity. The untrained RNN (0k) outputs random behavior (e.g. always "match"; so correct trials are the match trials). At 20k training episodes, distinct match and no match clusters begin to form. At 50k, clearly demarcated clusters in activity space are apparent, defining a "match" band (blue region) within a surrounding "no match" zone (red region). (B) Same plots as (A) but colored by last cue (target cue) and shaded by first cue in sequence. Color and shading shows clear organization at 50k; each cluster corresponds to a unique lead and last cue combination. Emergence of this conformation can begin to be seen at 20k. (C) PCA of solution space for match error trials with target cue 1 shows arrangement that leads to mostly likely error overlaps with cue sequences with lower lead cues (lighter shaded red and green). (D) Neighboring match error trials for target cue 3 sequences shows similar result, with most likely overlap for cues starting with cue 2 or 4. For comparisons in (A),(B) PCA was performed on all correct trials (51% for 0k, 83% for 20k, 97% for 50k). For (C),(D) PCA was performed on all relevant trials (e.g. for (C), all 1-X-X-1, X-1-X-1, X-X-1-1 trials). Top 3 PCs explained $>$87% of activity variance for all analyses. In (D), PCs 1 and 3 explained 75% and 83% of variance for 20k and 50k trained RNNs, respectively.

(for the same network after 20k training: $\theta_{12} = 61°$, $\theta_{23} = 76°$, and $\theta_{13} = 130°$; Fig. 3.5B). In particular, we observed that the first sorting angle, θ_{12}, consistently increased from an acute

angle toward orthogonality during training progression from the recency-like behavior to the target-selective behavior (Fig. 3.5C). This progression of θ_{12} explains the recency-like response behavior of the networks.

To illustrate this, first consider the extreme case where sorting directions 1 and 2 are the same, i.e., $\theta_{12} = 0°$. In this case, as illustrated in Supp. Fig. 3.13A, any overlap between points from the cloud defined by first cue 2 with the cloud defined by first cue 1 (large red and blue blobs in Supp. Fig. 3.13A, respectively) would more likely be 2-1-X-X than 2-Z-X-X (where Z is any cue other than cue 1 or 2; in this case cue 3, 4, or 5). Thus, between these two points clouds, 2-1-X-1 is more likely to be mistaken for 1-X-X-1 than 2-X-1-1, which is exactly the recency-like phenotype.

This explanation makes a strong prediction based on the organization of the remaining point clouds (with first cue 3, 4, 5), namely that a similar recency-like phenotype will occur between 4-5-X-5, 4-X-5-5, and 5-X-X-5, but not so for the other cue combinations (e.g. 2-3-X-3, 2-X-3-3, and 3-X-X-3) (see *Appendix*, Fig. 3.13A). We found that this prediction was highly accurate. While altogether, a recency-like phenotype was apparent (Fig. 3.5D; $p = 8.3828\text{e}{-5}$, one-sample Student's *t*-test), when stratified by target cue (which measures errors made in confusion with 1-X-X-1, 2-X-X-2, 3-X-X-3, etc.) a strong recency-like phenotype was seen with target cues 1 and 5 ($p = 0.0020$ and $p = 0.0046$, respectively; Student's *t*-test), but not with target cues 2, 3, 4 ($p = 0.7929$ and $p = 0.2833$ $p = 0.3312$, respectively; Student's *t*-test; Fig. 3.5E), exactly as predicted by a small $\theta_{12} \approx 0°$.

Further training led to increased cluster separation and also increased $\theta_{12} >> 0°$ toward orthogonality, which decreases likelihood of the recency-like phenotype (*Appendix*, Fig. 3.13B).

3.4.6 Using recency to learn

Given our finding that RNNs exhibit the recency-like phenotype but not due to a recency-based strategy, we investigated how explicit encouragement of a recency-based strategy would alter RNN behavior, specifically if it would alter the RNNs' solution mechanism. To do this, we

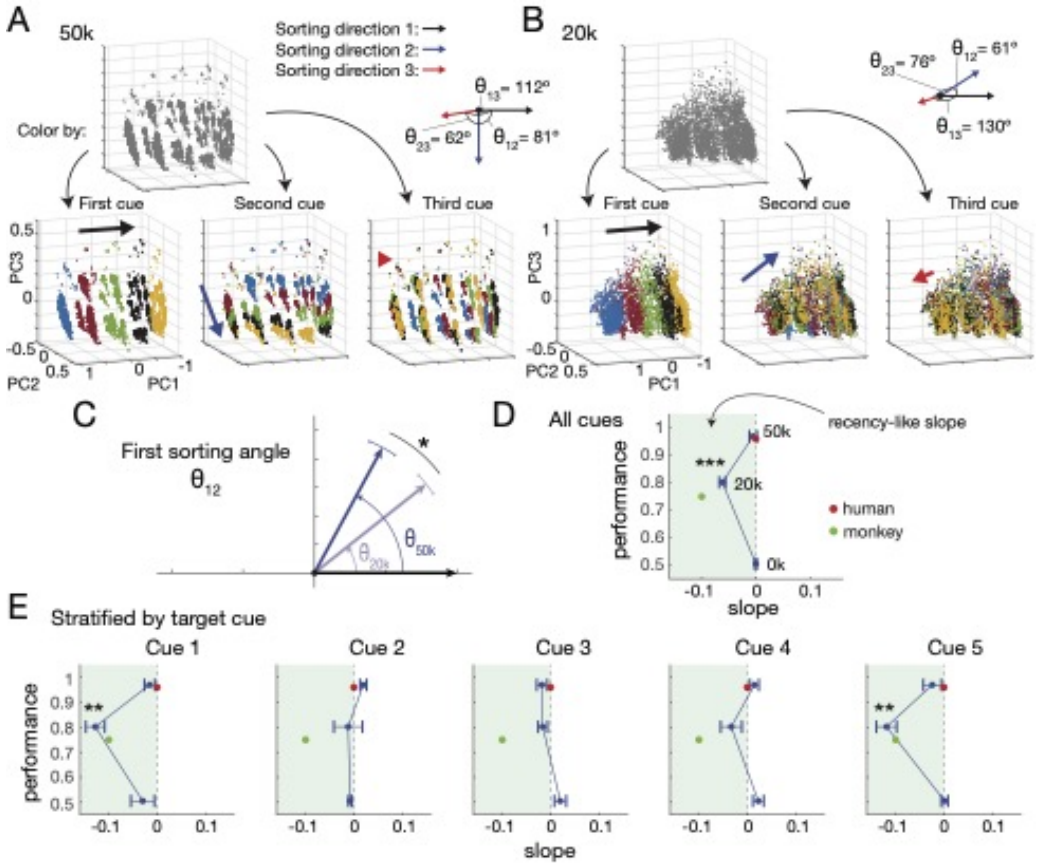


Figure 3.5. Cue-dependent sorting directions of activity solution space underlie recency-like phenotype. (A) Activity states of an RNN trained on 50k episodes after first three cues. Coloring by first cue (leftmost of colored PC plots), second cue (middle), and third cue (right) reveal distinct sorting directions that organize points within clusters. Sorting angles approach orthogonality in PC3 space. (B) Same as (A) for the network after 20k episodes. Activity space clusters are less distinct, and sorting angles less close to orthogonality. In particular, θ_{12} is more acute at 20k than 50k. (C) Independently trained networks consistently showed an increase of the angle between the first and second sorting directions, θ_{12}, from acute toward orthogonal with increased training from 20k to 50k episodes (after 20k training, $\theta_{12} = 38.5 \pm 4.9°$; after 50k training, $\theta_{12} = 61.5 \pm 5.9°$, average over 10 independently trained RNNs; *: $p = 0.001$, one-sided paired samples t-test. (D) Aggregated slope vs performance curve shows recency-like phenotype at 20k training episodes. (E) Same plots as in (D) but stratified by target cue identity. E.g. cue 1 plot measures serial position curve only when cue 1 is the target cue, i.e., 1-X-X-1, X-1-X-1, X-X-1-1. Curves stratified by cue 1 and 5 show strong recency-like phenotypes, whereas those for cues 2, 3, 4 do not. **: $p < 0.01$; ***: $p < 0.001$, one-sample t-test. Curves for (D),(E) are averages over 10 independently trained RNNs. For (A) and (B) PCA was performed on all correct trials (97% and 83% of all trials, respectively) and top 3 PCs explain 94% and 88% of activity state variance, respectively.

altered the reward scheme of the task to explicitly promote a recency-based strategy: "match" responses to X-A-X-A and X-X-A-A trials were now rewarded, with X-A-X-A resulting in a larger reward than X-X-A-A (A-X-X-A still had the highest reward; all other rewards were unchanged; see *Materials and Methods* for details).

As expected, under the new reward scheme, RNNs progressed through a larger slope deflection, indicating an exaggerated X-A-X-A/X-X-A-A ratio (*Appendix*, Fig. 3.14A). Interestingly, they also progressed in performance more rapidly than with the original reward scheme.

We checked the match error profile by lead cue, with the expectation that, unlike with the original reward scheme, RNNs with the recency-based reward scheme would have response distributions insensitive to lead cue during the period of recency-like phenotype. To our surprise we found this was not the case. Though less accentuated for the corresponding performance level in the original reward scheme (compare Fig. 3.6A and 3.3 where performances are similar at 72.4% and 77.6%, respectively), error profiles were still starkly uneven (ranging from 76% to 0%), in a directionally identical manner (Fig. 3.6A).

This suggests that since the end solution of both reward schemes is the same — reward is maximized by adopting a high target-selective strategy — recency simply affects the learning progression. More specifically, the recency-based reward scheme facilitates learning of the task, accelerating the solution state space transformation, albeit in the exactly the same manner (Fig. 3.6B and Supp. Fig. 3.14B).

3.5 Discussion

We presented a straightforward follow-up study of Wittig et al.'s important finding [Wit+16], which revealed two major twists: 1) what appears to be recency behavior may not in fact be true recency, and 2) in spite of this, a recency-based strategy may facilitate learning of the non-recency strategy progression. A lens from networks' state space solutions elucidated

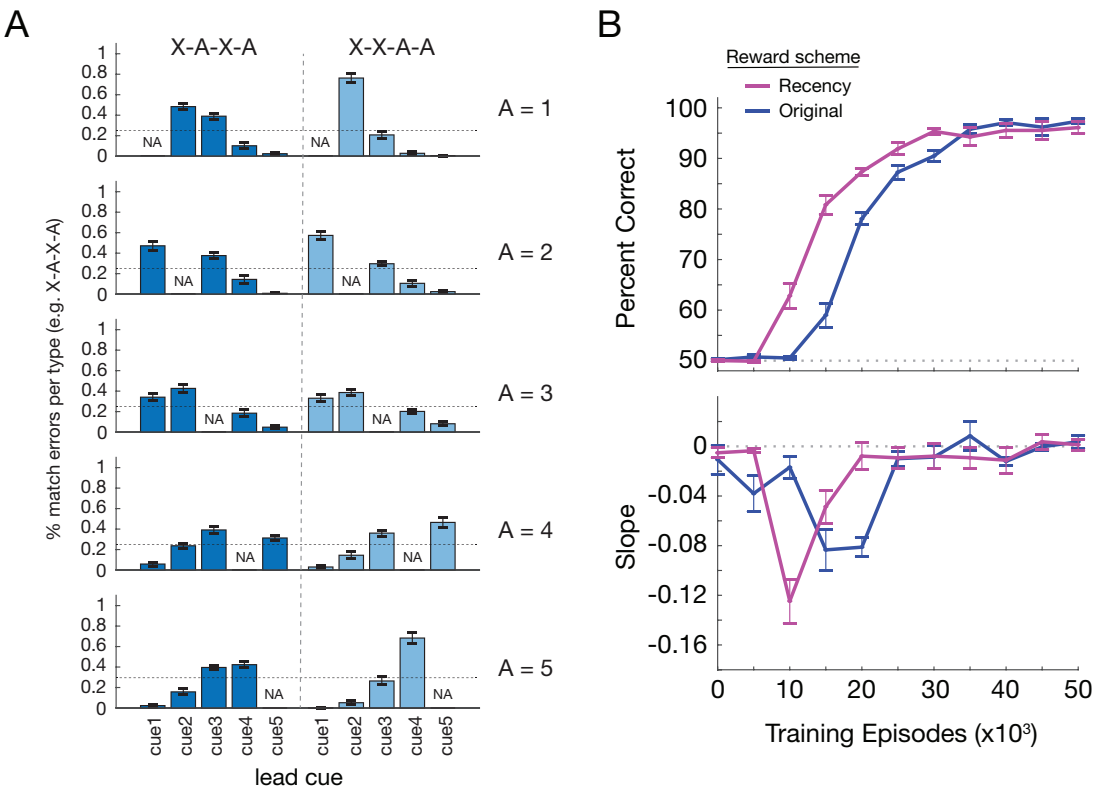


Figure 3.6. Recency-based reward scheme accelerates formation of the same solution space organization. (A) Serial position curve error profiles reveal similar uneven and directional trends as with the original reward scheme. Error rates across different lead cues ranged from 76% to 0%, far from the even 25% predicted by a pure recency-based strategy. Shown for RNN with similar performance and behavior phenotype as that in Fig. 3.3B (72.4% correct, 12k training trials). (B) Top, The recency-based reward scheme (magenta) leads to increases in performance earlier in training than the original reward scheme (blue). Bottom, The recency-based reward scheme causes an earlier and larger slope deflection in the expected direction during training than the original scheme. Bar and line plots represent mean of 10 independently trained RNNs. All error bars are SEM.

how these twists fit together into a coherent learning paradigm; one that may be similar to that used by monkeys and humans.

Our results support the hypothesis put forward by Wittig et al. [Wit+16] that a major component driving the differences observed of humans and monkeys on the working memory tasks may be task understanding, which we show can further be influenced by the capacity of the network dedicated to learning the task. Our finding suggests humans and monkeys may lie on two different points of the same learning progression.

RNNs trained until high task proficiency had activity trajectories suggestive of learned ignoring of distractor cues, yet upon deeper analysis revealed high-fidelity encoding of distractor cues as well as relevant cues. This finding follows the observation of several other studies that task irrelevant features are often encoded even when subjects explicitly understand their irrelevance [MB13]. In our analysis, activity space organization is shaped by task irrelevant cues (distractor cues) ("sorting directions") which causes the recency-like phenotype observed. Our results hypothesize that task irrelevant feature encoding rather than simple repetition may drive this observed behavior.

The finding that a recency-reward scheme facilitated attainment of the optimal target-selective strategy, raises an interesting possibility that some strategies might be useful simply to accelerate learning. Such strategies may themselves not be optimal, but act as a "catalytic strategy," helping shape the solution space's structure in a way that expedites achievement of the optimal solution. This concept is similar to "shaping," in which intermediate training goals are used to piece together a larger, often multi-step behavior. Catalytic strategies could then function as a type of internal shaping; for example, a monkey that derives some positive feeling from incorrect recency trials (perhaps the aesthetic pleasure of seeing a number repeat?) would get to optimal performance faster than a monkey zealously following the external reward program. Identifying employment of catalytic strategies — perhaps within the vast repertoire of heuristics humans use — promises to be a fascinating endeavour.

Our modeling work makes a number of predictions that can be tested:

1) In monkeys, is the match error profile across specific lead/distractor cues uniform as predicted by a pure recency-based strategy, or is it sharply varying as predicted by our RNN model? If the distribution is uniform, this suggests a true recency-based strategy, perhaps relating to prior experience or shaping behaviors not captured in our models which lack the rich experiential history of organisms like humans and monkeys. If the distribution is sharply varying, this would indicate the monkeys use a strategy well-described by our network model, showing a recency-like phenotype that in fact is not pure recency-based.

2) Our model predicts that using a recency-based reward scheme will speed up learning of the task. Adding this explicit recency-rewarding protocol during training of monkeys and comparing them to monkeys trained with the normal reward scheme can support or refute this prediction.

3) Our RNN models predict a particular solution space organization for the working memory tasks. This solution space result was robust, occurring repeatedly for independently trained networks, suggesting a potentially optimal solution for the task. Collection of large-scale neural recordings from monkeys (e.g. regions of prefrontal cortex thought to be critical to solving these working memory tasks) during and after training on the tasks (e.g. Match-First, set size 5) will allow assessment of how well the RNN model's solution space matches that of monkeys neural circuits.

Our findings reiterate the central message of Wittig et al. 2016: that extrapolation from one model system to another should be done with caution rather that assuming similarity based on coarse-grain measures. Wittig et al. showed monkey and human behaviors have different characteristic features even though performance may be comparable; here we show that the explanation for that difference may not be the most obvious, and that although they are different, monkeys and humans may lie on different points of the same learning continuum.

Finally, we note the inherent and unavoidable challenges of comparing model systems — humans and monkeys, and RNNs and humans and monkeys — and emphasize the importance of validation studies to assess findings and predictions from one model system to another. Monkeys have very different life histories than the humans they are compared to, and likewise, RNNs have even more starkly different life histories (i.e. none) prior to task introduction. In lab, monkeys must go through rigorous shaping protocols over the course of several months to successfully train them on experimental tasks. These include components as simple as sitting still or looking at a point on a screen consistently. Humans in contrast can be tested immediately or within a single day of training, inferring many of the implicit protocols of the experimental assessment (e.g. chair in the room likely means sit; screen likely is the place to look; language guidance). RNNs output utter nonsense corresponding to no meaningful behavior until training begins. The

shared features of humans, monkeys, and RNNs make them each useful in understanding the others, but their fundamental differences must be acknowledged in the analyses that extrapolate from one to the others.

Chapter 3, in full, is a reprint of the material as it appears in: Ben Tsuda, Barry J. Richmond, and Terrence J. Sejnowski. Understanding strategy differences between humans and monkeys with recurrent neural networks. *In preparation*, 2022. The dissertation author was the primary investigator and author of this paper.

3.6 Appendix

3.6.1 Code availability

The code and RNN models in this work will be made available at:

`https://github.com/tsudacode/humanVmonkey`

3.6.2 Supplementary Figures

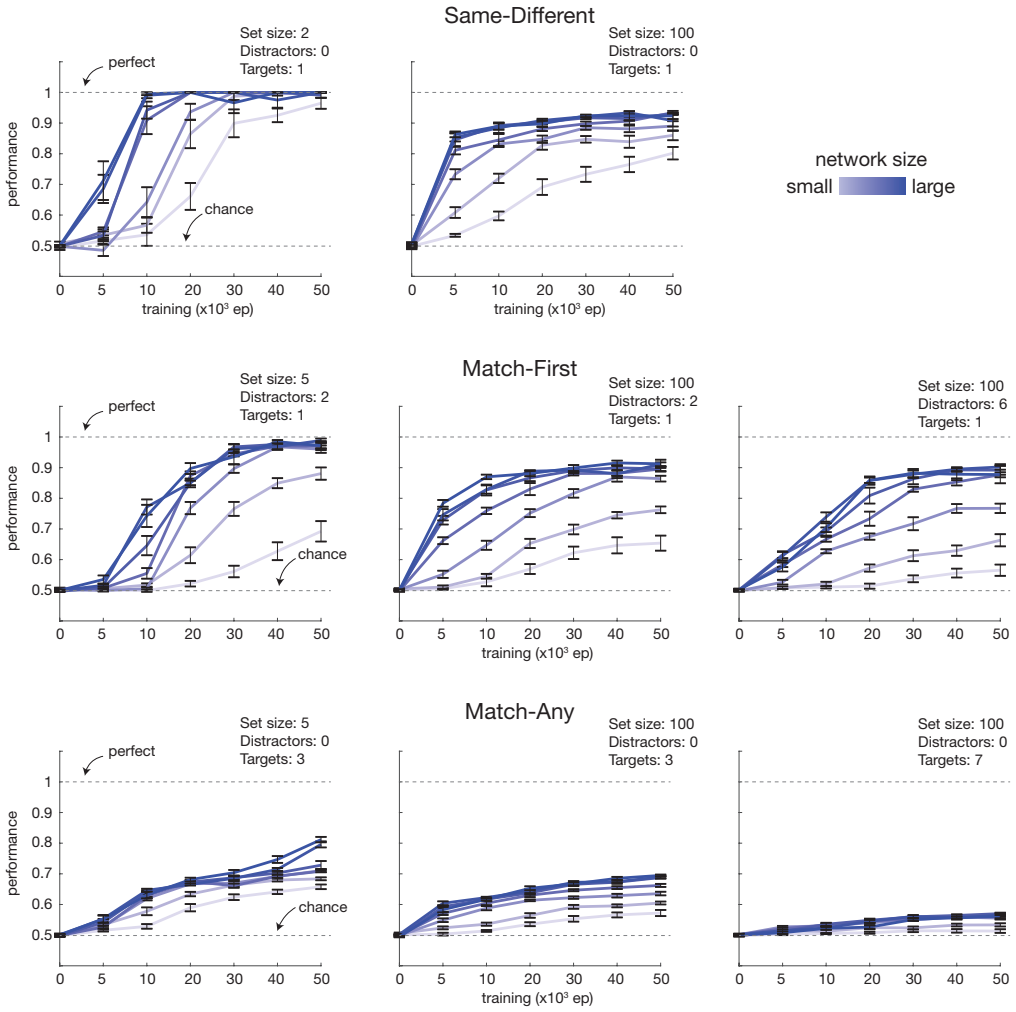


Figure 3.7. Performance over training episodes for RNNs across a size spectrum. Performance over eight variations of three working memory tasks from Wittig et al. 2016. Across all tasks, smaller networks (lighter blue colors) required more training episodes to achieve similar performance to larger networks (darker blue colors). For some tasks, networks across the whole size spectrum could gain proficiency (Same-Different, set size 2), for others only some networks reached maximum proficiency by the maximum training duration (e.g. Match-First, set size 5), and for others no network reached near perfect performance (e.g. Match-Any, set size 5). Similarly to humans and monkeys, RNNs performed the worst on Match-Any. All performance curves are the average performance of 10 independently trained networks of the given size, tested with 5000 test episodes at each training point. Error bars represent standard error over 10 independent networks tested.

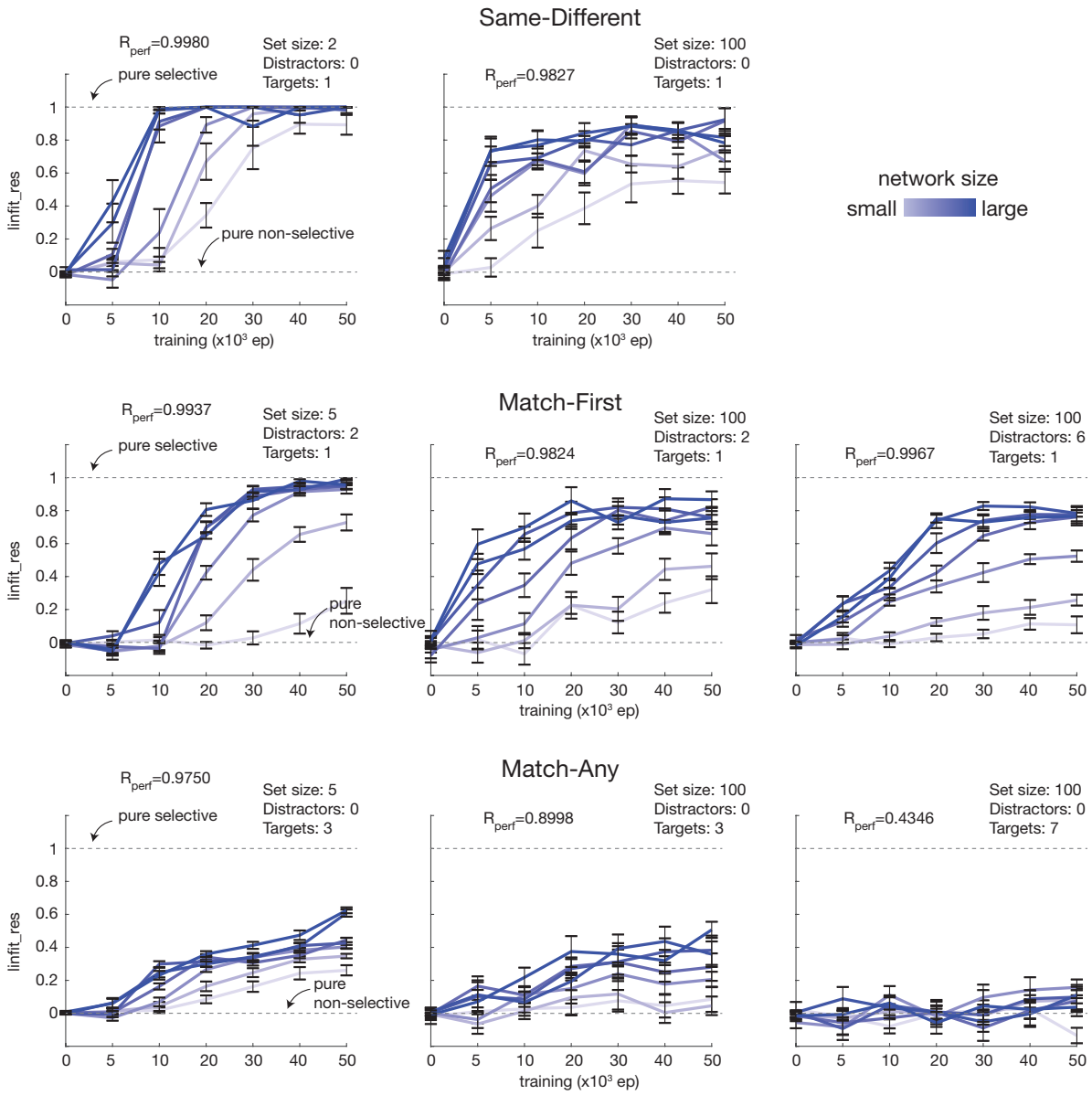


Figure 3.8. Serial position curve linear fit residuals on all task variations from Wittig et al. 2016. Residuals of serial position curve linear fits (linfit_res) as described in Wittig et al. 2016 closely mimic performance of RNNs on each task. A linfit_res of 1 represents a pure target-selective behavior whereas a value of zero represents a pure non-selective strategy. For each task variation, the correlation coefficient between linfit_res and performance at each network configuration (size x training) is given; all are upwards of 0.9 except Match-Any, set size 100 (R=0.4) which the RNNs had trouble learning in the training duration. All linfit_res curves are the average of 10 independently trained networks of the given size, tested with 5000 test episodes at each training point. Error bars represent standard error over 10 independent networks tested.

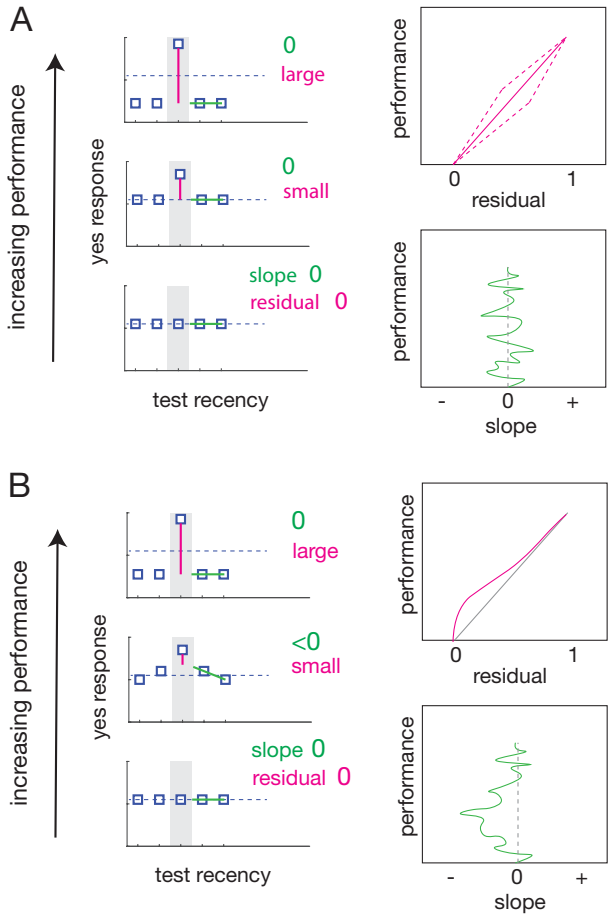


Figure 3.9. Two possible modes of strategy evolution. Schematics of two hypothetical strategy evolution scenarios shown for the Match-First task. For each part, left shows the progression of the serial position curve, right top shows the performance vs linear fit residual, and bottom right shows performance vs linear fit slope. (A) Strategy progression from random to increasingly target selective. Behavior begins as random with uniform yes rate across serial positions, then yes rate progressively increases at target location ("hit rate") and yes rate at all other positions ("false alarm rate") decreases uniformly. Residual of linear fit will increase with increasing performance and slope will trace random walk around 0. (B) Strategy progression from random to recency-like to target selective. From initial random behavior, yes rate at positions close to target location increase (or decrease more slowly) relative to yes rate at further positions, exhibiting recency-like behavior. Then hit rate continues to increase, while false alarm rate becomes uniform and low in a target selective strategy. Linear fit residual initially remains near 0 with initial increase in performance during recency-like behavior, then increases with further performance improvement. Slope beings at 0, deflects in an expected direction (negative for Match-First) then returns to zero when target selective strategy is achieved.

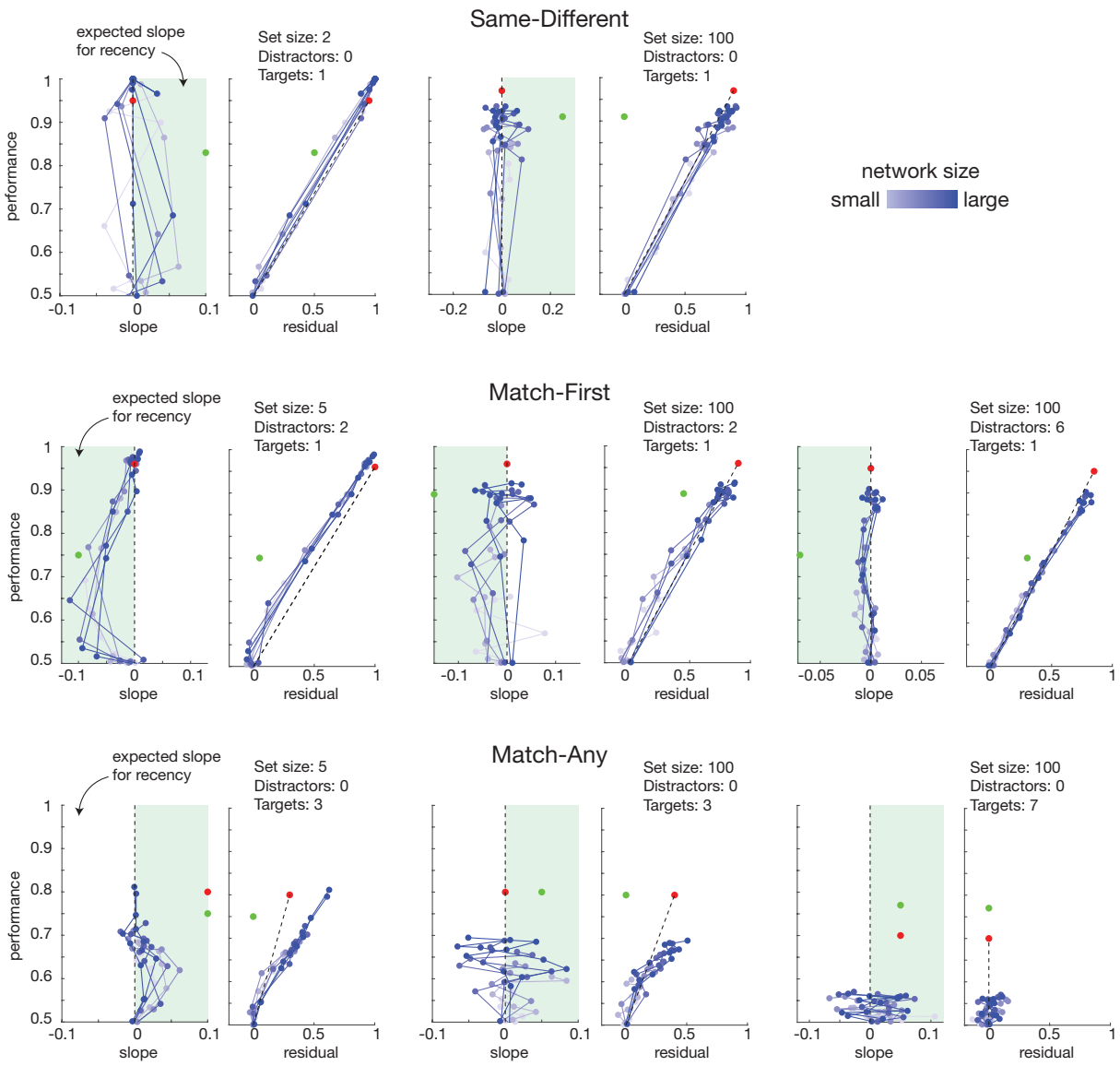


Figure 3.10. Serial position curve linear fit slope and residual for all task variations from Wittig et al. 2016. Slope and residual plots reveal clear instances where RNNs progress from random (low performance, small slope, small residual) to recency-like (higher performance, slope deflected in predicted (green) direction, low residual) to target-selective behavior (higher performance, small slope, high residual), for example Match-First, set size 5. Other RNNs did not have apparent recency-like behavior, though caveats on sampling depth and frequency require further investigation (e.g. Match-First, set size 100, distractors 2; Same-Different, set size 2; Supp. Fig. 3.11). In each slope plot, shaded green region indicates prediction prior on direction of slope deflection if recency-like behavior occurs. Dotted line in slope plots indicates 0 slope. Dotted line in residual plots connects from random (0 performance, 0 slope) to point achieved by human subjects in Wittig et al. 2016 (red dot). Human and monkey behavior is shown in all plots as red and green dots, respectively.

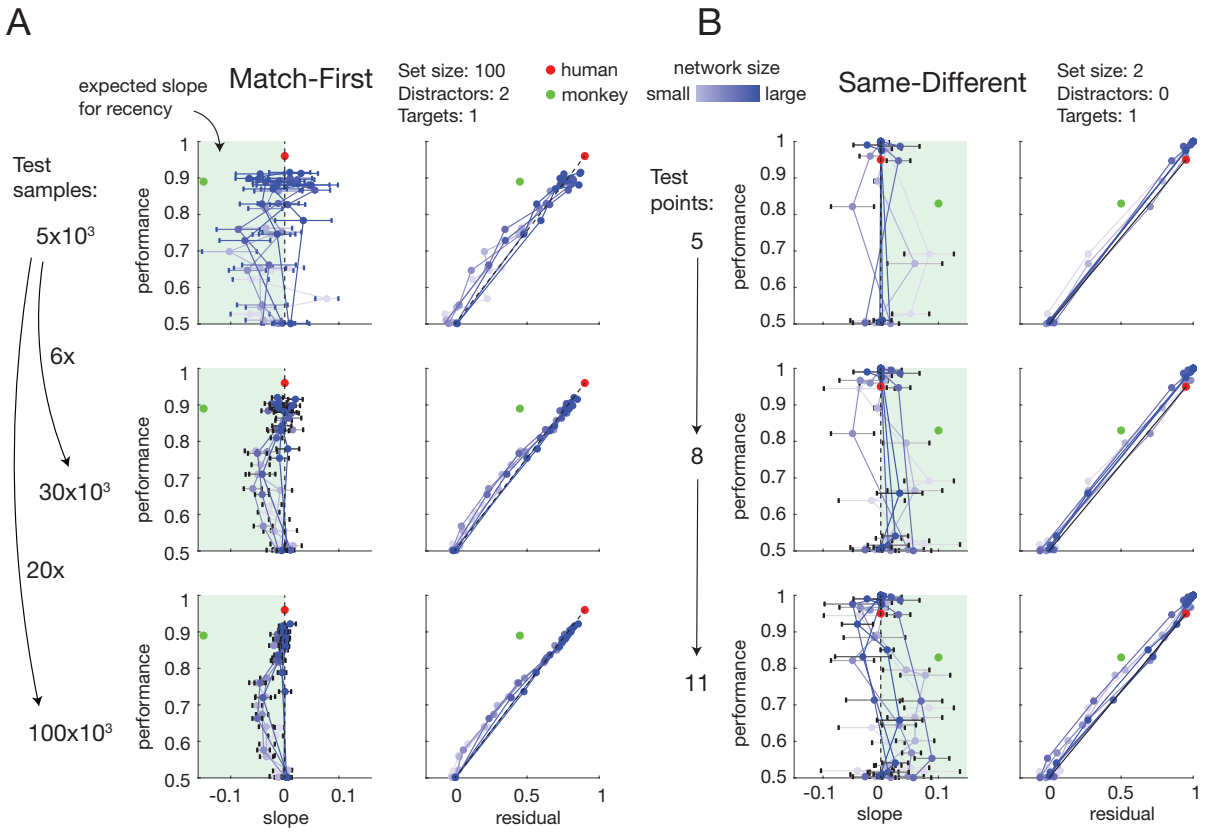


Figure 3.11. Undersampling depth and frequency masks occurrences of recency-like behavior by RNNs. (A) Example of undersampling depth. Serial position curve is constructed based on occurrence of error trials with particular cue arrangements. Task variations with large cue set size (100 vs 5 cues; 20x difference) require more sampling. Increased test sampling depth (5000 $\rightarrow$ 30000 $\rightarrow$ 100000 test samples) of same network reveals recency-like behavior when none was apparent at lower test sampling depth (5000 test samples). (B) Recency-like behavior could also be masked by sampling frequency for networks/tasks that rapidly progressed from random to target-selective strategy. Increased sampling frequency (training points $\in \{0, 11, 20, 40, 50\} \rightarrow \{0, 5, 11, 15, 20, 30, 40, 50\} \rightarrow \{0, 5, 7, 9, 11, 13, 15, 20, 30, 40, 50\}x10^3$) reveals recency-like behavior when none was apparent at lower frequency sampling.

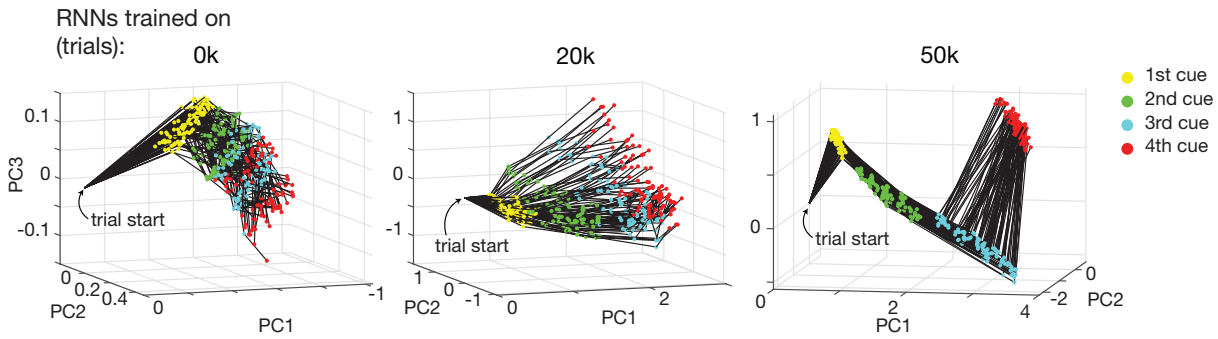


Figure 3.12. RNN population activity trajectories on Match-First trials. Network activity trajectories on test trials for three levels of training (trained on 0k, 20k, 50k trials). Colored points indicate network state after presentation of indicated cue in the trial sequence. Untrained networks (0k) do not exhibit any apparent ordering, with network states at different timepoints in different trials mixed up. Highly trained networks (50k) show precise trajectory shape and network state organization, with relevant cues (1st and 4th) moving network state in a direction apparently different than distractor cues (2nd and 3rd). Intermediately trained networks (20k) exhibit sequential ordering of network states without showing clear directional difference by specific cue position. For visualization and comparison purposes, all plots show random 100 trajectories aligned at the trial start for correct trials that occurred following a correct trial ("correct-after-correct"); previous trial outcome had a large binary influence on activity variability. For 50k trained RNNs, these made up 97% of correct trials, which in turn made up 97% of all trials. For 20k, this was 83% and 83%; and for 0k this was 51% and 51%. Top 3 PCs on all "correct-after-correct" trials explained >85% of variance in activity at each of the three training points assessed.

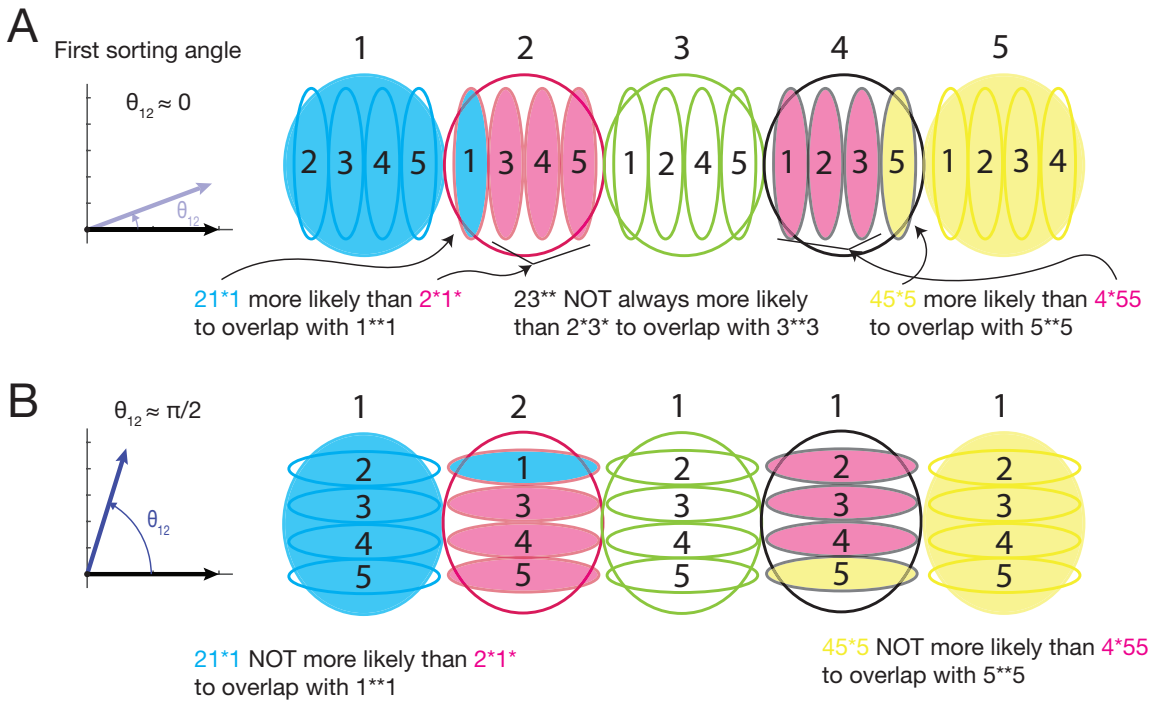


Figure 3.13. Predictions based on angle between first two cue-based sorting angles in RNN state space. θ_{12} is defined as the angle between the first and second sorting directions (see *Materials and Methods*). (A) When θ_{12} is small, the sorting direction of the second cue is aligned with that of the first, which manifests as cue-ordering based on second cue (smaller blobs) within clusters ordered by first cue (larger blobs) occurring in the same progression (increasing left to right). This conformation leads to particular spatial relationships, putting some clusters in closer proximity (and consequently at higher risk of overlap and erroneous readout) than others. The cluster defined by first cue 1 (big blue blob) contains the correct 1-X-X-1 match trials. Closest to this cluster is that with first cue 2 (big magenta blob). Within this cluster, the subcluster with second cue 1 is more proximal to the first cue 1 cluster than the subclusters with second cue 3, 4, or 5. This results in 2-1-X-X being more likely to overlap (and cause an erroneous match error) than 2-[3/4/5]-X-X, the former of which contains 2-1-X-1 sequences and the latter of which contains 2-X-1-1. This exactly predicts a recency-like relationship, i.e., match errors for 2-1-X-1 $>$ 2-X-1-1. A similar pattern occurs with first cue 4 cluster's (black large blob) subclusters' likelihood of overlap with first cue 5 cluster (large yellow blob). In contrast, overlaps between middle first cue clusters (2 (big magenta blob), 3 (big green blob), 4(big black blob)) do not have this spatial property, and would not be predicted to exhibit a recency-like relationship. For example, 2-3-X-X is less likely to overlap with the first cue-3-cluster (large green blob) than 2-4-X-X or 2-5-X-X, but more likely than 2-1-X-X. Given the uneven match error profiles (see Fig. 3B), this makes 2-3-X-3 $>$ 2-X-3-3 match errors (i.e., the recency phenotype) very unlikely. (B) When $\theta_{12} >> 0$ and closer to orthogonality, the proximity effect is removed so that match errors for X-A-X-A and X-X-A-A are equally likely across all clusters.

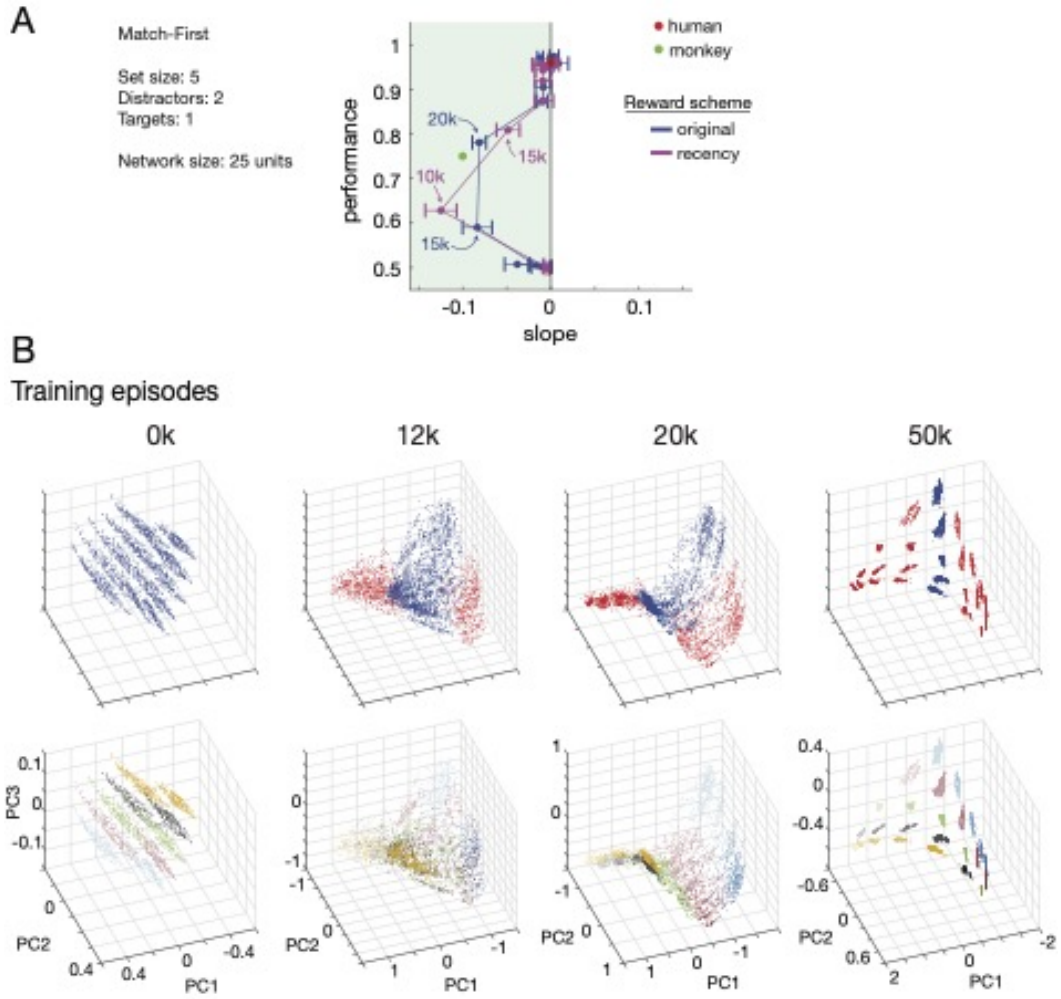


Figure 3.14. Recency reward scheme trained RNNs progress from recency to target-selective behavior and learn faster. (A) Slope vs performance plot for RNNs trained with a reward scheme that explicitly rewards recency behavior compared to original reward scheme. RNNs trained with recency scheme show the same slope vs performance progression, but appear to have an accentuated recency behavior at intermediate training (more negative slope) and move through the learning progression faster (compare recency reward scheme trained RNNs slope at 10k training trials to original at 15k). Labels in graph indicate trials trained on (e.g. 15k indicates RNN has been trained on 15e3 trials). Data shown is for 10 independently trained networks in each reward scheme. Error bars are SEM. (B). Example RNN's solution space during learning under recency reward scheme. The solution space of RNNs trained with the recency reward scheme follows a similar progression as those trained with the original scheme (Fig. 4A,B), with a very similarly arranged solution space map after extensive training and attainment of a target selective strategy (50k training trials). The progression occurs faster with recency reward scheme training, with performance and solution space after 12k training trials very similar to those of RNNs under the original scheme at 20k training trials. Top 3 PCs explained >88% of activity variance for all conditions.

Bibliography

[BS19] Silvia Bernardi and C. Daniel Salzman. The contribution of nonhuman primate research to the understanding of emotion and cognition and its clinical relevance. *Proceedings of the National Academy of Sciences*, 116: Dec. 2019, 26305–26312.

[Các+03] Mario Cáceres, Joel Lachuer, Matthew A. Zapala, John C. Redmond, Lili Kudo, Daniel H. Geschwind, David J. Lockhart, Todd M. Preuss, and Carrolee Barlow. Elevated gene expression levels distinguish human from non-human primate brains. *Proceedings of the National Academy of Sciences*, 100(22): 2003, 13030–13035.

[Elm+11] L. Caitlin Elmore, Wei Ji Ma, John F. Magnotti, Kenneth J. Leising, Antony D. Passaro, Jeffrey S. Katz, and Anthony A. Wright. Visual Short-Term Memory Compared in Rhesus Monkeys and Humans. *Current Biology*, 21: 2011, 975–979.

[ECF13] Aaron C. Ericsson, Marcus J. Crim, and Craig L. Franklin. A Brief History of Animal Modeling. *Missouri Medicine*, 110: 2013, 201–205.

[Fen+20] Guoping Feng, Frances E. Jensen, Henry T. Greely, Hideyuki Okano, Stefan Treue, Angela C. Roberts, James G. Fox, Sarah Caddick, Mu-ming Poo, William T. Newsome, and John H. Morrison. Opportunities and limitations of genetically modified nonhuman primate models for neuroscience research. *Proceedings of the National Academy of Sciences*, 117(39): 2020, 24022–24031.

[KS20] Robert Kim and Terrence J. Sejnowski. Strong inhibitory signaling underlies stable temporal dynamics and working memory in spiking neural networks. *Nature Neuroscience*, 24: 2020.

[Lee+19] Cathalijn H. C. Leenaars, Carien Kouwenaar, Frans R. Stafleu, Andre Bleich, Merel Ritskes-Hoitinga, Rob B. M. De Vries, and Franck L. B. Meijboom. Animal to human translation: a systematic scoping review of reported concordance rates. *Journal of Translational Medicine*, 17: 2019, 223.

[Lut14] Corrine K. Lutz. Stereotypic Behavior in Nonhuman Primates as a Model for the Human Condition. *Institute for Laboratory Animal Research Journal*, 55: 2014, 284–296.

[Man+13] Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503: 2013, 78–84.

[MB13] Louise Marshall and Paul M Bays. Obligatory encoding of task-irrelevant features depletes working memory resources. *Journal of Vision*, 13: 2013, 1–13.

[Mni+16] Volodymyr Mnih, Adria P Badia, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P Lillicrap, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for

Deep Reinforcement Learning. *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. Ed. by JMLR. Vol. 48. New York, 2016, 1928–1937.

[PP07] Derek C. Penn and Daniel J. Povinelli. On the lack of evidence that non-human animals possess anything remotely resembling a 'theory of mind'. *Philosophical Transactions of the Royal Society B*, 362: 2007, 731–744.

[RC19] Angela C. Roberts and Hannah F. Clarke. Why we need nonhuman primates to study the role of ventromedial prefrontal cortex in the regulation of threat- and reward-elicited responses. *Proceedings of the National Academy of Sciences*, 116: Dec. 2019, 26297–26304.

[Rus+18] Abigail A. Russo, Sean R. Bittner, Sean M. Perkins, Jeffrey S. Seely, Brian M. London, Antonio H. Lara, Andrew Miri, Najja J. Marshall, Adam Kohn, Thomas M. Jessell, Laurence F. Abbott, John P. Cunningham, and Mark M. Churchland. Motor Cortex Embeds Muscle-like Commands in an Untangled Population Response. *Neuron*, 97: 2018, 953–966.

[Tsu+20] Ben Tsuda, Kay M. Tye, Hava T. Siegelmann, and Terrence J. Sejnowski. A modeling framework for adaptive lifelong learning with transfer and savings through gating in the prefrontal cortex. *Proceedings of the National Academy of Sciences*, 117(47): 2020, 29872–29882.

[Ver+15] Jean-Michel Verdier, Isabelle Acquatella, Corinne Lautier, Gina Devau Stephanie Trouche, Christelle Lasbleiz, and Nadine Mestre-Frances. Lessons from the analysis of nonhuman primates for understanding human aging and neurodegenerative diseases. *Frontiers in Neuroscience*, 9: Mar. 2015, 64.

[Wan+18] Jane X Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Demis Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, 21: June 2018, 860–868.

[Wit+16] John H. Jr. Wittig, Barak Morgan, Evan Masseau, and Barry J. Richmond. Humans and monkeys use different strategies to solve the same short-term memory tasks. *Learning and Memory*, 23: 2016, 644–647.

[WR14] John H. Jr. Wittig and Barry J. Richmond. Monkeys rely on recency of stimulus repetition when solving short-term memory tasks. *Learning and Memory*, 21: 2014, 325–333.

Chapter 4

Neuromodulators generate multiple context-relevant behaviors in a recurrent neural network by shifting activity hypertubes

This chapter seeks to understand how molecules can modulate a neural circuit to effectively switch the mode of the neural network to generate a new set of behaviors. The investigation reveals how neuromodulators act on circuits to increase capacity and flexibility, how they dictate neural responses in an idiosyncratic manner, and how these properties relate to different modes of neuropsychiatric treatments. Through this work I elucidate a simple and powerful computational mechanism at play in networks of neurons and provide links that support and motivate further developments in precision medicine and machine learning technologies.

The work presented here is reproduced and adapted from: Ben Tsuda, Stefan C. Pate, Kay M. Tye, Hava T. Siegelmann, and Terrence J. Sejnowski. Neuromodulators generate multiple context-relevant behaviors in a recurrent neural network by shifting activity hypertubes. *In submission*, 2022.

4.1 Abstract

Mood, arousal, and other internal states can drastically alter behavior, even in identical external circumstances — a cold glass of water when you are thirsty is much more desirable than when you are sated. Neuromodulators are critical controllers of such neural states, with

dysfunctions linked to various neuropsychiatric disorders. Although biological aspects of neuromodulation have been well studied, the computational principles underlying how large-scale neuromodulation of distributed neural populations shifts brain states remain unclear. We use recurrent neural networks to model how synaptic weight modulation — an important function of neuromodulators — can achieve nuanced alterations in neural computation, even in a highly simplified form. We find that under structural constraints like those in brains, this provides a fundamental mechanism that can increase the computational capability and flexibility of a neural network by enabling overlapping storage of synaptic memories able to generate diverse, even diametrically opposed, behaviors. Our findings help explain how neuromodulators "unlock" specific behaviors by creating task-specific hypertubes in the space of neural activities and motivate more flexible, compact and capable machine learning architectures.

4.2 Introduction

Neuromodulators are a central mechanism in the biological control of neural states that manifest as mood, arousal, and other variable behavioral modes [Ina+12; Bur+17; MNH20; Bac+20; AK17; KE99; All+19]. Unlike standard, noise-sensitive models of context-dependent behaviors where exogenous cues are required to drive neurons [Man+13] (*Appendix*, Fig. 4.6), neuromodulation can modify nearly every aspect of *how* neurons transduce information, including intrinsic ion channels and synaptic strengths using a scalar signal. This enables stable alterations of network computations over longer timescales that are robust to fluctuations in external inputs [HM18] (*Appendix*, Fig. 4.6), supporting neural states like sleep [MNH20]. Pioneering studies on the lobster pyloric network [Mar12; ME84; JPH95] and other systems [KGF94; JGW09] have revealed how neuron-specific neuromodulation can precisely tailor central pattern generator rhythms. Yet it remains unknown how large-scale neuromodulation of vast distributed neural populations can control global network dynamics and dictate behavior as it does in large brains.

Fully understanding neuromodulation in brains is important for several reasons. First,

most psychiatric disorders either stem from or are directly related to neuromodulator dysregulation, as nearly all psychiatric drugs target neuromodulatory activity. Second, many of the psychiatric drugs currently in use only partially or imprecisely target neuromodulatory processes. Third, effects of many psychiatric treatments are highly variable, with some patients responding strongly and others failing to respond to multiple drugs. Fourth, neuromodulation acts via multiple mechanisms (as discussed below), allowing powerful circuit control but also making it difficult to fully understand how. Fifth, given the central role of neuromodulators in control of brains, a better understanding promises to make deep learning models based on brain architectures more flexible, more compact, and more efficient.

Neuromodulators affect several processes in brains including synaptic strengths, neural excitability, plasticity, and, indirectly, downstream circuit activity [New+12; NB12; Mar12; Wee+18]. Prior research has focused on different aspects of neuromodulation, including Yu and Dayan [YD05] who modeled the role of acetylcholine and norepinephrine in Bayesian probability estimations of uncertainties; Stroud et al. and Vecoven et al. [Str+18; Vec+20] who considered modulation of the neural activation function; Beaulieu et al. [Bea+20] who formulated neuromodulation as a separate network that masks effector networks; Miconi et al. [Mic+19] who used modulation of synaptic plasticity to train networks; and Hasselmo et al. [HSB95] who developed a model incorporating experimental work on multi-factor neuromodulator-specific circuit dynamics, particularly in hippocampal memory processes. Our model of synaptic weight modulation shares some similarities to previous models, particularly to the neural excitability models [Str+18; Vec+20]. Both lead to increased flexibility and versatility, yet they operate through independent mechanisms both biologically [NB12] and computationally (see *Appendix*).

We focus here on a critical aspect of neuromodulation in brains — synaptic weight modulation [Mar12; NB12; KGF94; JPH95; Has06] — of which no general, biologically-plausible model exists. We consider a simplified approximation in which neuromodulators act as nearly uniform synaptic weight amplifiers or dampeners within a local region of a neural network. We show how this form of neuromodulation establishes overlapping synaptic memories

corresponding to unique dynamic activity landscapes within a structurally-conserved neural network to generate unique behaviors. We demonstrate how neuromodulated circuits give rise to idiosyncratic, non-linear dose-response properties that can differ depending on the mode of neuromodulation. Using a well-established neuromodulation-mediated behavioral paradigm in *Drosophila*, we show how this form of neuromodulation naturally handles intermediate neural states, and as such, generalizes models of discrete internal state switching [CPM19; Cer+20] to continuous state transitions. Although many mechanisms may influence behavioral shifts, we show that a simple multiplicative factor applied to weights already acts as a powerful network control device, allowing neuromodulators to vastly increase the capability and complexity of computation in brains and making artificial neural networks more flexible, compact and capable.

4.3 Materials and methods

4.3.1 Modified Go-NoGo tasks

The classic Go-NoGo task has two possible stimuli (positive pulse referred to as positive stimulus or +; no pulse, also referred to as null stimulus or $\emptyset$). The agent is trained to give a positive output (+1, "Go") for the positive stimulus and zero output (0, "NoGo") for the null stimulus. In the modified Go-NoGo task we added a second possible behavioral set: NoGo for the positive stimulus and negative output (-1, "AntiGo") for the null stimulus. The network was trained on the classic Go-NoGo behavior in the absence of neuromodulator and on the new NoGo-AntiGo behavior in the presence of neuromodulator. The 3 behavior and 9 behavior variants of the modified Go-NoGo task followed a similar paradigm with additional added behaviors. In the 3 behavior version, a third behavior of positive stimulus $\rightarrow$ AntiGo, null stimulus $\rightarrow$ Go was added. In the 9 behavior version all possible stimulus $\rightarrow$ output response pairs were added. As such, Behavior 1 was + $\rightarrow$ AntiGo, $\emptyset \rightarrow$ AntiGo; Behavior 2 was + $\rightarrow$ NoGo, $\emptyset \rightarrow$ AntiGo; Behavior 3 was + $\rightarrow$ Go, $\emptyset \rightarrow$ AntiGo; ...; Behavior 9 was + $\rightarrow$ Go, $\emptyset$ Go. In our simulations each trial lasted 200 timesteps of 5ms each for a total represented duration of

1 second.

4.3.2 Neuromodulatory neural network model

For our simulations we used a continuous rate recurrent neural network (RNN) model with biologically plausible parameters similar to RNNs in prior works [KS20; Tru+20]. Consistent with biological neural networks, we implemented Dale's Law using the method in Song et al. [SYW16] such that each neuron was either excitatory or inhibitory. For all our simulations we used a RNN with $N = 200$ neuron units, 80% excitatory and 20% inhibitory. In the RNN each neuron can be connected to any other neuron with probability p_{con} (p_{con} was initialized at 0.8 in our simulations), and each neuron receives weighted input from connected neurons to produce a firing rate governed by the neural dynamical equation

$$\tau \frac{dx}{dt} = -x + Wr + W_{in}u + N(0, 0.1) \tag{4.1}$$

where $\tau \in \mathbb{R}^{1\times N}$ is the synaptic decay time constant for the N neurons in the network, $x \in \mathbb{R}^{1\times N}$ is the synaptic current variable for the N neurons, $W \in \mathbb{R}^{N\times N}$ is the matrix of synaptic weights between all N neurons, $r \in \mathbb{R}^{1\times N}$ is the output firing rates of the N neurons in the network, $W_{in} \in \mathbb{R}^{1\times N}$ are weights associated with external input u, and $N(0, 0.1)$ is added noise drawn from a normal distribution with mean 0 and variance 0.1. The output firing rate for the neurons is given by an elementwise nonlinear transfer function transformation of the synaptic current variable. In our network we used the standard logistic sigmoid function as implemented by prior models [KLS19; KS20]:

$$r = \frac{1}{1 + e^{-x}} \tag{4.2}$$

The synaptic connectivity matrix W was randomly initialized from a normal distribution with zero mean and standard deviation $g/\sqrt{N \cdot p_{con}}$, where g is the gain. We set $g = 1.5$ as previous

studies have shown that networks operating in a high gain regime ($g \geq 1.5$) support rich dynamics analogous to those of biological networks [KS20; KLS19; LB13]. The synaptic decay time constants were randomly initialized to a value in the biologically plausible range of 20–100 ms. As in Kim et al. 2019, we used the first-order Euler approximation method to discretize Equation 4.1 for the simulations; for neuron i:

$$x_{i,t} = (1 - \frac{\Delta t}{\tau})x_{i,t-1} + \frac{\Delta t}{\tau}(\sum_j W_{ji} r_{ji,t-1} + W_{ui} u_{t-1}) + N(0, 0.1) \tag{3}$$

Output was generated by taking all recurrent network neurons' activities and passing them through a weighted output unit

$$o_{network} = W_{out} r + b_{out}$$

where $W_{out} \in \mathbb{R}^{N \times 1}$ are the neural output weights and b_{out} is the output unit's bias term. RNNs were trained by backpropagation through time using AdamOptimizer with a least square error objective function.

To apply an amplifying or dampening neuromodulatory effect, target neurons' weights were scaled by the neuromodulatory factor. For whole network neuromodulation this effect was applied to all neurons in the RNN; for subpopulation neuromodulation the effect was applied only to the selected subpopulation of neurons.

For the modified Go-NoGo task, RNNs were trained until one of two possible stop criteria was met: 1) average trial least square error over the last 50 trials was under a threshold of 1, or 2) 10,000 training trials was reached. Performance on the task was then assessed by evaluating the percentage of test trials that matched the following performance criteria: for Go trials, output was required to reach 1.0 ± 0.2 by timestep 120 (full trial was 200 timesteps); for NoGo trails, output was required to be 0.0 ± 0.2 and for AntiGo trails -1.0 ± 0.2 at timestep 120.

4.3.3 Comparison to context-dependent cued model

For comparison, we created a cue-driven RNN model and trained it on the Modified Go-NoGo task. The cue, which we refer to as the "context cue", was delivered through an additional input channel and signaled which output behavior was desired. We created models with two types of cues: transient cues were constant value inputs present only during the stimulus input period (t=0 to t=75); persistent cues were constant value inputs across the whole trial. For comparisons between models, we ran models with cue pairs of +1.0/-1.0 (2.0 sep in *Appendix*, Fig. 4.6d,e), +0.5/-0.5 (1.0 sep), and +0.2/-0.2 (0.4 sep).

To compare the neuromodulatory and context-cued RNNs tolerance to noise, we ran two types of simulation experiments. First, we added Gaussian noise of zero mean and standard deviation ranging from 0 to 4 to all input signals and measured RNN performance as the difference between target and actual output. At each level of noise, we simulated 4 independent models 100 times each and averaged their performances (*Appendix*, Fig. 4.6d). In a second experiment, after training, we simulated pure intrinsic network activity (no inputs). We again added Gaussian noise of zero mean and standard deviation ranging from 0 to 4. For each batch of 100 simulations on 4 replicate networks, we examined the consistency of final network states (t=200). To measure this, we computed the mean Euclidean distance of the 100 simulations final-time-step states from the centroid, giving a measure of final network state spread. Larger mean Euclidean distance (higher spread) indicated more variable activity trajectories; lower mean distance (lower spread) indicated highly consistent activity trajectories (*Appendix*, Fig. 4.6e).

The phase portraits with flow fields were created by simulating a network to produce two state trajectories from distinct behavioral contexts. For this we used a neuromodulatory network (f_{nm}=9) and a context-cued model with persistent cues 0/+2.0. For each network configuration (with and without neuromodulator, and cued model), we computed the derivative of the neurons' rates across the trajectories to generate vector fields depicting the intrinsic flow fields of the

network in the absence of any driving input. For the exogenously cued model, we then calculated the external drive required across the alternative cue-driven trajectory (cue=+2.0) to achieve the corresponding activity trajectory. We projected state trajectories and flow fields into the first three PCA space for visualization (*Appendix*, Fig. 4.6f).

4.3.4 Neuromodulation of multiple subpopulations and multiple levels

For neuromodulation of non-overlapping subpopulations, same-sized groups of neurons were choosen randomly without any overlap and neuromodulator applied to each for a given behavior. For overlapping subpopulations, groups of neurons were chosen randomly allowing overlap (*Appendix*, Fig. 4.11).

Neuromodulation at different levels ("multi-factor networks") was done by applying different neuromodulation factors (f_{nm}). For the 9-behavior modified Go-NoGo this was done using factors $\in [1:1:9]$, i.e., for Behavior 1 no factor was applied ($f_{nm} = 1$), for Behavior 2 $f_{nm} = 2$, for Behavior 3 $f_{nm} = 3$, et cetera.

To test networks across the range of subpopulation sizes with single or multiple neuromodulator factors on n-behavior modified Go-NoGo tasks, stop criteria were adjusted to account for the increased behaviors: 1) average trial least square error over last n*25 trials was under threshold of 1, or 2) 15,000 training trials was reached. Performance on the tasks was assessed as before. For overlapping subpopulations, overlap was quantified in two ways. For each network, the number of neurons neuromodulated in 2 or more subpopulations was measured (*Appendix*, Fig. 4.11d). Overlap was also quantified by measuring the average number of neuromodulated subpopulations a neuron in the network was a member of (*Appendix*, Fig. 4.11e).

4.3.5 Single neuron inputs, functional clustering, and selectivity index

Neural activity in a RNN is a complex function of all the neuron activities tracing all the way back in time. To understand how neuromodulation shifted synaptic inputs at the single neuron level, we considered the first time point in a trial. For any trial, at t=0 all activities

are randomly initialized from a normal distribution. As a result, at t=1, a neuron reacts only to the weighted inputs of its incoming connections, uncontaminated by propagating recurrent activity dynamics from past timepoints. Analysis of neuron activity at this timepoint is shown in *Appendix*, Fig. 4.12a–c.

In order to examine whether trained models contained functionally specialized neurons, we grouped neuron activities by combination of subtask (which maps one-to-one with neuromodulatory state) and stimulus given ("stimulus-subtask combinations"). We averaged activity of each neuron over time and trials within each group. This resulted in a matrix of time-trial averaged neuron activities with a number of rows equal to the stimulus-subtask combinations, and number of columns equal to the number of neurons. Using k-means, we clustered neurons with similar activity levels across stimulus-subtask combinations. We computed a silhouette score to find the optimal number of clusters, which for the RNN in *Appendix*, Fig. 4.12, was 6. The silhouette score computed for 5 and 6 clusters differed only by 0.7% and the additional cluster was very small and similar to an existing cluster, so we conducted further analysis with 5 clusters for simplicity.

To measure the selectivity of individual neurons for particular stimulus-subtask combinations, we calculated a "selectivity index" (si) for each neuron j:

$$\mathrm{si}_j = \frac{\bar{r}_j^{max} - \bar{r}_j^{second\_max}}{\bar{r}_j^{max}}$$

where $\bar{r}_j$ is the average firing rate of neuron j over the trial duration, $\bar{r}_j^{max}$ indicates the maximum $\bar{r}_j$ across all the stimulus-subtask combinations and $\bar{r}_j^{second\_max}$ indicates the second highest $\bar{r}_j$ over all the stimulus-subtask combinations. The selectivity index thus captures a normalized approximation of how uniquely active a neuron was for a given stimulus-subtask combination.

4.3.6 Network population dynamics

To represent whole network population activity dynamics we sought a low dimensional representation of whole population activity. We used principal component analysis (PCA) since the leading components capture the largest projections of activity variability, which we hypothesized would effectively separate our neuromodulatory conditions if large differences occurred [CY14]. We found this was the case. We found qualitatively similar results using multidimensional scaling which finds projections designed to best preserve distances in high-dimensional activity space. For our figures we display the first 3 PCs, as these captured a large amount of the activity variance (80–92% explained across the analyses) and effectively represent the activity dynamic differences in the analyses.

To map the neuromodulation-dependent activity subspace, we generated 100 independent stimuli series consisting of random numbers drawn from a uniform distribution between 0 and 1 at each time point (t=0 to t=200) and fed this into the RNN with and without neuromodulation (shotgun stimulus mapping from *Appendix*, Fig. 4.15a), which defined non-overlapping subspaces of activity space.

To analyze neuromodulation transition curves we compared activity under intermediate neuromodulation levels with linear interpolation. Linear interpolation was done by evenly dividing the distance between no and full neuromodulation activity states into 9 sections analogous to the 9 neuromodulation levels assessed. These 9 points in activity space were then used to generate output that is plotted in *Appendix*, Fig. 4.16. The geometry of the neuromodulation-based transition was assessed by calculating the Euclidean distance of intermediate neuromodulation level states at a given trial timepoint to the nearest point on the line connecting no and full neuromodulation states at that timepoint. These distances are plotted in Fig 4.3i. The angle of departure (AoD) was defined as the angle formed by the line between no and full neuromodulation states and the line between no neuromodulation and the first neuromodulation level states,

which can be calculated as:

$$\vec{v_1} = \vec{p_F} - \vec{p_N}$$

$$\vec{u_1} = \vec{p_{L1}} - \vec{p_N}$$

$$AoD = cos^{-1} \frac{\vec{u_1} \cdot \vec{v_1}}{|\vec{u_1}||\vec{v_1}|}$$

where $\vec{p_N}$ is the network state with no neuromodulation, $\vec{p_F}$ is the network state with full neuromodulation, $\vec{p_{L1}}$ is the network state with the first level of neuromodulation.

4.3.7 EC50

The EC50 of a network was defined as the level of neuromodulation that led to half the output of full neuromodulation. For the results reported, we used EC50 calculated for the positive stimulus. For this stimulus in the modified Go-NoGo task, a non-neuromodulated network outputs +1 and a fully neuromodulated network outputs 0 (measurements for output level were taken at 0.5 s through the trial); the EC50 for the network in this case is the amount of neuromodulator required to output 0.5. The EC50 was calculated by fitting a sigmoid curve to the progression of output (from +1 to 0 in this case) with increasing neuromodulation level (Fig. 4.3d)

$$output = 1 - \frac{1}{1 + e^{a \cdot f_{nm} + b}}$$

where f_{nm} is the neuromodulation level. EC50 neuromodulation level was calculated by finding the intersection of the sigmoid and the half-maximal output; for half-maximal output of 0.5, EC50 $= -b/a$. Sigmoid curves were fit using a least squares fit.

4.3.8 *Drosophila* sugar sensitivity task

We implemented a computational version of Inagaki et al. 2012 to train our network models. During training, models were presented with a constant sugar concentration (external input proportional to sugar concentration) for 100 timesteps (equivalent to 500 ms) and trained to output a probability of PER. For fed and 2-day starved training we used a piece-wise linear approximation estimated from Inagaki et al. 2012. To compare boxplots of MAT, one-way ANOVA followed by *t*-test with Bonferroni correction was used as in Inagaki et al. 2012.

4.3.9 Mean acceptance threshold (MAT)

Analogous to the analysis done for flies in Inagaki et al. 2012, for each RNN a sigmoid was fit

$$PER = \frac{1}{1 + e^{-a \cdot log_2 \frac{x_{sugar}}{MAT}}}$$

where a is the slope of the sigmoid. When $PER = 0.5$ then $x_{sugar} = MAT$, as defined in Inagaki et al. 2012 [Ina+12]. Sigmoid curves were fit using a least squares fit.

For intermediate neuromodulatory level (f_{nm}=3) MAT variability analysis, a normalized change in MAT (%ΔMAT) was calculated:

$$\%\Delta MAT = \frac{MAT_{f_{nm}=3} - MAT_{f_{nm}=1}}{MAT_{f_{nm}=5} - MAT_{f_{nm}=1}}$$

where $MAT_{f_{nm}=x}$ is the RNN's MAT with neuromodulation factor x. %ΔMAT gives a network normalized metric for how much the intermediate neuromodulation (f_{nm}=3) moved the fly from no neuromodulation (f_{nm}=1) to full neuromodulation (f_{nm}=5) sensitivity.

4.3.10 Electrical modulation

We administered electrical modulation as an external current applied for the duration of the trial. "On-target" modulation was applied to the neuromodulated neuron population and "random" modulation was applied to a randomly selected group of neurons of equal size; these could include both neuromodulated or non-neuromodulated neurons. All neurons (both excitatory and inhibitory) within the selected subpopulation were given identical external current modulation. For fixed electrical modulation simulations, a current of magnitude 1 was applied (+1 for excitatory modulation; -1 for inhibitory).

For graded electrical modulation, networks that did not achieve e-mod50 at maximum stimulation (-9 units) were assigned a surrogate e-mod50 value of -10 to calculate correlation to EC50. To account for possible missed correlation due to this substitution, correlation of EC50 to output at maximum modulation (-9) was also calculated (*Appendix*, Fig. 4.18c).

4.4 Results

4.4.1 Neuromodulation creates multiple weight regimes within shared synaptic connections

The effects of neuromodulators on synaptic weights present a mode of circuit control [NB12] that is poorly understood in brains — both how it is implemented at scale and the computational mechanisms by which it shifts coordinated activity to generate different behaviors. Several recent studies on cell type diversity have made clear that brains contain a complex array of neuromodulators that act with carefully coordinated spatiotemporal precision [Xie+21; RF18; Zha+21; LBN18; Pat+18]. As a first step, we sought to assess whether a simplified form of neuromodulation — modelled as a uniform multiplicative factor acting on synaptic weights in a recurrent neural network (RNN) — could help us understand how neuromodulators control neural state. Although other modes of neural network control such as exogenous contextual cuing have been shown to successfully shift network behavior [Man+13], uniform weight modulation

represents a completely different biological and computational mechanism, which, given the complex, non-linear, and often unpredictable nature of RNNs, requires explicit assessment.

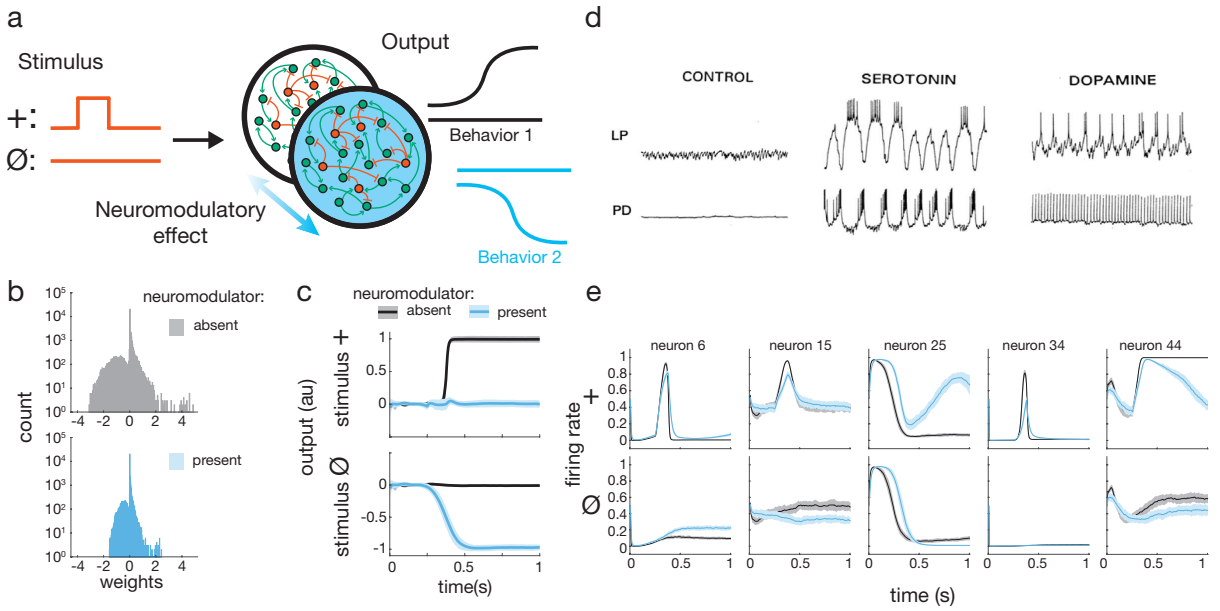


Figure 4.1. Neuromodulation weight scaling separates overlapping synaptic memory regimes. a, Modified Go-NoGo task. Given a stimulus (either + or ∅), in absence of neuromodulatory effect the recurrent neural network (RNN) should produce outputs from the Behavior 1 repertoire and in presence of neuromodulator, from Behavior 2. **b**, Approximation of neuromodulatory effect implemented in the model: all synaptic weights in the RNN are multiplied by a constant factor, here 0.5. **c**, Mean output to + and ∅ stimuli of 10 independently trained RNNs on the modified Go-NoGo with global neuromodulation factor 0.5. Shading represents standard deviation. **d**, Individual neuromodulators elicit unpredictable transforms of firing patterns in crustacean stomatogastric ganglion (STG) neurons. Reprinted from Neuron 76, Marder, Neuromodulation of Neuronal Circuits: Back to the Future, 1-11, 2012, with permission from Elsevier. **e**, Five example neurons' activity patterns from neuromodulated model RNN show complex nonlinear transformations analogous to crustacean STG activity changes under neuromodulation.

We used a modification of the classic Go, No Go experimental paradigm ("modified Go-NoGo;" Fig. 4.1a; see *Materials and Methods*) to assess whether given identical input stimuli, uniformly shifting all the weights in a RNN — for example, scaling all weights by a factor of $\frac{1}{2}$ (Fig. 4.1b) — could elicit an independent behavior from the same network. We found that neuromodulation in this form was able to generate distinct behaviors for the task (Fig. 4.1c), demonstrating that this simple mechanism operating in brains can effectively separate synaptic

memory regimes within a fixed circuit and access them through uniform scaling of weights to "unlock" specific behaviors (*Appendix*, Fig. 4.8). We found this result held over a wide range of neuromodulatory factors (*Appendix*, Fig. 4.9). Furthermore, reminiscent of neuromodulator effects on individual neuron activity patterns observed in lobster stomatogastric ganglion (Fig. 4.1d) and other organisms [Mar12; JPH95; Joh+11; KGF94], we found that both global activity (*Appendix*, Fig. 4.7a) and individual neuron activities were unpredictable, displaying non-linear transforms (Fig. 4.1e and *Appendix*, Fig. 4.7b).

4.4.2 Targeted neuromodulation can toggle across multiple global network states

In brains, neuromodulators are released in specific regions — some tightly localized, others broadcast widely — to influence local and global neural output. We found that RNNs with neuromodulated subpopulations of sizes across a broad range (100%–10% of the whole population) consistently supported the opposing behaviors of the task (Fig. 4.2a–c). Just as some neuromodulators affect neurons in a cell-type specific manner, for example selectively influencing activity of excitatory or inhibitory neurons with corresponding receptors [WM14], we found targeting of neuromodulator in this manner also was able to support the task (Fig. 4.2d,e).

To assess the flexibility of this neuromodulatory mechanism, we asked whether multiple unique behaviors could be learned and unlocked from a single network through targeted neuromodulation. Using an extended version of the modified Go-NoGo task (see *Materials and Methods*), we found that neuromodulation of distinct subpopulations or with distinct neuromodulation levels could support multiple behaviors, up to the maximum 9-behavioral Go-NoGo task we tested (Fig. 4.2f,g and *Appendix*, Figs. 4.10, 4.11; see *Materials and Methods*).

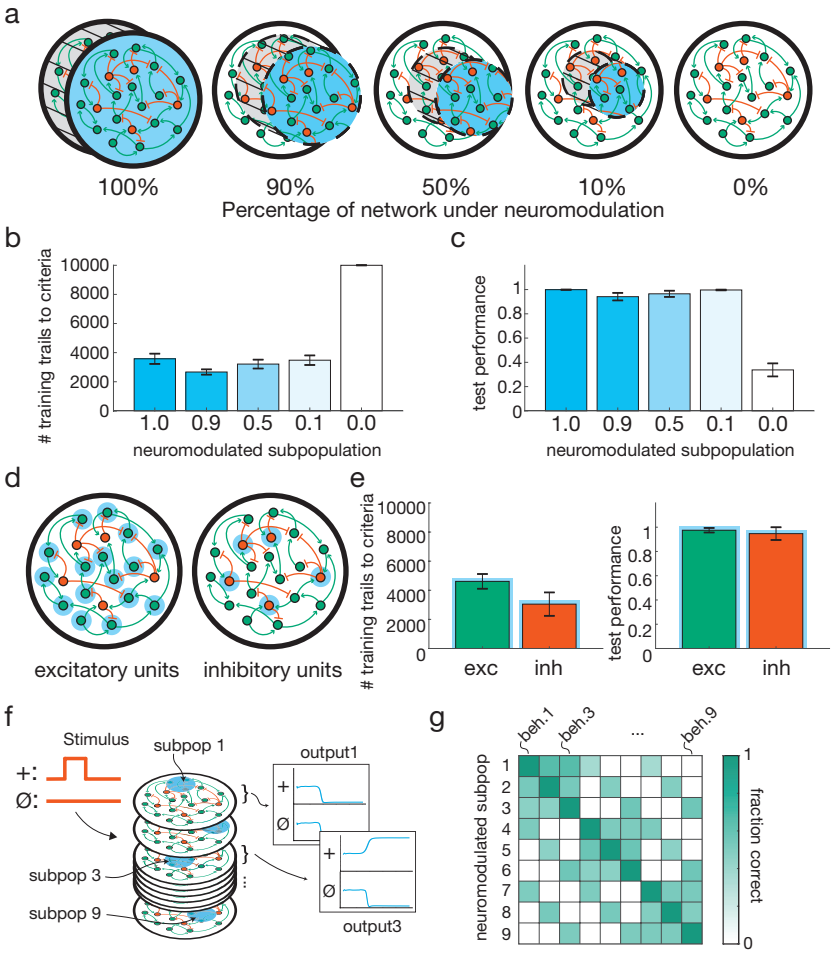


Figure 4.2. Targeted neuromodulation flexibly supports multiple behaviors. a, A range of different sized neural subpopulations embedded within a RNN were neuromodulated (factor (f_{nm}) = 0.5). 100% was positive control demonstrated in Fig. 4.1; 0% was negative control. **b**,**c**, RNNs with embedded neuromodulated subpopulations across the size spectrum could support the opposing behaviors of the modified Go-NoGo. **b**, Number of training trials to reach stop criteria (see *Materials and Methods*). **c**, Test performance (1 is 100% correct; see *Materials and Methods*). **d**, Neuromodulation of exclusively excitatory or inhibitory neurons (blue annuli). **e**, Excitatory or inhibitory neuromodulation supported learning of modified Go-NoGo. **f**, 9-behavior modified Go-NoGo task with unique neuromodulated subpopulations (subpops) and example corresponding outputs. **g**, RNNs successfully learned the task from **f** with 9 targeted subpops (each 10% of the RNN, non-overlapping; f_{nm}=2.5). Application of neuromodulator to any subpop unlocked a specific behavior set (beh.) from the 9-behavior repertoire (fraction of trials correct is ≈1 on diagonal; see *Materials and Methods*). Off-diagonal fraction correct due to partial output overlap between behavioral sets.

4.4.3 Distinct global network activity hypertubes with non-linear transition dynamics emerge from neuromodulation

To understand how this form of neuromodulation leads to network behavior shifts, we analyzed the coordinated activity of all neurons in the RNN in the absence and presence of neuromodulator. At the individual neuron level, neuromodulation shifted the net difference of excitatory and inhibitory inputs, which in turn altered the recurrent propagation of activity over time and resultant internal network dynamics (*Appendix*, Figs. 4.12–4.14). At the whole population level, neural activity trajectories for the same stimulus with and without neuromodulator followed non-overlapping, stereotyped paths, or hypertubes [GB18; Low+18; Nie+21], in activity space (Fig. 4.3a). Through amplification of synaptic weights, neuromodulation effectively resets all the pins in the pinball machine, altering activity flow patterns through the RNN (*Appendix*, Fig. 4.6f and 4.15a).

The distinct hypertubes in activity space derive from a common underlying neural network. This suggests that there must be a transition between the hypertubes accessible through intermediate amounts of neuromodulation. To characterize this transition, we applied intermediate levels of neuromodulation to the RNN after training, which mapped a smooth transition from trajectories of the non-neuromodulated hypertube to those of the fully neuromodulated hypertube (Fig. 4.3b). Furthermore, intermediate neuromodulation generated intermediate outputs from the network (Fig. 4.3c). In this way, just as RNN neural trajectories have been shown to naturally address temporally varying sensory-motor patterns [GB18], neuromodulated neural trajectories provide a means to naturally respond to intermediate, even unexperienced neural states (e.g. hunger levels). To characterize the transition in output behavior, we measured the output of the RNN at the midpoint of each trial for each level of neuromodulation. We found that increasing neuromodulator levels led to non-linear (exponential or sigmoidal) progression from non-neuromodulated (Go: +1) to fully neuromodulated behavior (NoGo: 0) (Fig. 4.3d).

We next asked whether neuromodulatory transition dynamics were tightly constrained,

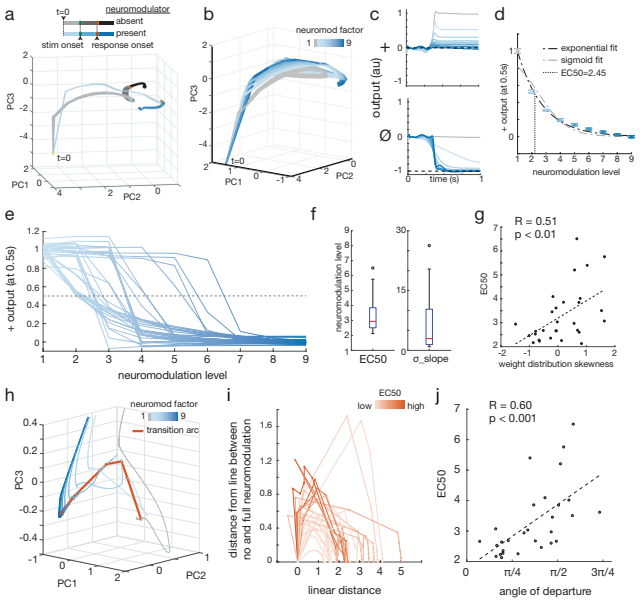


Figure 4.3. Neuromodulation separates activity hypertubes with idiosyncratic nonlinear transition dynamics. **a**, Global network activity dynamics after PCA-based dimensionality reduction for positive stimulus. Activity trajectories follow stereotyped, non-overlapping paths, or hypertubes in activity space. **b**, For a RNN trained with neuromodulatory factor of 9, intermediate levels of neuromodulation lead to partial transitions (traces in shades of blue) toward full neuromodulation activity hypertube (traces in darkest blue). **c**, Partial neuromodulation maps to intermediate output behaviors. **d**, Transition from none to full neuromodulation behavior is non-linear (best-fit exponential & sigmoid shown) and defines network sensitivity, measured as EC50. **e**, Output transition for 29 networks independently trained on the modified Go-NoGo task and tested across intermediate levels of neuromodulation. **f**, 29 independently trained RNNs exhibit large variability in transition dynamics, with EC50 ranging from 2.1 to 6.5 and sigmoid slope (σ_slope) from 0.9 to 26.3. **g**, Network EC50 is positively and significantly correlated with skewness of global weight distribution. **h**, Zoomed image demarcating path at a given timepoint over a network's neuromodulation transition manifold (orange; transition arc) which connects hypertubes corresponding to different neuromodulation levels in activity space (grey–blue trajectories). Blue shading same as in **b**. **i**, At a given timepoint (here t=100), intermediate levels of neuromodulation (represented by individual points) trace an arc between no neuromodulation (leftmost point on each curve) and full neuromodulation (rightmost point on each curve) states in phase space. Purely linear interpolation would lay along the x-axis at "distance from line (y-axis)" = 0. Each curve represents an individual network's transition arc across neuromodulation levels. Arcs are shaded by relative EC50. Arcs represent cross-section of full transition manifold connecting hypertubes across all timepoints at intermediate neuromodulation; each point on the arc is average of a cross section of a single hypertube. **j**, Angle of departure (angle formed between direct path and first neuromodulation level hypertube) exhibits a strong positive correlation with network EC50. Transition in direction more orthogonal or away from full neuromodulation state results in lower sensitivity, i.e. higher EC50.

defining a conserved property of neuromodulation, or highly variable depending on individual network characteristics. To test this, we independently trained 29 RNNs. All RNNs exhibited non-linear transition dynamics best fit by an exponential or sigmoid function (Fig. 4.3e and *Appendix*, Fig. 4.15b), but networks' sensitivities to neuromodulator and rates of transition varied drastically. To quantify this variability, we defined a "half maximal effective concentration" (EC50) as the amount of neuromodulator required to generate a half maximal output (see *Materials and Methods*). The EC50 of individual networks trained with a full neuromodulator factor of 9 ranged from 2.1 to 6.5 (3.1x range; Fig. 4.3f, left) and rate of transitions (steepness of the transition dynamics sigmoid) varied widely as well from 0.9 to 26.3 (Fig. 4.3f, right). This result reveals a previously unknown phenomenon that may contribute to the wide individual variability of neuropsychiatric drug sensitivities observed clinically [Fur+19]; a "circuit-based sensitivity."

We next sought to characterize what properties of the networks contribute to the variability in sensitivity to neuromodulator. We found that the skewness of networks' weight distributions exhibited a positive correlation with EC50s ($R=0.51$, $p<0.01$; Fig. 4.3g), suggesting networks with more positively skewed weights (longer tail of strong excitatory weights) were less sensitive to neuromodulator. To further understand the source of network sensitivity variability, we characterized the shape of the networks' activity transition curves across neuromodulation levels (Fig. 4.3h). At a given trial timepoint, purely linear interpolation yielded linear sensitivity relationships with invariant EC50 (*Appendix*, Fig. 4.16). In contrast, progressive neuromodulation defined an arc (Fig. 4.3h,i), which, collectively across all timepoints formed a curved transition manifold connecting each neuromodulation-specific activity hypertube. The geometry of this transition arc (measured as the angle of departure; see *Materials and Methods*) was strongly correlated to network sensitivity (Fig. 4.3j). This suggests that while individual networks can achieve identical performance on the trained task (no and full neuromodulation), the geometry of their population activities at intermediate neuromodulations is unique, leading to emergent sensitivity profiles.

Excessive neuromodulation can also occur either pathologically or pharmacologically.

To model this, we applied neuromodulation at levels higher than those used during training and found neural dynamics could sometimes (but not always) diverge from trained activity hypertubes into an adjacent region of activity space, translating into inappropriate output behavior (*Appendix*, Fig. 4.15c–e).

4.4.4 Neuromodulated RNN replicates dopamine-mediated starvation-dependent sugar sensitivity in *Drosophila*

Given our finding that neuromodulation provides a natural means of handling intermediate, unexperienced neural states, we next sought to evaluate our model's ability to recapitulate the behavioral effects of neuromodulation observed *in vivo*. The neuromodulator dopamine controls the sugar sensitivity behavior of *Drosophila* [Ina+12], as measured by proboscis extension reflex (PER) probability, which increased with both duration of starvation (fed, 1 day, 2 day starved) and concentration of L-dopa administered in their diet (0, 3, 5 mg/ml) (Fig. 4.4a).

To assess if our neuromodulation model could reproduce these results, we trained RNNs with neuromodulated subpopulations (20% subpopulation) to reproduce the fed and 2 day starved sugar sensitivity curves of flies (no neuromodulation (f_{nm}=1) for fed; f_{nm}=5 for 2 day starved). We then tested the RNNs' behaviors at an intermediate, never-before experienced neuromodulator level (f_{nm}=3). The RNNs produced a shifted sensitivity curve very similar to that exhibited by 1 day starved flies and the flies fed an intermediate L-dopa concentration of 3 mg/ml (Fig. 4.4b).

The behavior of the RNNs reliably mimicked the intermediate behaviors of flies *in vivo* because intermediate neuromodulation caused a continuous shift in the RNN's activity hypertube between "fed" and "2 day starved/5 mg/ml L-dopa" hypertubes (*Appendix*, Fig. 4.17a–c). Furthermore, our model reveals that transition manifolds (defined by the hypertubes connecting intermediate neuromodulation levels) are unique to networks, predicting wide-ranging natural variability in fly starvation-based sugar sensitivity profiles (*Appendix*, Fig. 4.17d). Neuromodulation leads to natural handling of never-before experienced neural states by creating a network configuration such that the neuromodulatory transition manifold has a geometry that

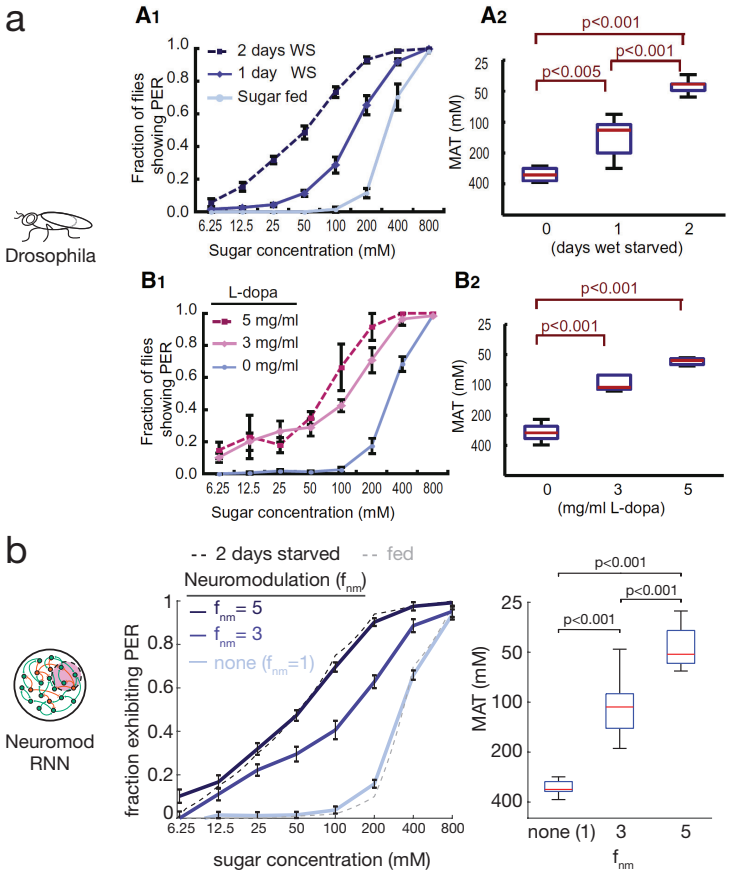


Figure 4.4. Neuromodulated RNNs reproduce *Drosophila* sugar sensitivity behaviors. a, *Drosophila* sugar sensitivity behaviors from Inagaki et al. 2012 measured as PER behavior vs sugar concentration. Reprinted from Cell 148, Inagaki et al., Visualizing Neuromodulation In Vivo: TANGO-Mapping of Dopamine Signaling Reveals Appetite Control of Sugar Sensing, 583-595, 2012, with permission from Elsevier. **b**, Neuromodulated RNNs trained on extremes of *Drosophila* sugar sensitivity (no neuromodulation factor (f_{nm}=1) for fed and f_{nm}=5 for 2 days starved) exhibit similar intermediate (f_{nm}=3; untrained) and extreme (no neuromodulation (f_{nm}=1) and f_{nm}=5; trained) behaviors (n=10; error bars are SEM; same statistical test as in Inagaki et al. 2012 for boxplots, see *Materials and Methods*).

leads to intermediate outputs.

4.4.5 Electrical modulation shifts neural dynamics through an independent circuit effect

Other endogenous and exogenous influences can alter neural circuit dynamics through mechanisms that may be shared or independent, and understanding relationships between such interventions is vital for safe and effective treatment. We used our model to compare whether

exogenously delivered electrical modulation of a neuromodulated circuit — analogous to use of optogenetics experimentally [Dei15] and deep-brain stimulation (DBS) and transcranial magnetic stimulation clinically [Kra+21; Ris+20; Gou+19] — alters network activity in an analogous manner to chemical neuromodulation or operates through an independent effect (Fig. 4.5a).

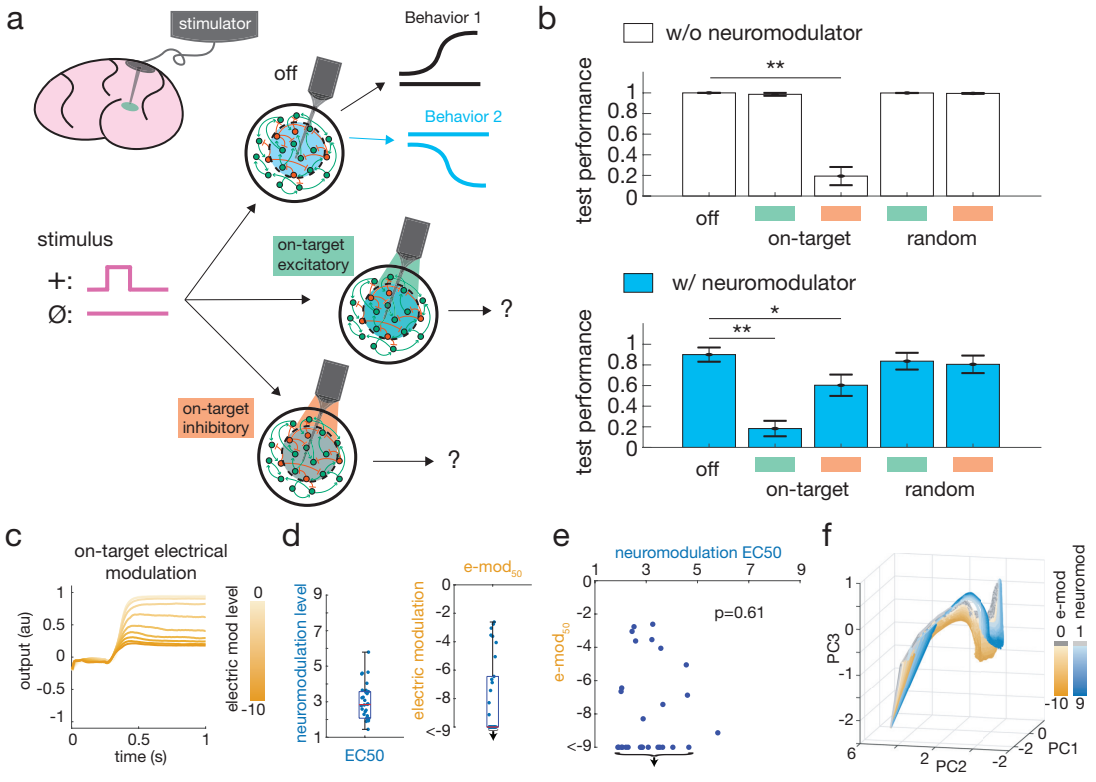


Figure 4.5. Targeted electrical modulation shifts network dynamics through independent circuit effect. a, Schematic of DBS and analogous electrical modulation (e-mod) of a neuromodulated RNN. **b**, Test performance was significantly impaired in absence of neuromodulation (white bars) when inhibitory on-target electrical modulation (e-mod) was given (p=3.91e-11) and with neuromodulation (blue bars) when excitatory and inhibitory on-target e-mod was given (p=2.20e-08 and p=2.18e-02, respectively). *:p<0.05, **:p<0.01. **c**, RNN output in absence of neuromodulation to + stimulus with increasing inhibitory e-mod. **d**, For 30 RNNs: Left: neuromodulation EC50. Right: electrical e-mod$_{50}$. Down arrows in **d**, **e** indicate RNNs that did not achieve e-mod$_{50}$ at maximum stimulation. **e**, No significant correlation between networks' EC50 and e-mod$_{50}$. **f**, Neuromodulation and electrical modulation push activity trajectories along different transition manifolds enabling independent transition dynamics.

We computationally applied electrical modulation (excitatory or inhibitory current; see *Materials and Methods*) to RNNs trained with neuromodulated subpopulations and found that

while electrical current given to random subpopulations did not affect the RNNs' performances, current delivered to the neuromodulated subpopulation could significantly affect network output (Fig. 4.5b), shifting behaviors directionally toward the opposing neuromodulation condition behavioral set (*Appendix*, Fig. 4.18a,b).

In each RNN, increasing the level of targeted electrical input — for example inhibitory modulation in the absence of neuromodulator — led to a graded transition in behavior similar to the transition observed with graded administration of neuromodulator (Fig. 4.5c). To compare these conditions, we measured the amount of electrical input that led to output of 50% of the fully-neuromodulated condition (e-mod$_{50}$) analogous to EC50 for neuromodulator levels. Interestingly, RNNs also exhibited idiosyncratic circuit-based sensitivity to electrical modulation, with e-mod$_{50}$ under inhibitory modulation ranging from -2.6 input current to not achieving e-mod$_{50}$ by the maximum modulation we administered (-9 input current; >3.5x range) (Fig. 4.5d, right).

We then assessed whether networks' electrical and chemical sensitivities were related, and found no significant correlation (Fig. 4.5e). Since some RNNs' outputs did not reach e-mod$_{50}$, saturating before the maximum electrical input given, — further evidence of a different mechanism at play — we also measured each RNN's output at maximum electrical input (input=-9) and similarly found no significant correlation to EC50 (*Appendix*, Fig. 4.18c). The lack of correlation suggests that networks insensitive to chemical modulation may still be highly sensitive to electrical modulation and vice versa. Consistent with this, we found that electrical modulation progressively shifted population dynamics along a manifold transition distinct to neuromodulation, enabling different rates of transition (Fig. 4.5f). This may help explain why some patients who fail pharmacologic treatments sometimes respond dramatically to DBS. By utilizing an independent mechanism to chemical modulation, DBS exploits a parallel circuit-sensitivity to achieve therapeutic efficacy.

4.5 Discussion

Neuromodulation in brains drives unique neural function in health and disease. Using an RNN model, we showed how neuromodulators, through simple scaling of synaptic weights, can generate unique behavioral modes from a single RNN. The importance of our findings is not that our model was able to solve tasks like Go, No Go, which other computational models with mechanisms like contextual cuing can also solve. Rather, we showed how neuromodulation, a completely independent and previously uncharacterized control mechanism that is highly relevant biologically and clinically, is able to rapidly reconfigure a network with only a single scalar input.

Our model provides insights into neuromodulation that are related to other recently elucidated principles of neural computation. We showed that neuromodulation leads to separation of distinct activity hypertubes, similar to those observed by Goudar and Buonomano [GB18] and Nieh et al. [Nie+21], with neuromodulation effectively disentangling neural trajectories by separating them in phase space analogous to the work of Russo et al. [Rus+18] in motor cortex. Just as neural trajectories provide transformations in phase space that naturally handle temporal variation of sensory-motor patterns [GB18], neuromodulation leads to transformations in phase space that elucidate a biological mechanism for handling intermediate and continuously transitioning neural states, even if never experienced before. We demonstrated the biological use of this property through replication of Inagaki et al.'s findings in *Drosophila* [Ina+12]. In this way, the level of neuromodulation acted as a controller on the amount of disentangling of neural trajectories, using internal neural state (amount of weight modulation) to control output behavior. Such a system is robust to external noise (*Appendix*, Fig. 4.6), since far apart neural states generate trajectories that are widely separated in phase space.

Through this analysis we discovered a feature of networks previously unreported to our knowledge: "circuit-based sensitivity," which helps explain the clinical observation of high variability in drug and other therapeutic response [Fur+19], alongside more standard

explanations like enzyme variant-dependent drug metabolism and clearance rates. This emergent sensitivity property of neuromodulated networks is related to the well-known "many solutions" phenomenon of neural networks where different weight configurations can produce identical output [Meh+20; Mah+19]. Unlike studies focusing on variability in networks producing identical output, our model allowed study for the first time of the transition dynamics under neuromodulation, revealing unique geometric configurations of phase space underlying emergent network sensitivity profiles.

Future studies aimed at identifying circuit parameters that control this transition dynamic geometry will be critical for understanding and use in therapeutic optimization. The idiosyncratic nature of circuit-based sensitivity aligns with current efforts in precision medicine calling for the need to consider each patient as an idiosyncratic individual — here we provide computational evidence for this claim and its particular importance in neuropsychiatric treatment [Pro+19]. Fully understanding the relationship between chemical and electrical modulation and sensitivity is also crucial. Although our simplified model suggests how the modes of modulation influence dynamics (see *Appendix*), further analytical and experimental investigation into their relationship as network dynamics evolve over time could provide deeper insights.

Our formulation of the neuromodulatory effect on synaptic weights is a simplification of the true biological mechanism. Elaborating our model to support differential weight modulation (e.g. via multiple neuromodulators and neuromodulator receptor subtypes on specific cell-types) [HJ10; Bac+20], neuromodulator multiplexing [WM14; LBN18], and metamodulation [NB12; RS10; KE99] will likely lead to even more sophisticated network behavior. Our model also uses arbitrary neuromodulation levels, whereas brains likely use specific levels for optimal functionality. Future investigation into which levels are optimal and methods of learning these will be important.

Finally, neuromodulation provides interesting directions for machine learning (ML). By separating synaptic memory regimes in a single network, we demonstrate how a network can have much greater flexibility and increased capacity, supporting a library of unique behaviors

for overlapping external contingencies. Furthermore, each behavior can be rapidly accessed through targeted application of the relevant neuromodulatory factor. High capacity, compact networks with high-speed access to different output modes presents a promising component for ML development and storage-limited applications like edge computing. Additionally, through the separation of memory regimes that effectively splits a single RNN into multiple processors, this mechanism may provide a means of realizing the super-Turing capability of specific RNN configurations as defined by Cabessa and Siegelmann [CS14]. Future theoretical assessment of neuromodulated RNNs' capacity will establish if this simple mechanism is sufficient to exceed the Turing limit.

Chapter 4, in full, is a reprint of the material as it appears in: Ben Tsuda, Stefan C. Pate, Kay M. Tye, Hava T. Siegelmann, and Terrence J. Sejnowski. Neuromodulators generate multiple context-relevant behaviors in a recurrent neural network by shifting activity hypertubes. *In submission*, 2022. The dissertation author was the primary investigator and author of this paper.

4.6 Appendix

4.6.1 Code availability

The code and RNN models in this work will be made available at:

`https://github.com/tsudacode/neuromodRNN`

4.6.2 Supplementary Figures

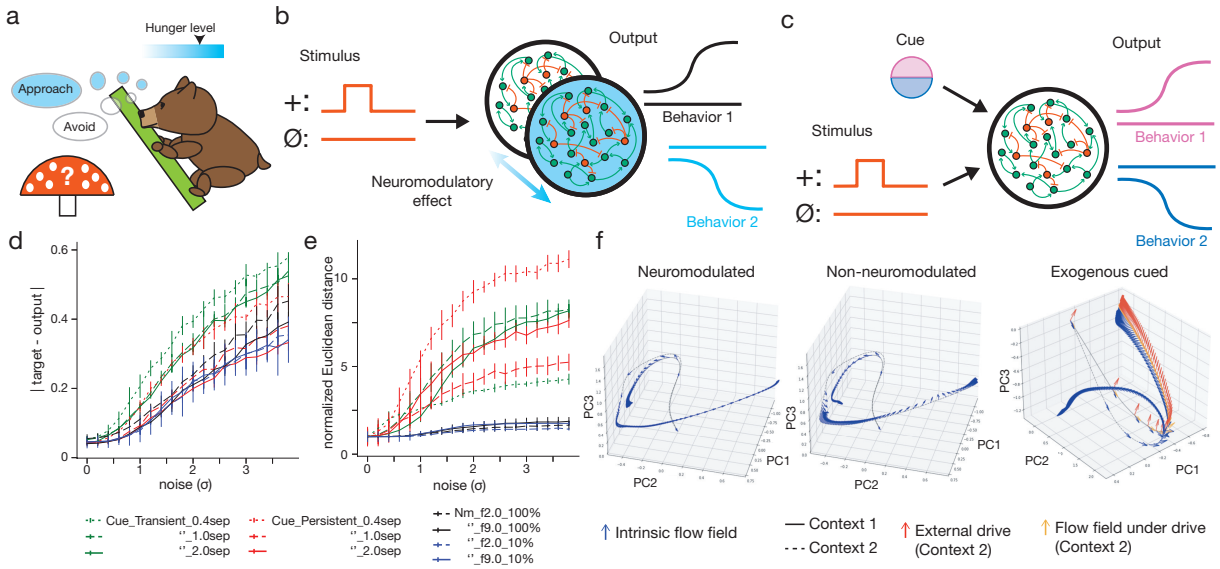


Figure 4.6. Noise robustness of neuromodulated and external cue-driven networks. **a**, Neuromodulation is a fundamental biological mechanism, underlying e.g. how hunger level dictates a bear's behavior when encountering a mystery mushroom. **b**, Modified Go-NoGo task in neuromodulated network. Neuromodulatory effect alters network configuration to produce different behaviors. **c**, Standard, context-dependent model with contextual cue input. External cue is required to drive activity to generate different outputs. **d**, Output error for neuromodulated and exogenously cued models with increasing input noise. Noise is drawn from normal distribution with standard deviation σ. Transiently cued models (green) — biologically more realistic than persistently cued (red) — have errors that increase more rapidly with increasing noise compared to neuromodulated (black—100% of network, blue—10% network; dashed lines factor 2.0, solid lines factor 9.0) or persistently cued models (red). **e**, Average Euclidean distance of neural trajectory endpoint between replicates when given no driving input across increasing levels of input noise. Neuromodulated networks (blue, black) generate network dynamics robust to noise, unlike cued models (green, red) which rapidly become unpredictable, exhibiting high variability — measured as increased Euclidean distance of trajectory endpoints on replicate trials — as noise levels increase. Cued models in **d**,**e** are highly dependent on specific cue values and separation amplitude between cues (dotted lines represent networks that had input cue amplitudes separated by 0.4 arbitrary units (au), dashed lines for cues separated by 1.0 au, solid line for cues separated by 2.0 au). **f**, Neuromodulation changes the flow field of the network in activity space (differences of vector fields in left and middle PCA activity plots). Blue vectors represent network's internal flow field (no driving input) along trajectories. Cue-based model (right) relies on external input (red arrows) to drive the network along desired trajectory in phase space (yellow arrows for Context 2 trajectory). For visualization clarity, flow field vectors indicate direction of activity change (without magnitude) and are shown only when network activity is changing more than a threshold of 0.02 units of Euclidean distance per timestep in full network activity space.

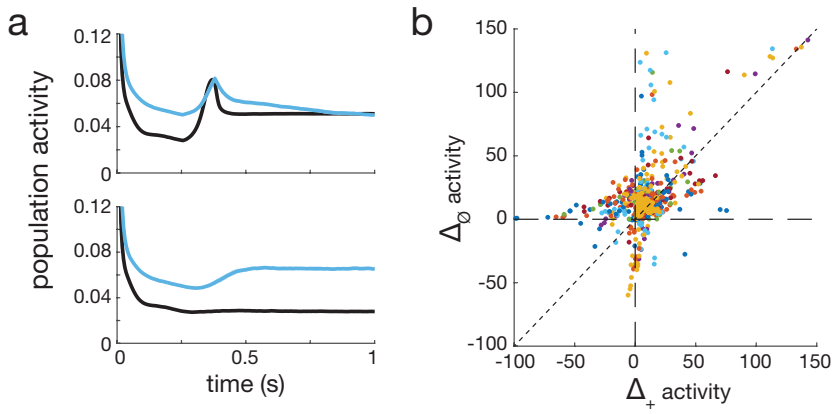


Figure 4.7. Neural activity under neuromodulation. **a**, Mean whole network population activity for example RNN over 100 trials after training. Activity under neuromodulation (blue) is not simple transform of activity without neuromodulation (black). **b**, Difference of mean activity with and without neuromodulation (Δactivity) on + vs ∅ stimulus trials for individual neurons. Each color represents an independently trained RNN (10 colors total). Points representing simple scaling of neural activity under neuromodulation would lie on dotted diagonal line.

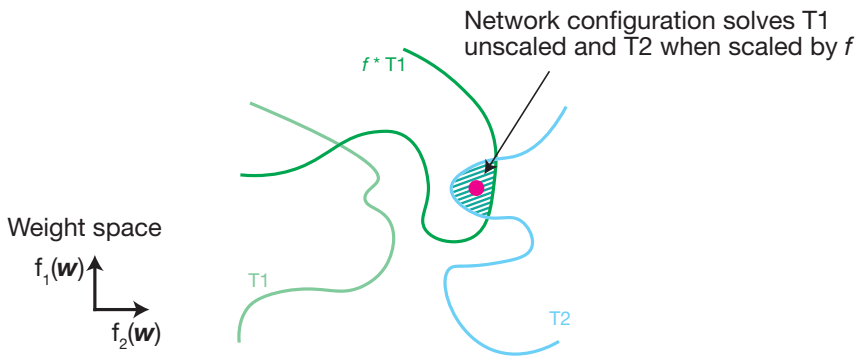


Figure 4.8. Relationship between neuromodulated weight configurations. For the contradictory end behaviors in response to shared stimuli, as in the modified Go-NoGo task, a single network without neuromodulation cannot simultaneously learn both tasks as depicted in this schematic by the lack of overlap between weight space that solves task 1 (T1) and task 2 (T2). By scaling weights in the network by a factor f, neuromodulation allows overlap between the $f \cdot$T1 and T2 spaces. The network solves task 1 when weights are unscaled (T1), and task 2 when weights are scaled ($f \cdot \text{T1} \cap \text{T2}$).

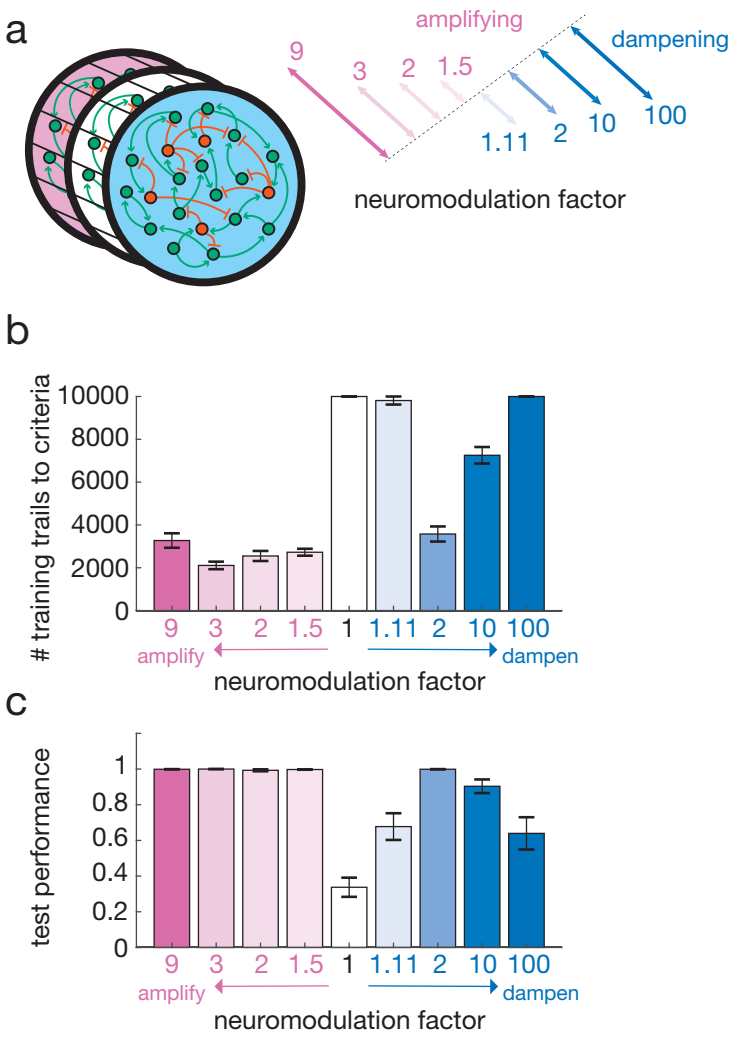


Figure 4.9. Neuromodulation weight scaling mechanism works over a range of factors. a, A range of neuromodulation factors were tested on the modified Go-NoGo task. Factors were applied to weights initialized as described in *Materials and Methods*. **b**, All amplifying factors tested supported task learning in a similar number of training trials. Dampening factors that were either too small (e.g. 1.11) or too large (e.g. 100) led to longer training. **c**, All amplifying factors tested had perfect task performance. Dampening factors that were either too small or too large led to impaired performance, though better than without neuromodulation (factor=1). Extreme strong dampening effectively silences all transmission between neurons, impairing information flow in the network. Too little scaling (e.g. 1.11 factor dampening) did not create enough separation to distinguish the behaviors.

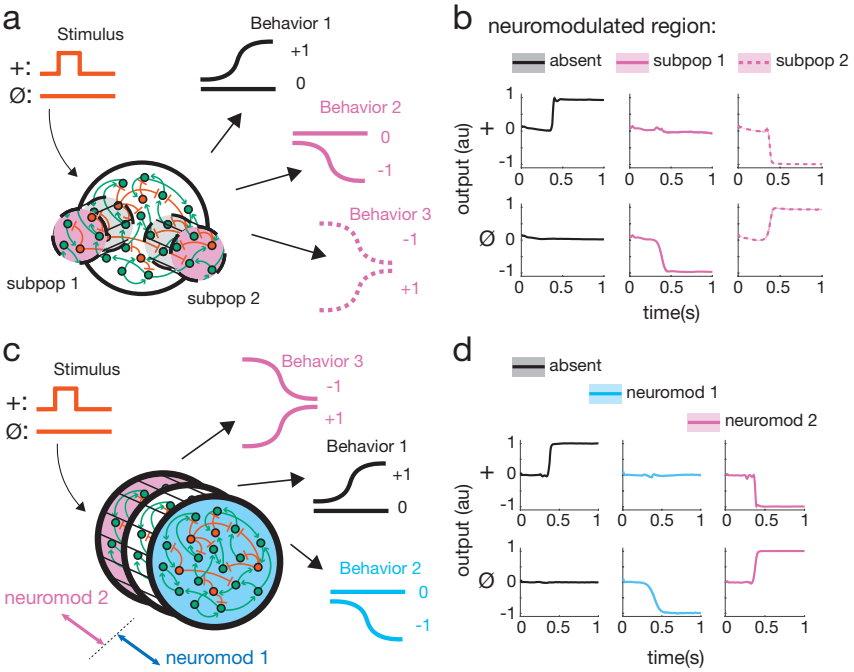


Figure 4.10. Multi-neuromodulator RNNs support multiple behaviors. **a**, 3-behavior modified Go-NoGo. Neuromodulation (factor (f_{nm})=2.5) of either subpopulations ("subpop") within a RNN (subpop1, subpop2) unlock unique behaviors (Behavior 2 (+ stimulus $\rightarrow$ 0 output, $\emptyset \rightarrow$ -1), Behavior 3 (+ $\rightarrow$ -1, $\emptyset \rightarrow$ +1), respectively). **b**, Multi-subpop targeted RNN from **a** successfully learns task. Top row: RNN output to + stimulus when different subpops are neuromodulated; bottom row: same for $\emptyset$ stimulus. **c**, Same task where different levels of neuromodulation (neuromod 1 (f_{nm}=0.5), neuromod 2 (f_{nm}=1.5)) are applied to whole RNN. **d**, Multi-neuromodulator RNN successfully learns task.

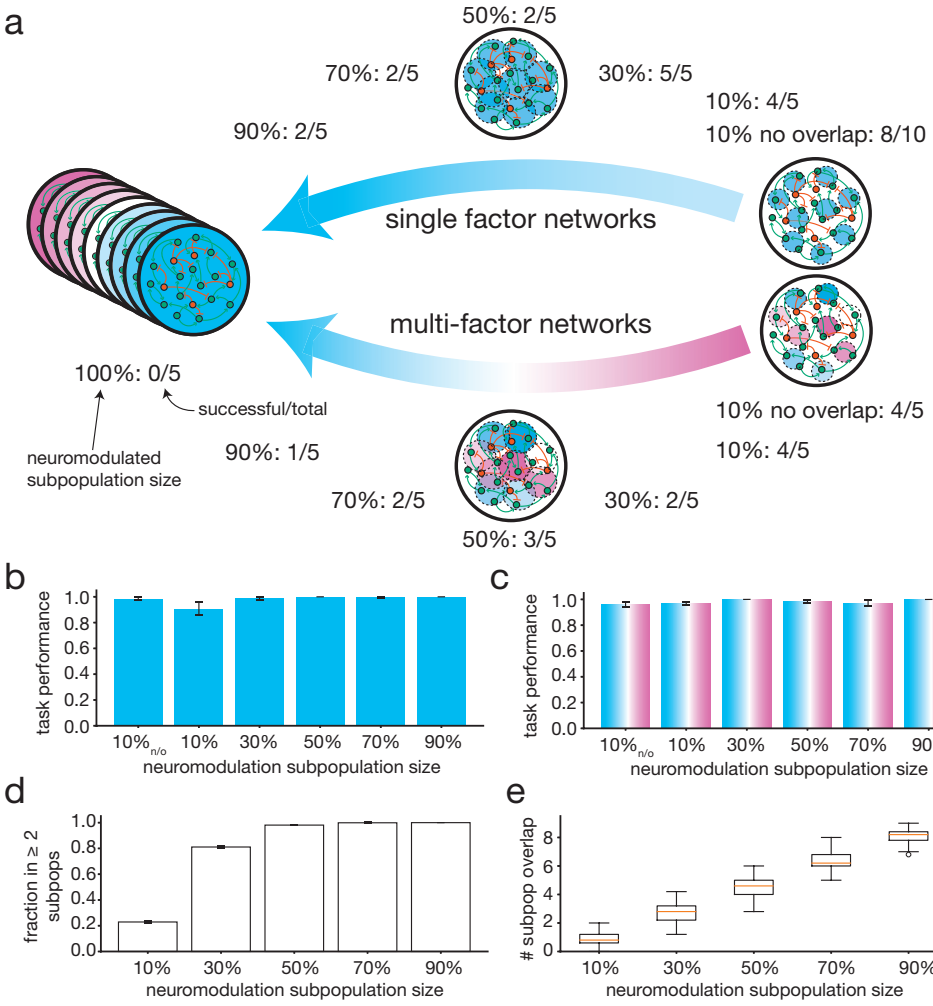


Figure 4.11. Spectrum of neuromodulation subpopulation size and factor. **a**, Different conformations of neuromodulation of a single network support multi-behavior task (9-behavior modified Go-NoGo). Neuromodulation with a single factor (e.g. $f_{nm} = 2.5$) of non-overlapping and overlapping subpopulations across the spectrum of sizes could learn the full 9-behavior task, with larger overlapping subpopulations less consistently learning the full task ("successful/total" refers to independent networks that achieved successful training loss criteria over total attempted). Neuromodulation of the full network with different factors ($f_{nm} \in [1{:}1{:}9]$) consistently supported the 3 behavior task (5/5 successful/total), but not >3 behavior tasks (4-behavior 0/5; 5-behavior 0/5; 9-behavior 0/5 successful/total). Subpopulation neuromodulation with different factors ($f_{nm} \in [1{:}1{:}9]$) could also learn the 9-behavior task with overlapping and non-overlapping subpopulations, similarly exhibiting less consistent learning with larger overlapping subpopulations. **b**, Single factor networks test performance on 9-behavior task across networks that achieved training loss criteria. **c**, Same as **b** but for multi-factor networks. **d**, Fraction of neurons neuromodulated for $\geq$2 conditions across range of neuromodulated subpopulation sizes. **e**, Mean number of conditions a neuron was neuromodulated for across range of subpopulation sizes. For **d**,**e**, 5 replicates per condition. All error bars are SEM.

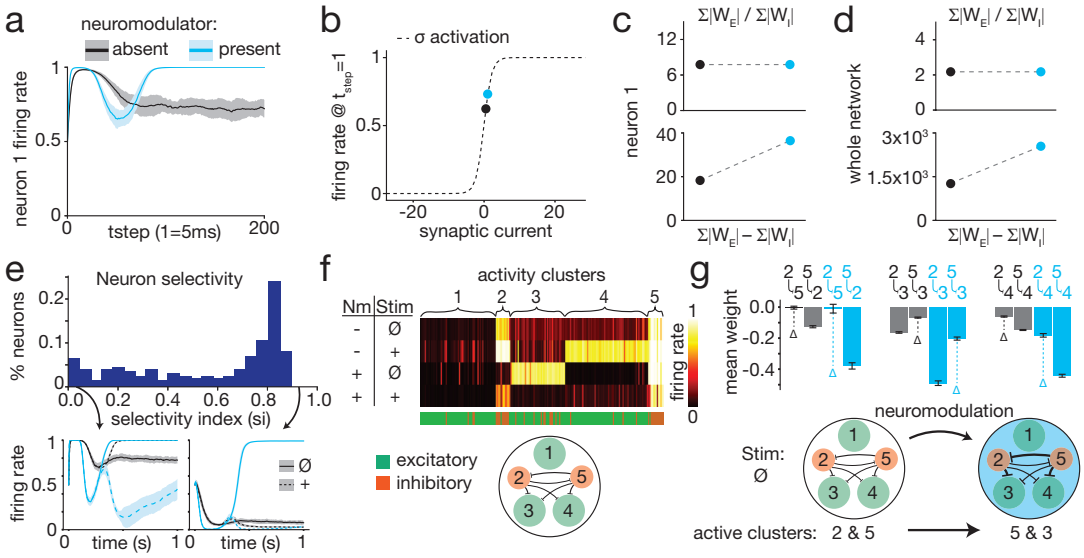


Figure 4.12. E-I difference scaling drives differential activity patterns. **a**, Differential response of an individual neuron (neuron 1) in a whole-network neuromodulated RNN to same stimulus depending on neuromodulation presence. **b**, At the start of a trial, neuromodulation causes neuron 1 to receive different synaptic current input, shifting its firing rate. **c**, The different synaptic input under neuromodulation occurs due to amplification of the net difference in incoming excitatory and inhibitory weights; E/I balance is unchanged. **d**, E-I difference across the whole network is also amplified; E/I remains unaltered. **e**, Though all neurons in the RNN are influenced by the same neuromodulation, some exhibited activity selective for particular stimulus-neuromodulation combinations (high si; example neuron bottom right, si=0.89; see *Materials and Methods*); others were less selective (low si; example neuron bottom left, si=0.01). **f**, Neurons formed 5 clusters—2 predominantly inhibitory, 3 predominantly excitatory—whose activity coded for different stimulus-neuromodulation combinations. Each column of the heatmap is the mean firing rate of an individual neuron across conditions with excitatory/inhibitory identity labeled below. (Nm: neuromodulation present or absent; Stim: stimulus presented) **g**, Amplification of relative weight differences (Δs of mean weight) between inhibitory clusters drives cluster activity switches under neuromodulation: increased inhibition of clusters 2 and 4 by cluster 5 and disinhibition of cluster 3. In all panels light blue represents modulator present and grey/white modulator absent. Error bars are SEM.

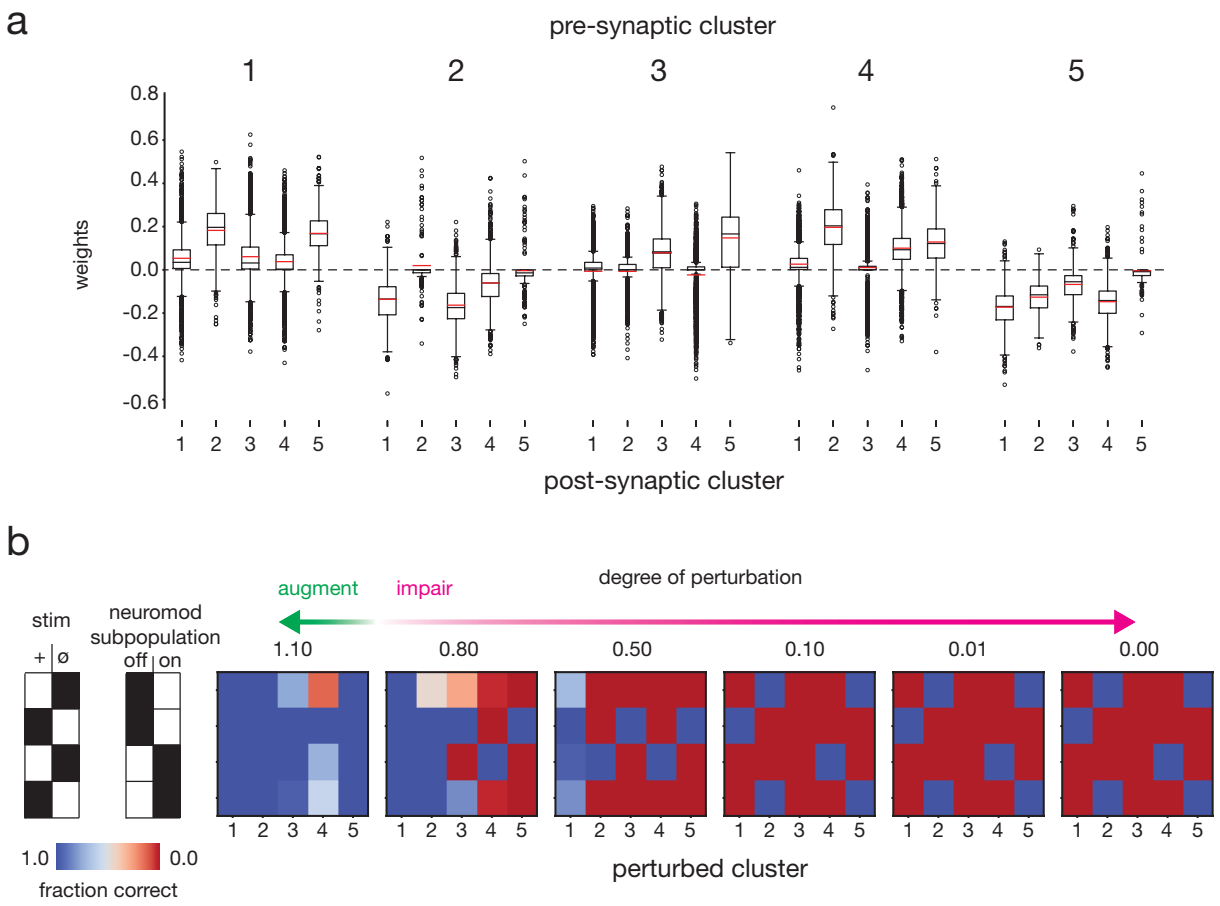


Figure 4.13. Inter-cluster weights and functional clustering for network in *Appendix*, Fig. 4.12. **a**, Weight distribution from each cluster (pre-synaptic) to other clusters (post-synaptic). Clusters 2 and 5 were predominantly composed of inhibitory neurons resulting in inhibitory tone on downstream clusters, while clusters 1, 3, and 4 were predominantly excitatory. **b**, Augmentation (perturbation=1.1x) and impairment (i.e., synaptic lesions; perturbation $\in$ [0.80,0.50,0.10,0.01,0.00]x) of specific clusters led to impairment of performance. For most clusters, increased degree of synaptic impairment led to decreased performance, with higher sensitivity for stimulus-context associated clusters (*Appendix*, Fig. 4.12f; e.g. cluster 4 for + stim, neuromodulation off). In some stimulus-contexts, strong impairment or full lesion of clusters led to recovery of performance (perturbation ≤ 0.10 for clusters 2 and 5 with null stimulus without neuromodulation or positive stimulus with neuromodulation) suggestive of a counter-intuitive switch in underlying cluster dynamics.

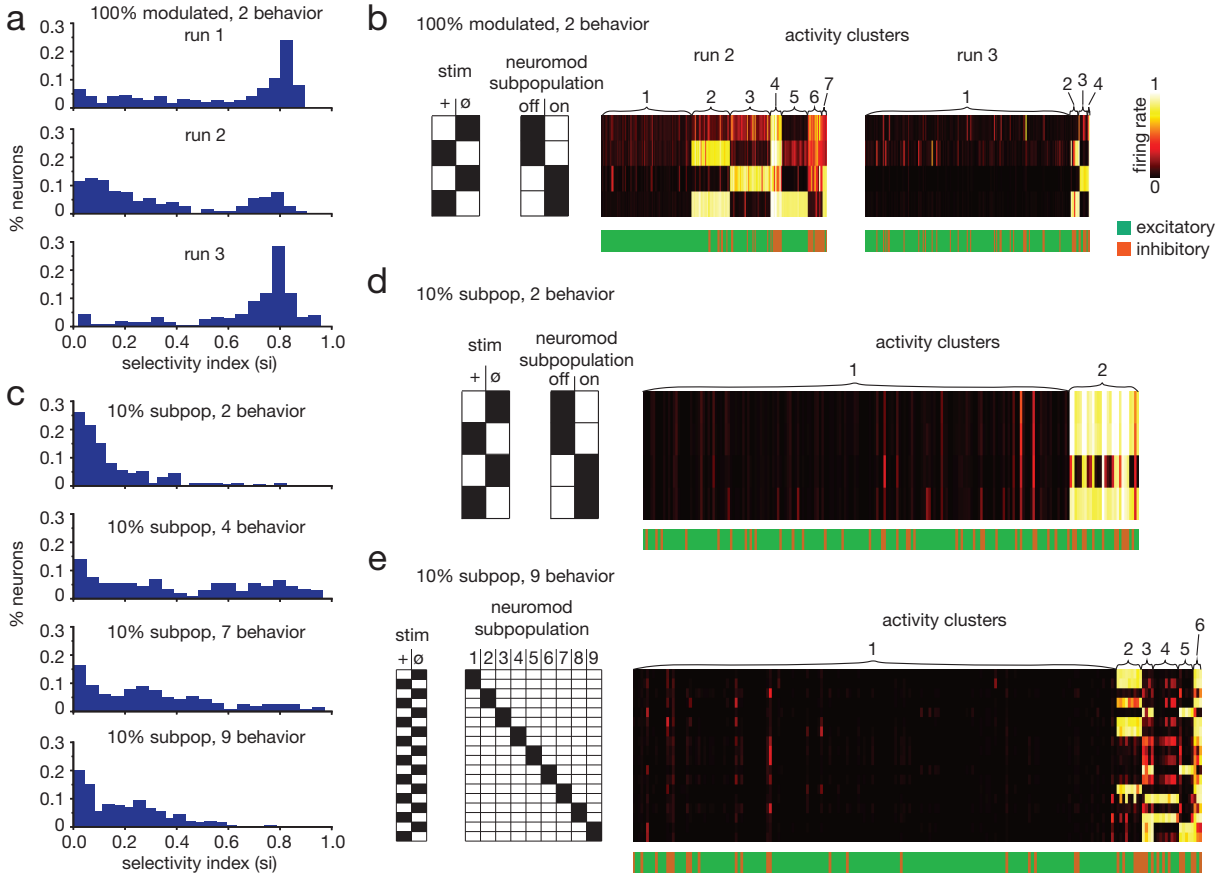


Figure 4.14. Independent networks have unique clustering patterns & different network configurations exhibit less selective neurons and complex, overlapping clustering profiles. **a**, Neural selectivity profiles for 3 networks trained independently on the same condition (100% modulated, 2-behavior). All histograms exhibit high selectivity peak. **b**, Each network from **a** exhibits a unique neural activity clustering pattern including number of clusters and excitatory-inhibitory composition of clusters (see *Appendix*, Fig. 4.12f for run 1 cluster heatmap). **c**, Selectivity index histograms for example networks with different configurations and tasks. Compared to a network in which all neurons were neuromodulated trained on the 2-behavior modified Go-NoGo, network configurations with smaller neuromodulated subpopulations and more behaviors exhibited less selective neurons and more non-selective neurons. **d**, A network with 10% neuromodulated subpopulation trained on the 2-behavior modified Go-NoGo formed 2 activity clusters, showing high overlap across stimulus-context conditions. **e**, The stimulus-context clustering profile for a network with 10% neuromodulated subpopulations (non-overlapping) trained on the 9-behavior modified Go-NoGo task. Cluster profiles are complex and highly overlapping. **d**,**e** cluster heatmaps and neuron type labels are same as **b**.

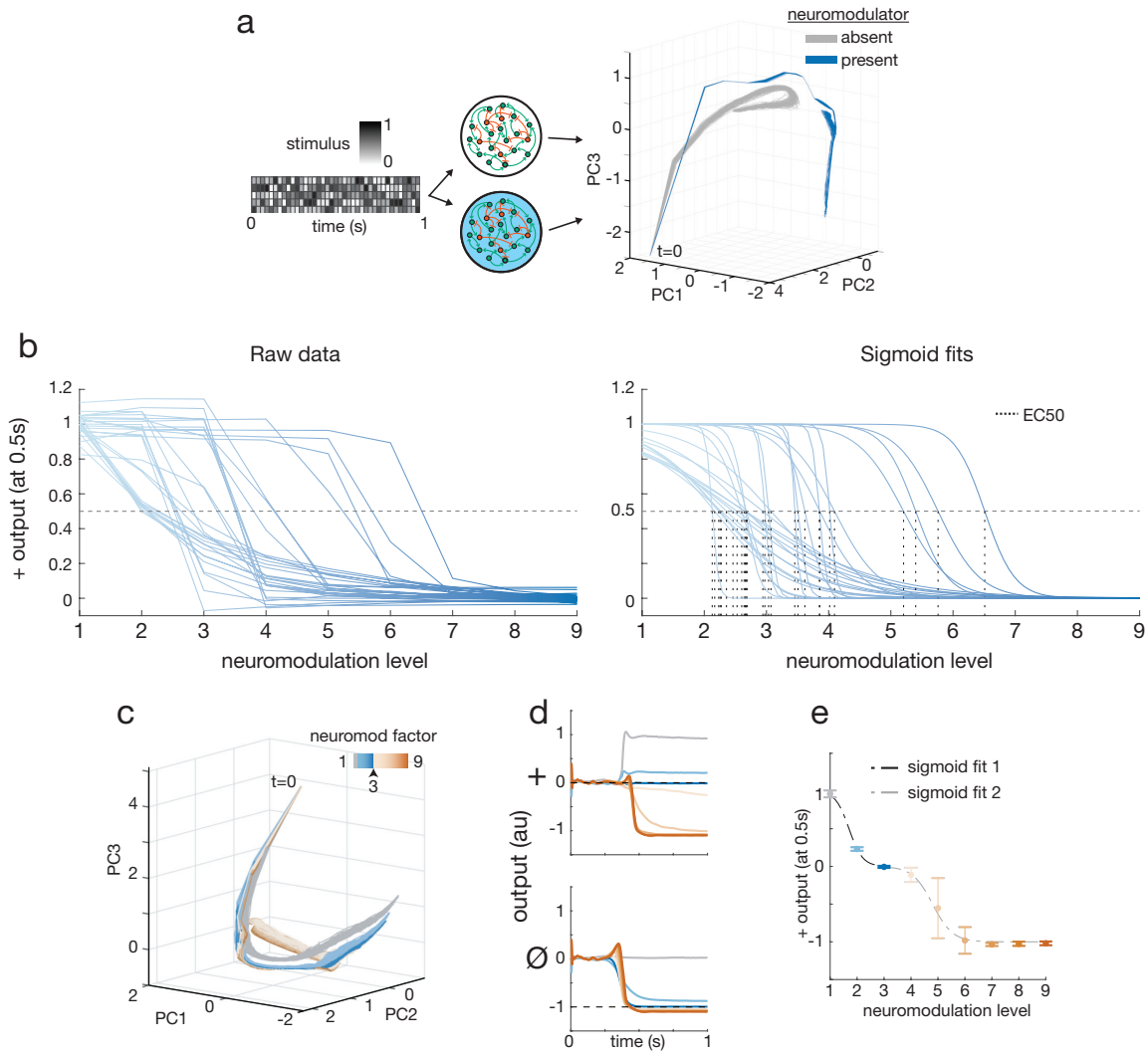


Figure 4.15. Variability in network neuromodulatory transitions. **a**, Shotgun stimulus mapping of activity space inhabited by network in absence and presence of neuromodulation defines separable activity manifolds on which individual trajectories occur. **b**, Left: Output to positive stimulus at time point 0.5s of 29 networks at varying levels of neuromodulation. Each network was independently trained with amplifying neuromodulation factor 9 on the modified Go-NoGo task and then tested at intermediate neuromdulation levels. Right: Sigmoid fits to raw data with EC50 of each curve indicated by dotted vertical line. EC50s ranged from 2.1 to 6.5. Slope of sigmoids (σ_slope) ranged from 0.9 to 26.3. **c**, For + stimulus, excessive neuromodulation pushes network activity into different space. **d**, Over-neuromodulation can drive inappropriate behavior (top) or no change (bottom). **e**, Transition dynamics for overmodulation is best fit by double sigmoid; first fits normal neuromodulation transition (blue) and second fits abberant neuromodulation transition (orange). For all PCA, top 3 PCs accounted for 80–92% of activity variance.

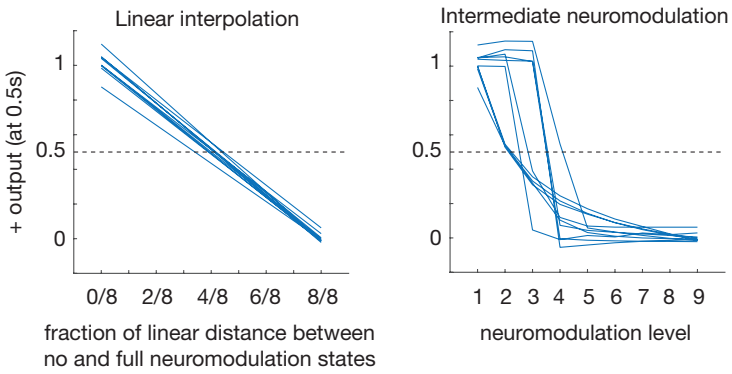
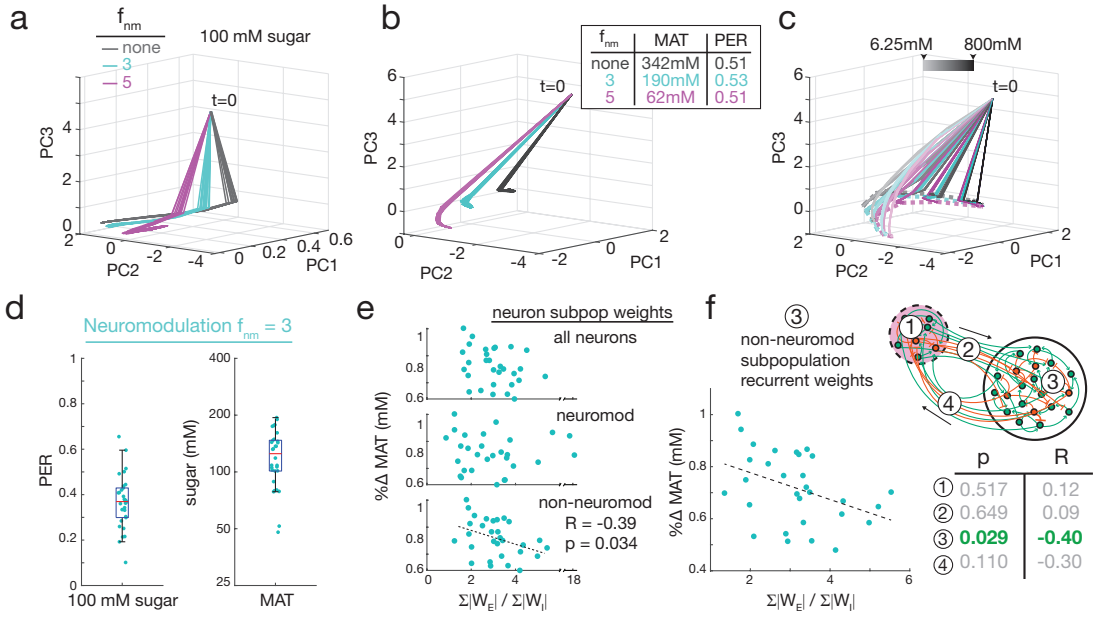


Figure 4.16. Unique geometry of transition manifold in phase space leads to highly variable network sensitivity (EC50). Comparison of outputs for pure linear interpolation between no and full neuromodulation states (left) and intermediate neuromodulation levels (right) for same networks. Linear interpolation (left) gives linear output transition, with all networks tightly following a similar transition and intersecting half-maximal output (0.5) very close to the linear interpolation halfway point (4/8 distance). Intermediate neuromodulation (right) results in nonlinear output transition dynamic with high variability. For clarity, plots show only first 10 simulated networks (out of 29 total).



f_{nm}	MAT	PER
none	342mM	0.51
3	190mM	0.53
5	62mM	0.51

Figure 4.17. RNN models of *Drosophila* exhibit emergent transition behaviors and high variability of neuromodulator sensitivity. **a**, For 100mM sugar, a RNN with intermediate neuromodulation (f_{nm}=3; cyan) generates activity between the neuromodulation extremes (none; grey and f_{nm}=5; magenta). **b**, Activity trajectory at MAT sugar concentration for intermediate neuromodulation of a RNN lies between those for neuromodulation extremes. **c**, Across sugar concentrations, intermediate neuromodulation trajectories (cyan) lay between neuromodulation extremes (grey and magenta), forming a 3-layer activity curtain ending on curved line (dotted lines) defined by trajectory endpoints across the sugar spectrum. **d**, Independently trained RNNs (n=30) exhibited high variability of PERs at 100mM sugar (0.10 to 0.66) and MATs (48 to 194mM). **e**, Normalized MAT change (%ΔMAT) vs E/I ratio for whole RNN and subpopulations. %ΔMAT had significant negative correlation to E/I ratio of the non-neuromodulated neurons ($p<0.05$, R=-0.39). **f**, %ΔMAT E/I ratio correlation was driven by recurrent weights within the non-neuromodulated subpopulation ③, rather than recurrent weights within the neuromodulated subpopulation ①, weights from neuromodulated to non-neuromodulated subpopulation ②, or weights from non-neuromodulated to neuromodulated subpopulation ④ as shown by Pearson correlation p-values (p) and coefficients (R) (for ①,②,④ correlation significance scores shown, scatter plots not shown).

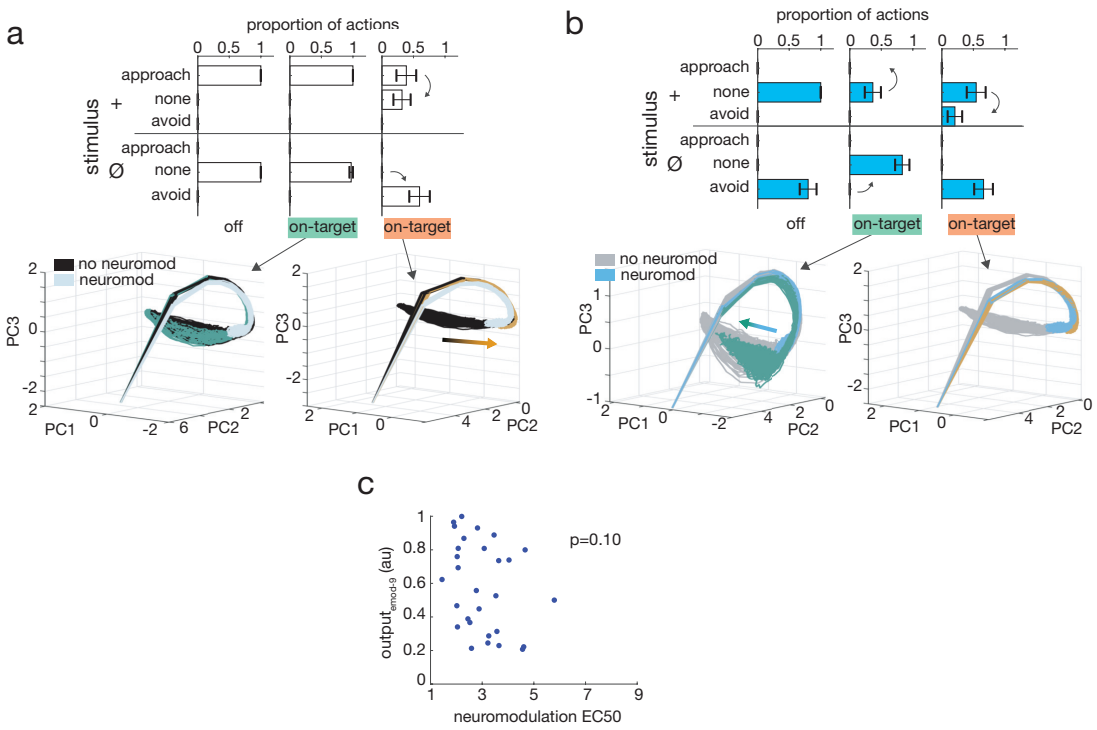


Figure 4.18. Targeted electrical modulation shifts network dynamics through independent circuit effect. a, Behavioral and activity shifts in absence of neuromodulation. Top: behavioral shifts with on-target e-mod. Bottom: activity dynamics under e-mod. Activity trajectories without e-mod are in black (neuromodulated trajectories in light blue for reference); e-mod can shift trajectories off black toward neuromodulated activity space (orange arrow). **b**, Same as **a** with present neuromodulation. Top: behavioral shifts. Bottom: Activity trajectories without e-mod are in blue (non-neuromodulated trajectories in grey for reference); e-mod can shift trajectories off blue toward non-neuromodulated activity space (green arrow). **c**, Some networks did not reach an output level equivalent to half the response of maximal neuromodulation (EC50) even at the highest level of electrical modulation given of -9 units. To assess all networks electrical modulation sensitivity in comparison to neuromodulation sensitivity, network output at maximum electrical modulation is compared to EC50. Like EC50 vs e-mod$_{50}$, there is no significant correlation (p=0.10).

4.6.3 Relationship to isolated gain modulation

Gain modulation changes the slope of the unit activation function (the neuron's "intrinsic excitability") by changing a gain parameter g:

$$r = f(x; g)$$

where r is the firing rate and x is the synaptic current variable. For the sigmoid activation function this corresponds to:

$$r = \frac{1}{1 + e^{-gx}}$$

To see the effective change on the activation function of amplifying or dampening the weights we can compare the effect of gain modulation to weight modulation on the equation that governs neural dynamics:

$$\tau \dot{x}_i = -x_i + \sum_j W_{ji} r_{ji} + W_{ui} u + N(0, 0.1)$$

If we assume no input and no noise, we get a simplified equation describing neuron dynamics:

$$\tau \dot{x}_i = -x_i + \sum_j W_{ji} r_{ji}$$

Gain modulation g gives:

$$\tau \dot{x}_i = -x_i + \sum_j W_{ji} r_{ji}(g)$$

while weight modulation with a neuromodulation factor m gives

$$\tau \dot{x}_i = -x_i + \sum_j m \cdot W_{ji} r_{ji}(1)$$

To compare each type of modulation, we can consider the modified terms:

$$\begin{aligned} 1 \cdot r(g) &= \frac{1}{1+e^{-gx}} && \text{(gain effect)} \\ m \cdot r(1) &= \frac{1}{\frac{1}{m} + \frac{1}{m} \cdot e^{-x}} && \text{(weight effect)} \\ &= \frac{1}{\frac{1}{m} + e^{-x-lnm}} \end{aligned}$$

These effects are equivalent when

$$g = -\frac{ln(\frac{1-m}{m} + e^{-x-lnm})}{x} \tag{4.3}$$

So generally gain modulation, g, is only equivalent to a weight modulation by m if the gain term is precisely the time varying function of x defined by Equation 4.3. Furthermore, for some values of m, there is no equivalent g for certain values of x. E.g. for $m = 2$, g is defined by Equation 4.3 only for $x < 0$. For arbitrary values of x with fixed, constant g and m:

$$r_{ji}(g) \neq m \cdot r_{ji}(1)$$

except when $g = m = 1$, i.e., when there is no modulation. Thus, weight neuromodulation and gain modulation operate through different effects.

4.6.4 Chemical and electrical modulation

Chemical (neuromodulation) and electrical modulation in our network operate in directionally similar manner with qualitatively different effects. This can be seen by inspecting the

equation governing each neuron i's synaptic current variable and thereby the network's activity dynamics:

$$\tau \dot{x}_i = -x_i + \sum_j W_{ji} r_{ji} + W_{ui} u + N(0, 0.1)$$

For any given neuron we can break up the terms by whether an input neuron is in the neuromodulated subpopulation or not:

$$\tau \dot{x}_i = -x_i + \sum_k W_{ki} r_{ki} + \sum_q W_{qi} r_{qi} + W_{ui} u + N(0, 0.1)$$

where k is the index for non-neuromodulated neurons and q is the index for neuromodulated neurons.

Neuromodulation in our model acts by scaling the target neurons outgoing weights by a factor f:

$$\tau \dot{x}_i = -x_i + \sum_k W_{ki} r_{ki} + f \cdot \sum_q W_{qi} r_{qi} + W_{ui} u + N(0, 0.1)$$

Electrical stimulation acts by adding exogenous synaptic current to target neurons. For a given neuron i in the non-neuromodulated subpopulation, electrical stimulation of the neuromodulated subpopulation is felt through altered incoming firing rates:

$$\tau \dot{x}_i = -x_i + \sum_k W_{ki} r_{ki} + \sum_q W_{qi} r_{qi,Estim} + W_{ui} u + N(0, 0.1)$$

and a neuron i in the neuromodulated subpopulation is additionally affected through direct stimulation:

$$\tau \dot{x}_i = -x_i + \sum_k W_{ki} r_{ki} + \sum_q W_{qi} r_{qi,Estim} + W_{ui} u + u_{Estim} + N(0, 0.1)$$

Thus we can see that both chemical and electrical stimulation act through the same term in the equation that governs neural synaptic currents yet in different manners. Neuromodulation directly scales presynaptic weighted inputs from neuromoduated neurons, whereas electrical stimulation acts by altering the firing rate of presynaptic neuromoduated neurons with an additional direct influence on the synaptic current if the neuron of interest is in the electrically stimulation subpopulation.

The similarities of these forms of modulation (acting through the same terms of the synaptic current equation) indicates why they can have similar affects on network output in some circumstances. Nevertheless, the differences in how they affect the synaptic current equation are propagated through the recurrent connections of the network at each time step which drives the distinct dynamical changes seen under chemical versus electrical modulation.

Bibliography

[All+19] William E. Allen, Michael Z. Chen, Nandini Pichamoorthy, Rebecca H. Tien, Marius Pachitariu, Liqun Luo, and Karl Deisseroth. Thirst regulates motivated behavior through modulation of brainwide neural population dynamics. *Science*, 364: 2019, eaav3932.

[AK17] Michael C. Avery and Jeffrey L. Krichmar. Neuromodulatory Systems and Their Interactions: A Review of Models, Theories, and Experiments. *Frontiers in Neural Circuits*, 11: Dec. 2017, 108.

[Bac+20] Julien Bacqué-Cazenave, Rahul Bharatiya, Grégory Barrière, Jean-Paul Delbecque, Nouhaila Bouguiyoud, Giuseppe Di Giovanni, Daniel Cattaert, and Philippe De Deurwaerdère. Serotonin in Animal Cognition and Behavior. *International Journal of Molecular Sciences*, 21: 2020, 1649.

[Bea+20] Shawn Beaulieu, Lapo Frati, Thomas Miconi, Joel Lehman, Kenneth O. Stanley, Jeff Clune, and Nick Cheney. Learning to Continually Learn. *24th European Conference on Artificial Intelligence (ECAI)*. Ed. by Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarın, and Jérôme Lang. Vol. 325. Santiago de Compostela, Spain: IOS Press, 2020, 992–1001.

[Bot07] Matthew M. Botvinick. Multilevel structure in behaviour and in the brain: a model of Fuster's hierarchy. *Philosophical Transactions of the Royal Society B*, 362: 2007, 1615–1626.

[Bur+17] Anthony Burgos-Robles, Eyal Y Kimchi, Ehsan M Izadmehr, Mary Jane Porzenheim, William A Ramos-Guasp, Edward H Nieh, Ada C Felix-Ortiz, Praneeth Namburi, Christopher A Leppla, Kara N Presbrey, Kavitha K Anandalingam, Pablo A Pagan-Rivera, Melodi Anahtar, Anna Beyeler, and Kay M Tye. Amygdala inputs to prefrontal cortex guide behavior amid conflicting cues of reward and punishment. *Nature Neuroscience*, 20: Apr. 2017, 824–835.

[CS14] Jeremie Cabessa and Hava T. Siegelmann. The Super-Turing Computational Power of Plastic Recurrent Neural Networks. *International Journal of Neural Systems*, 24(8): 2014, 1450029.

[CPM19] Adam J. Calhoun, Jonathan W. Pillow, and Mala Murthy. Unsupervised identification of the internal states that shape natural behavior. *Nature Neuroscience*, 22: 2019, 2040–2049.

[Cer+20] Nathan Cermak, Stephanie K. Yu, Rebekah Clark, Yung-Chi Huang, Saba N. Baskoylu, and Steven W. Flavell. Whole-organism behavioral profiling reveals a role for dopamine in state-dependent motor program coupling in C. elegans. *eLife*, 9: 2020, e57093.

[CY14] John P Cunningham and Byron M Yu. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17: 2014, 1500–1508.

[Dei15] Karl Deisseroth. Optogenetics: 10 years of microbial opsins in neuroscience. *Nature Neuroscience*, 18: 2015, 1213–1225.

[Fur+19] Toshi A Furukawa, Andrea Cipriani, Philip J Cowen, Stefan Leucht, Matthias Egger, and Georgia Salanti. Optimal dose of selective serotonin reuptake inhibitors, venlafaxine, and mirtazapine in major depression: a systematic review and dose-response meta-analysis. *Lancet Psychiatry*, 6: July 2019, 601–609.

[GB18] Vishwa Goudar and Dean V Buonomano. Encoding sensory and motor patterns as time-invariant trajectories in recurrent neural networks. *eLife*, 7: 2018, e31134.

[Gou+19] Flavia V. Gouveia, Darryl C. Gidyk, Peter Giacobbe, Enoch Ng, Ying Meng, Benjamin Davidson, Agessandro Abrahao, Nir Lipsman, and Clement Hamani. Neuromodulation Strategies in Post-Traumatic Stress Disorder: From Preclinical Models to Clinical Applications. *Brain Sciences*, 9: Feb. 2019, 45.

[HM18] Sara A. Haddad and Eve Marder. Circuit Robustness to Temperature Perturbation Is Altered by Neuromodulators. *Neuron*: 2018.

[HJ10] Ronald M. Harris-Warrick and Bruce R. Johnson. Checks and balances in neuromodulation. *Frontiers in Behavioral Neuroscience*, 4: July 2010, 47.

[Has06] Michael E. Hasselmo. The role of acetylcholine in learning and memory. *Curr Opin Neurobiol.*, 16: 2006, 710–715.

[HSB95] Michael E. Hasselmo, Eric Schnell, and Edi Barkai. Dynamics of Learning and Recall at Excitatory Recurrent Synapses and Cholinergic Modulation in Rat Hippocampal Region CA3. *Journal of Neuroscience*, 15: July 1995, 5249–5262.

[HP14] Michael J Higley and Marina R Picciotto. Neuromodulation by acetylcholine: examples from schizophrenia and depression. *Current Opinion in Neurobiology*, 29: 2014, 88–95.

[Ina+12] Hidehiko K. Inagaki, Shlomo Ben-Tabou de-Leon, Allan M. Wong, Smitha Jagadish, Hiroshi Ishimoto, Gilad Barnea, Toshihiro Kitamoto, Richard Axel, and David J. Anderson. Visualizing Neuromodulation In Vivo: TANGO-Mapping of Dopamine Signaling Reveals Appetite Control of Sugar Sensing. *Cell*, 148: Feb. 2012, 583–595.

[JGW09] Jian Jing, Rhanor Gillette, and Klaudiusz R. Weiss. Evolving Concepts of Arousal: Insights from Simple Model Systems. *Reviews in the Neurosciences*, 20: 2009, 405–427.

[Joh+11] Bruce R. Johnson, Jessica M. Brown, Mark D. Kvarta, Jay Y. J. Lu, Lauren R. Schneider, Farzan Nadim, and Ronald M. Harris-Warrick. Differential Modulation of Synaptic Strength and Timing Regulate Synaptic Efficacy in a Motor Network. *Journal of Neurophysiology*, 105: 2011, 293–304.

[JPH95] Bruce R. Johnson, Jack H. Peck, and Ronald M. Harris-Warrick. Distributed Amine Modulation of Graded Chemical Transmission in the Pyloric Network of the Lobster Stomatogastric Ganglion. *Journal of Neurophysiology*, 74: July 1995, 437–452.

[KE99] Paul S. Katz and Donald H. Edwards. "Metamodulation: the control and modulation of neuromodulation". *Beyond Neurotransmission: Neuromodulation and its Importance for Information Processing*. Ed. by Paul S. Katz. Oxford, UK: Oxford University Press, 1999. Chap. 10, 349–381.

[KGF94] Paul S. Katz, Peter A. Getting, and William N. Frost. Dynamic neuromodulation of synaptic strength intrinsic to a central pattern generator circuit. *Nature*, 367: Feb. 1994, 729–731.

[KLS19] Robert Kim, Yinghao Li, and Terrence J Sejnowski. Simple framework for constructing functional spiking recurrent neural networks. *PNAS*, 116(45): 2019, 22811–22820.

[KS20] Robert Kim and Terrence J. Sejnowski. Strong inhibitory signaling underlies stable temporal dynamics and working memory in spiking neural networks. *Nature Neuroscience*, 24: 2020.

[Kra+21] Joachim K. Krauss, Nir Lipsman, Tipu Aziz, Alexandre Boutet, Peter Brown, Jin Woo Chang, Benjamin Davidson, Warren M. Grill, Marwan I. Hariz, Andreas Horn, Michael Schulder, Antonios Mammis, Peter A. Tass, Jens Volkmann, and Andres M. Lozano. Technology of deep brain stimulation: current status and future directions. *Nature Reviews Neurology*, 17: 2021, 75–87.

[LB13] Rodrigo Laje and Dean V Buonomano. Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nature Neuroscience*, 16(7): 2013, 925–933.

[LBN18] Xinping Li, Dirk Bucher, and Farzan Nadim. Distinct Co-Modulation Rules of Synapses and Voltage-Gated Currents Coordinate Interactions of Multiple Neuromodulators. *Journal of Neuroscience*, 38: Oct. 2018, 8549–8562.

[Low+18] Ryan J. Low, Sam Lewallen, Dmitriy Aronov, Rhino Nevers, and David W. Tank. Probing variability in a cognitive map using manifold inference from neural dynamics. *bioRxiv*: 2018. URL: https://doi.org/10.1101/418939.

[Mah+19] Niru Maheswaranathan, Alex H. Williams, Matthew D. Golub, Surya Ganguli, and David Sussillo. Universality and individuality in neural dynamics across large populations of recurrent networks. *Advances in Neural Information Processing Systems 32, NeurIPS*. 2019.

[Mar12] Eve Marder. Neuromodulation of Neuronal Circuits: Back to the Future. *Neuron*, 76: Oct. 2012, 1–11.

[ME84] Eve Marder and Judith S. Eisen. Electrically Coupled Pacemaker Neurons Respond Differently to Same Physiological Inputs and Neurotransmitters. *Journal of Neurophysiology*, 51: June 1984, 1362–1374.

[MNH20] David A. McCormick, Dennis B. Nestvogel, and Biyu J. He. Neuromodulation of Brain State and Behavior. *Annual Review of Neuroscience*, 43: Apr. 2020, 391–415.

[Meh+20] Johannes Mehrer, Courtney J. Spoerer, Nikolaus Kriegeskorte, and Tim C. Kietzmann. Individual differences among deep neural network models. *Nature Communications*, 11: 2020, 5725.

[Mic+19] Thomas Miconi, Aditya Rawal, Jeff Clune, and Kenneth O. Stanley. Backpropamine: training self-modifying neural networks with differentiable neuromodulated plasticity. *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. 2019.

[NB12] Farzan Nadim and Dirk Bucher. Neuromodulation of neurons and synapses. *Current Opinion in Neurobiology*, 29: 2012, 48–56.

[New+12] Ehren L. Newman, Kishan Gupta, Jason R. Climer, Caitlin K. Monaghan, and Michael E. Hasselmo. Cholinergic modulation of cognitive processing: insights drawn from computational models. *Frontiers in Behavioral Neuroscience*, 6: June 2012, 24.

[Nie+21] Edward H. Nieh, Manuel Schottdorf, Nicolas W. Freeman, Ryan J. Low, Sam Lewallen, Sue Ann Koay, Lucas Pinto, Jeffrey L. Gauthier, Carlos D. Brody, and David W. Tank. Geometry of abstract learned knowledge in the hippocampus. *Nature*: 2021.

[Pat+18] Tommaso Patriarchi, Jounhong Ryan Cho, Katharina Merten, Mark W. Howe, Aaron Marley, Wei-Hong Xiong, Robert W. Folk, Gerard Joey Broussard, Ruqiang Liang, Min Jee Jang, Haining Zhong, Daniel Dombeck, Mark von Zastrow, Axel Nimmerjahn, Viviana Gradinaru, John T. Williams, and Lin Tian. Ultrafast neuronal imaging of dopamine dynamics with designed genetically encoded sensors. *Science*, 360: June 2018, eaat4422.

[Pro+19] Nicole R. Provenza, Evan R. Matteson, Anusha B. Allawala, Adriel Barrios-Anderson, Sameer A. Sheth, Ashwin Viswanathan, Elizabeth McIngvale, Eric A. Storch, Michael J. Frank, Nicole C. R. McLaughlin, Jeffrey F. Cohn, Wayne K. Goodman, and David A. Borton. The Case for Adaptive Neuromodulation to Treat Severe Intractable Mental Disorders. *Frontiers in Neuroscience*, 13: Feb. 2019, 152.

[RF18] Gabriele Radnikow and Dirk Feldmeyer. Layer- and Cell Type-Specific Modulation of Excitatory Neuronal Activity in the Neocortex. *Frontiers in Neuroanatomy*, 12: 2018.

[RS10] J. A. Ribeiro and A. M. Sebastiao. Modulation and metamodulation of synapses by adenosine. *Acta Physiologica*, 199: 2010, 161–169.

[Ris+20] Luisa De Risio, Marta Borgi, Mauro Pettorruso, Andrea Miuli, Angela Maria Ottomana, Antonella Sociali, Giovanni Martinotti, Giuseppe Nicolo, Simone Macri, Massimo di Giannantonio, and Francesca Zoratto. Recovering from depression with repetitive transcranial magnetic stimulation (rTMS): a systematic review and meta-analysis of preclinical studies. *Translational Psychiatry*, 10: 2020, 393.

[Rus+18] Abigail A. Russo, Sean R. Bittner, Sean M. Perkins, Jeffrey S. Seely, Brian M. London, Antonio H. Lara, Andrew Miri, Najja J. Marshall, Adam Kohn, Thomas M. Jessell, Laurence F. Abbott, John P. Cunningham, and Mark M. Churchland. Motor Cortex Embeds Muscle-like Commands in an Untangled Population Response. *Neuron*, 97: 2018, 953–966.

[SYW16] H. Francis Song, Guangyu R. Yang, and Xiao-Jing Wang. Training Excitatory-Inhibitory Recurrent Neural Networks for Cognitive Tasks: A Simple and Flexible Framework. *PLOS Computational Biology*, 12: 2016, 1–30.

[Str+18] Jake P. Stroud, Mason A. Porter, Guillaume Hennequin, and Tim P. Vogels. Motor primitives in space and time via targeted gain modulation in cortical networks. *Nature Neuroscience*, 21: Dec. 2018, 1774–1783.

[Tru+20] Cleber A Trujillo, Jason W Adams, Priscilla D Negraes, Cassiano Carromeu, Leon Tejwani, Allan Acab, Ben Tsuda, Charles A Thomas, Neha Sodhi, Katherine M Fichter, Sarah Romero, Fabian Zanella, Terrence J Sejnowski, Henning Ulrich, and Alysson R Muotri. Pharmacological reversal of synaptic and network pathology in human MECP2-KO neurons and cortical organoids. *EMBO Molecular Medicine*, 13: 2020, e12523.

[Vec+20] Nicolas Vecoven, Damien Ernst, Antoine Wehenkel, and Guillaume Drion. Introducing neuromodulation in deep neural networks to learn adaptive behaviors. *Plos One*, 15: 2020, e0227922.

[WM14] Jason C Wester and Chris J McBain. Behavioral state-dependent modulation of distinct interneuron subtypes and consequences for circuit function. *Current Opinion in Neurobiology*, 29: 2014, 118–125.

[Xie+21] Peng Xie, Lijuan Liu, Xiuli Kuang, Yimin Wang, Lei Qu, Hui Gong, Shengdian Jiang, Anan Li, Zongcai Ruan, Liya Ding, Zizhen Yao, Chao Chen, Mengya Chen, Tanya L. Daigle, Rachel Dalley, Zhangcan Ding, Yanjun Duan, Aaron Feiner, Ping He, Chris Hill, Karla E. Hirokawa, Guodong Hong, Lei Huang, Sara Kebede, Hsien-

Chi Kuo, Rachael Larsen, Phil Lesnar, Longfei Li, Qi Li, Xiangning Li, Yaoyao Li, Yuanyuan Li, An Liu, Donghuan Lu, Stephanie Mok, Lydia Ng, Thuc Nghi Nguyen, Qiang Ouyang, Jintao Pan, Elise Shen, Yuanyuan Song, Susan M. Sunkin, Bosiljka Tasic, Matthew B. Veldman, Wayne Wakeman, Wan Wan, Peng Wang, Quanxin Wang, Tao Wang, Yaping Wang, Feng Xiong, Wei Xiong, Wenjie Xu, Min Ye, Lulu Yin, Yang Yu, Jia Yuan, Jing Yuan, Zhixi Yun, Shaoqun Zeng, Shichen Zhang, Sujun Zhao, Zijun Zhao, Zhi Zhou, Z. Josh Huang, Luke Esposito, Michael J. Hawrylycz, Staci A. Sorensen, X. William Yang, Yefeng Zheng, Zhongze Gu, Wei Xie, Christof Koch, Qingming Luo, Julie A. Harris, Yun Wang, and Hongkui Zeng. Morphological diversity of single neurons in molecularly defined cell types. *Nature*, 598: 2021, 174–181.

[YSY14] Ken-ichi Yamamoto, Toshikazu Shinba, and Mitsunobu Yoshii. Psychiatric symptoms of noradrenergic dysfunction: A pathophysiological view. *Psychiatry and Clinical Neurosciences*, 68: 2014, 1–20.

[YD05] Angela J. Yu and Peter Dayan. Uncertainty, Neuromodulation, and Attention. *Neuron*, 46: May 2005, 681–692.

[Zha+21] Stephen X. Zhang, Andrew Lutas, Shang Yang, Adriana Diaz, Hugo Fluhr, Georg Nagel, Shiqiang Gao, and Mark L. Andermann. Hypothalamic dopamine neurons motivate mating through persistent cAMP signalling. *Nature*, 597: 2021, 245–249.

Chapter 5

Conclusions

The brain is a memory machine with many features and levels of complexity built upon each other, seamlessly integrated through millions of years of evolution. From the interaction of molecules, cells, networks, and systems emerge colorful behaviors, vivid memories across large time spans, abstract reasoning, and consciousness. These emergent properties can only be understood by studying those interactions: networks made up of cells governed by molecules, aggregated and fused into systems that create an integrated single central nervous system. In this dissertation a small contribution to this effort is made. Primarily focusing on the prefrontal areas of the brain, I studied how systems of circuits coordinate (Chapter 2), how the circuits upon which those systems are built can learn and develop (Chapter 3), and how that process depends on the modulatory effects of molecules acting in time and space (Chapter 4).

In each chapter future directions were discussed, yet other exciting future work reaches across the levels of investigation considered in each chapter. Revealing how strategies develop and evolve within circuit components (Chapter 3) set within a hierarchical system (Chapter 2) and how this is influenced by both the architecture and importantly disruptions (e.g. lesions) will provide insight and predictions for how PFC subregions and related regions learn together. Implementing simulations of learning impairments (e.g. decreased learning rate or impaired synaptic transmission) in this context and how they affect the system's flexibility, inductive transfer, and memory savings will be important for both understanding and possible treatment

options for disorders that affect prefrontal learning like autism spectrum disorder and Rett syndrome. Studies of how neuromodulators can regulate learning and coordinate multiple behaviors (Chapter 4) in this setting will relate findings to clinical and experimental observations of neuromodulatory action within the PFC. These latter investigations would shed light on unknowns like the role of diurnal and seasonal fluctuations of neuromodulator levels in mood disorders like seasonal affective disorder [Tyr+16; Mat+15], and the role of dopamine in working memory impairments in schizophrenia [Bra+21].

To develop effective treatments for neurological and psychiatric illnesses, we must understand basic properties of how the brain operates across levels. The studies in this dissertation and others begin to build the multi-level foundation necessary for unraveling deep mysteries of memory and learning in the brain, of which hints are accruing. Coordination of multiple neuromodulators (Chapter 4; [Pri+21]) acting on a specialized neural network architecture[1] suggest the basis of how episodic memory may operate. Emergent representations that vary in abstraction across circuits in hierarchical systems (Chapter 2; [Bot07]), attentional mechanisms [IKN98; Vas+17], and dynamic components supporting use of past information and prediction of future occurrences (Chapter 2; [Fri05]) hint at some pieces of consciousness. Future studies built on these foundations suggest discoveries on the horizon and hope for urgently needed new treatments for diseases of the brain.

[1]The specialized architecture of the hippocampus is the focus of our study: Tsuda B, Sejnowski TJ. The hippocampus as a spatiotemporal autoencoder. *In preparation*, 2022

Bibliography

[Bot07] Matthew M. Botvinick. Multilevel structure in behaviour and in the brain: a model of Fuster's hierarchy. *Philosophical Transactions of the Royal Society B*, 362: 2007, 1615–1626.

[Bra+21] Urs Braun, Anais Harneit, Giulio Pergola, Tommaso Menara, Axel Schäfer, Richard F. Betzel, Zhenxiang Zang, Janina I. Schweiger, Xiaolong Zhang, Kristina Schwarz, Junfang Chen, Giuseppe Blasi, Alessandro Bertolino, Daniel Durstewitz, Fabio Pasqualetti, Emanuel Schwarz, Andreas Meyer-Lindenberg, Danielle S. Bassett, and Heike Tost. Brain network dynamics during working memory are modulated by dopamine and diminished in schizophrenia. *Nature Communications*, 12: 2021, 3478.

[Fri05] Karl Friston. A theory of cortical responses. *Philosophical Transactions of the Royal Society B*, 360: 2005, 815–836.

[IKN98] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11): 1998, 1254–1259.

[Mat+15] Granville J Matheson, Martin Schain, Rita Almeida, Johan Lundberg, Zsolt Cselényi, Jacqueline Borg, Andrea Varrone, Lars Farde, and Simon Cervenka. Diurnal and seasonal variation of the brain serotonin system in healthy male subjects. *Neuroimage*, 112: 2015, 225–231.

[Pri+21] Luke Y. Prince, Travis Bacon, Rachel Humphries, Krasimira Tsaneva-Atanasova, Claudia Clopath, and Jack R. Mellor. Separable actions of acetylcholine and noradrenaline on neuronal ensemble formation in hippocampal CA3 circuits. *PLOS Computational Biology*, 17(10): Oct. 2021, 1–37.

[Tyr+16] Andrea E Tyrer, Robert D Levitan, Sylvain Houle, Alan A Wilson, José N Nobrega, and Jeffrey H Meyer. Increased Seasonal Variation in Serotonin Transporter Binding in Seasonal Affective Disorder. *Neuropsychopharmacology*, 41: 2016, 2447–2454.

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017.