

# Adding to and building up very small nervous systems

A DISSERTATION PRESENTED  
BY  
CHENGUANG LI  
TO  
THE DEPARTMENT OF BIOPHYSICS

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY  
IN THE SUBJECT OF  
BIOPHYSICS

HARVARD UNIVERSITY  
CAMBRIDGE, MASSACHUSETTS  
MAY 2024

©2024 – CHENGUANG LI  
ALL RIGHTS RESERVED.

## Adding to and building up very small nervous systems

### ABSTRACT

This dissertation asks and tries to answer two questions: first, how might one add to the living nervous system of a small animal using artificial methods? Second, how might one design an algorithm that reproduces some of the hallmark characteristics of natural intelligence, many of which are still missing from current artificial neural networks?

To address the first question, we present an approach that integrates deep reinforcement learning agents with the nervous system of the model organism *Caenorhabditis elegans*, a nematode with 302 neurons. We integrate the artificial and biological networks using optogenetics, and design our artificial agent to navigate animals to given targets. We find that agents can learn appropriate strategies for different sets of neurons, even when neuronal roles in behavior are very different from each other. We show several possible applications of the RL-*C. elegans* system, including mapping out neural policies that are sufficient to drive target behaviors, studying the behavior of RL agents in biologically relevant environments, and accomplishing goals that utilize the strengths of both artificial and biological intelligences.

To answer the second question, we consider the view of nervous systems that models behaviors and computations as emergent properties of many small interacting components. We combine this emergent view with three other strongly supported features of all nervous systems: that neural functions heavily rely on predictive principles, that neurons are noisy, and that many diverse and fundamental computations in the brain are implemented via attractor dynamics. The resultant model uses only local prediction and noise updates, and yet can seek out and optimize reward, balance exploration and exploitation, and adapt to dramatic changes in architectures or environments. When networks have a choice between different tasks, they can form preferences that depend on patterns of noise and initialization, and we show that these preferences can be biased by network architectures or by changing learning rates. Our algorithm presents a flexible, biologically plausible way of interacting with environments without requiring an explicit environmental reward function, allowing for behavior that is both highly adaptable and autonomous.

# Contents

|     |  |           |
|-----|--|-----------|
| o   | INTRODUCTION   | <b>1</b>  |
| o.1 | What do I want to understand? . . . . .                                    | 2         |
| o.2 | My initial approach and a rerouting . . . . .                              | 5         |
| o.3 | Summary of thesis . . . . .  | 6         |
| o.4 | A broader note . . . . .   | 8         |
| 1   | ADDING TO A VERY SMALL BRAIN   | <b>9</b>  |
| 1.1 | Introduction . . . . .   | 10        |
| 1.2 | Testing the AI-animal system . . . . .                                     | 21        |
| 1.3 | Methods . . . . .  | 30        |
| 1.4 | Additional notes . . . . .   | 36        |
| 1.5 | Conclusion . . . . .   | 40        |
| 2   | QUESTIONS ANSWERED AND UNANSWERED  | <b>43</b> |
| 2.1 | Mapping neuronal policies . . . . .  | 44        |
| 2.2 | Agents predicted similarities between neural circuits . . . . .            | 47        |
| 2.3 | Cooperative artificial and biological neural networks . . . . .            | 51        |
| 2.4 | RL agents could navigate animals in novel environments . . . . .           | 56        |
| 2.5 | Summary of the RL- <i>C. elegans</i> project . . . . .                     | 57        |
| 2.6 | Questions unanswered . . . . .   | 59        |
| 3   | HOW I LEARNED TO STOP WORRYING<br>AND TRUST THE NETWORKS                   | <b>68</b> |
| 3.1 | Introduction . . . . .   | 69        |
| 3.2 | Task allocation in branches can be predicted by inductive biases . . . . . | 74        |
| 3.3 | Branch specialization can be controlled by curriculum learning . . . . .   | 79        |
| 3.4 | Discussion . . . . .   | 80        |
| 3.5 | Methods . . . . .  | 82        |
| 3.6 | Conclusion and next steps . . . . .  | 84        |
| 4   | BUILDING UP A VERY SMALL BRAIN: OPTIMIZATION                               | <b>88</b> |

|     |   |            |
|-----|---|------------|
| 4.1 | Introduction . . . . .  | 89         |
| 4.2 | Related work . . . . .  | 94         |
| 4.3 | Methods . . . . .   | 96         |
| 4.4 | Results . . . . .   | 98         |
| 4.5 | Explore/exploit modes persist in entropy calculations with different window sizes.  | 103        |
| 4.6 | The effect of noise parameters on behavior. . . . .                                 | 104        |
| 4.7 | Analysis of attractor dynamics . . . . .  | 111        |
| 5   | <b>BUILDING UP A VERY SMALL BRAIN:<br/>BEYOND OPTIMIZATION</b>                      | <b>129</b> |
| 5.1 | PaN adapts to both internal and environmental changes . . . . .                     | 130        |
| 5.2 | PaN autonomously forms task preferences . . . . .                                   | 133        |
| 5.3 | Discussion . . . . .  | 137        |
| 5.4 | Self-supervised behavior: a novel framework for environmental interaction . . . . . | 139        |
| 6   | <b>CONCLUSION</b>   | <b>144</b> |
| 6.1 | Research trajectories, reframed . . . . .   | 145        |
| 6.2 | Technical summary of thesis . . . . .   | 146        |
| 6.3 | Final thoughts; a vignette from the history of ML . . . . .                         | 147        |
|     | <b>APPENDIX A SPECULATION</b>   | <b>152</b> |
| A.1 | What is still missing from PaN? . . . . .   | 153        |
| A.2 | What could the emergent brain account for? . . . . .                                | 157        |
|     | <b>REFERENCES</b>   | <b>190</b> |

# Listing of figures

|     |  |    |
|-----|--|----|
| 1.1 | A system that integrates deep RL with the <i>C. elegans</i> neural network. . . . .  | 12 |
| 1.2 | Ensemble training stabilized performance. . . . .  | 16 |
| 1.3 | L2 regularization did not noticeably improve learning and often made it worse. . .   | 17 |
| 1.4 | Dropout did not noticeably improve performance when applied to actor networks, critic networks, or both. . . . .   | 18 |
| 1.5 | Higher numbers of agents in an ensemble led to more consistent learned action probability matrices. . . . .  | 19 |
| 1.6 | Agents are independently trained and actor policies display variation through randomness in data augmentation, weight initialization, and batch selection. . . . . | 20 |
| 1.7 | The system learned to navigate the <i>C. elegans</i> Line 1 to a target. . . . .   | 22 |
| 1.8 | The system could successfully navigate different optogenetic lines to targets. . . .   | 28 |
| 1.9 | MCB poster thumbnail. . . . .  | 38 |
| 2.1 | Final timesteps for angle variables have larger average magnitudes for agents trained on all lines. . . . .  | 45 |
| 2.2 | The system learned to navigate different optogenetic lines to a target with neuron-specific strategies. . . . .  | 48 |
| 2.3 | Agent policies can predict agent performance on other lines. . . . .   | 50 |
| 2.4 | Animals with agents can correct errors and generalize to novel situations. . . . .   | 54 |
| 3.1 | An excerpt from a poster presented at SVRHM 2021 (Shared Visual Representations in Human and Machine Intelligence), a NeurIPS workshop. . . . .                    | 71 |
| 3.2 | Sample images generated for the Gabor dataset. . . . .   | 74 |
| 3.3 | In networks with differently shaped branches, tasks are allocated based on predicted inductive biases. . . . .   | 75 |
| 3.4 | Even with identical branch architectures, networks tend to specialize. . . . .   | 78 |
| 3.5 | When tasks alternate quickly, network branches tend to specialize. When tasks alternate slower, they are more distributed across branches. . . . .                 | 81 |
| 4.1 | Graphical abstract from Cosyne 2024. . . . .   | 92 |
| 4.2 | Active inference, adapted from Figure 2.3 of Parr and Pezzulo <sup>156</sup> . . . . .   | 95 |

|      |   |     |
|------|---|-----|
| 4.3  | Information flow through the Prediction and Noise network and its update loop. . . . .  | 97  |
| 4.4  | Examining behavior of a two-neuron PaN network. . . . .                                 | 99  |
| 4.5  | Attractors for a two-neuron PaN network. . . . .  | 100 |
| 4.6  | PaN networks learn to maximize sensory signals. . . . .                                 | 102 |
| 4.7  | Closer comparisons of entropy metrics for PaN and $\epsilon$ -greedy agents. . . . .    | 105 |
| 4.8  | PaN behavior in a 3-armed bandit task under different noise settings. . . . .           | 106 |
| 4.9  | Attractors without noise in the two-neuron network. . . . .                             | 121 |
| 4.10 | Motor noise (noise at $x_1$ ) is necessary for reward-seeking behavior. . . . .         | 125 |
| 4.11 | Support for approximations made in Sections 4.7.2 and 4.7.2. . . . .                    | 126 |
| 5.1  | PaN networks adapt to internal and environmental change in bandit environments. . . . . | 131 |
| 5.2  | Open-field search task. . . . .   | 132 |
| 5.3  | Networks exhibit preferences that can be biased. . . . .                                | 134 |
| 5.4  | PaN networks in open-field search tasks can be biased. . . . .                          | 136 |

DEDICATED TO MY PARENTS KUI WEI AND LIYU LI, MY BROTHER DAVID, AND THE AU-  
THORS OF THE BOOKS THAT HAVE TAUGHT ME THAT KNOWING WHAT I WANT TO DO CAN BE  
GOOD—BUT NOT KNOWING CAN BE MUCH BETTER.



# Acknowledgments

LIFE AND SCIENCE SHOULD BE FUN. I've had many wonderful labmates, friends, and colleagues who made both those things exactly that. William Weiter, Jason Chen, Jonah Brenner, Michael Buch, Sammy Hassan, Gino Domel, Jeffrey Aceves, Ben Thorne, Cory McCartan, Anton Graf, Tim Hallacy, Surya Bhupatiraju, Merrick Smela, and Vikram Sundar are only a few of the people that made me enjoy graduate school as much as I did. I also feel incredibly lucky to have stayed in touch with some of my Tri-Cities, Cambridge, and undergrad friends over the years. I've appreciated having them through weddings and hardships and video games and email threads, even though we were usually separated by a few thousand miles: Lucy, Michelle, Sophie, Joanna, Linda, Tiffany, Jared, Amanda, Malavika, Edwin, another Sophie, Peter, Chris, Stephen. And I'm grateful for the friends I've made in this field, from graduate school interviews to the Woods Hole 2021 CBMM summer course to the 2023 Janelia Theoretical Neuroscience workshop.

I'm also very grateful for my English professors Louisa Thomas and James Wood, who gave me refreshing perspectives and new modes of thought. Aside from appreciating the chance to learn from these experts in their fields, I'm absolutely certain that my English classes made me a better scientist.

And finally, I of course want to thank my advisors, Gabriel and Sharad, not just for their valuable scientific guidance, but also for their remarkable patience over the years. Even when they disagreed with me, or when I wasn't making any sense, they gave me their support and let me do whatever I wanted. That freedom was the most important thing I could have asked for as a graduate student, and they let me have it the entire time.

# 0

## Introduction

On Richard Feynman’s blackboard at the time of his death, surrounded by equations and miscellany, were the following words: *What I cannot create, I do not understand*. The spirit of this phrase has always been my approach to understanding the brain. It has also caused me a great deal of frustration.

Progress in artificial neural networks (ANNs) has been almost science fictional. It was 2019 when I started graduate school, and this was the year DeepMind released MuZero<sup>177</sup>—a magnum opus

of a particular type of artificial intelligence called *reinforcement learning* (RL)<sup>190</sup>. MuZero could play video games, chess, Go, and shogi, all learned from scratch and to a level of expertise better than the best people in the world. Right on its heels in 2021 came AlphaFold2<sup>97</sup>, another DeepMind creation, and in terms of contributions to human knowledge, AlphaFold2 was even better than MuZero. AlphaFold2 solved the decades-old problem of protein folding, and members of my lab began to speculate whether it would be the first algorithm to win a Nobel Prize.

Then at the end of 2022, OpenAI released ChatGPT.<sup>185</sup> It was the first time in global dialogue that people were seriously considering the implications of a general artificial intelligence. Since then updates have been rolling out with metronomic regularity, jumping from text to video to audio generation in a matter of months.<sup>29,18</sup> It makes you think that maybe the singularity is on its way. And maybe it is, or at least, some version of one.

But in this thesis, I want to talk about what modern AI *can't* do rather than what it can. Because here is a hole, a huge gaping chasm of a hole that I find very suspicious in the artificial intelligences we are able to make, which is that for all the miracles of machine learning and current ANNs, we still do not have an algorithm that can do what the 302-neuron nematode *Caenorhabditis elegans* does. We're missing too many key features of animal behavior to claim that we're close, and I will describe some of these features below. To me, these missing components imply that we do not yet fully understand one of the smallest, most thoroughly-studied animal nervous systems in existence. In that case, how far along can we say we are for the others?

## 0.1 WHAT DO I WANT TO UNDERSTAND?

Following Feynman, I wanted to have a sense of what I want to understand before I tried to build it. This was more challenging than I anticipated at the beginning. Because there is no accepted concrete definition of natural intelligence, my approach was to draw up a list of computational features

that virtually all animals are capable of implementing.

So what do I mean when I say that we have no algorithm that can do what a worm does? I am *not* looking for an algorithm that can literally crawl or mate or eat. Those specific actions are not what interest me about living nervous systems. What I want instead is an algorithm that can explain and reproduce a number of attributes of living organisms, attributes that, in combination, make natural intelligence unique in the realm of computation.

1. **Complete autonomy in function.** In an autonomous algorithm, one should be able to press “go” and then leave it to its own devices. The agent should be able to take actions, collect its own training data, and exist as its own entity; a self-contained thing.
2. **Reward-seeking.** An animal must be able to represent and chase reward. It has to be able to pursue goals that are relevant to survival and reproduction at minimum. Ideally, it learns to get better at pursuing these goals over time.
3. **Adaptability.** If a *C. elegans* is moved from soil, its natural environment, to a featureless agar plate, it manages to navigate around the new environment despite large changes in sensation and motor response (that is, swimming through jelly is probably very different from climbing pieces of soil). Animals are exceptional at adapting not just to dynamic environments, but also to internal changes: given new sensors and neurons as their bodies grow over time, they learn to incorporate them into their behaviors. In the case of severe injury, such as the loss of a limb or degeneration of the nervous system, animals prove to be incredibly resilient as well.
4. **Flexible task-switching.** Animals do not doggedly pursue one reward for their entire lives like RL agents do when trying to maximize scores in a video game. Instead, animals rarely stay with one task for longer than a few minutes or hours.<sup>95,114,87</sup> This easy flexibility allows

animals to pursue multiple goals over their lifetimes, like food, mates, shelter, etc., which are necessary for survival and reproduction. From a computational standpoint, autonomous and flexible task-switching make natural intelligence able to persist in complex environments while juggling multiple goals at once.

5. **Local update rules and information transfer.** Neurons communicate through local interactions. Locality is not a trivial trait, a box to be ticked in order to comply with biological plausibility. When combined with algorithmic autonomy, it actually changes the entire nature of the computational system. This is because every computation must emerge from the interactions of many components. Goals themselves must be emergent rather than dictated. A great deal of thinking has been done on the implications of a self-organized computational system that execute behaviors based on a principle of emergence, and one can browse the related literature on computational autopoiesis for attempted definitions and discussions. <sup>162,136,94</sup>

All of the attributes above are fairly concrete. For each one, there can be a yes-or-no statement made as to whether an algorithm possesses the feature, or a (possibly invented) quantification of how good an algorithm is at the trait. Some features have been achieved by artificial algorithms, like reward-seeking by RL agents, but there is no existing artificial algorithm that can achieve every feature at once. At the same time, virtually all animals can achieve every listed feature—even the animals with only a few hundred neurons. So it isn't the precise behavior of a nematode that I aim to reproduce, but rather the general, rich, natural intelligence that it and other animals possess, regardless of their size.

## 0.2 MY INITIAL APPROACH AND A REROUTING

This is not to say that people haven't tried to build an algorithm with the capabilities of a worm. OpenWorm<sup>173</sup> was a project at MIT that attempted to simulate the full nervous system of *C. elegans* using existing connectome and physiological data. It went to the level of ion channels in biological neuron models, an approach shared by the Blue Brain Project<sup>135</sup>, but has not managed to progress very far in understanding the *C. elegans* nervous system. Recently other models<sup>12,154</sup> have attempted again to model the worm brain *in silico*, but at best these models can reproduce short-term responses to stimuli instead of the global kinds of attributes listed in Section 0.1.

Then at the end of 2023, a preprint was released by Haspel et al.<sup>77</sup> on why the simulation of an entire nervous system was a worthwhile project for neuroscience, and how one might go about accomplishing it. Haspel et al.'s claim was that the field is still severely data-limited, which could be true, but made it hard for me to know where to begin as one graduate student at the start of my degree. I wanted to build an entire simulated nervous system and, like Haspel et al.,<sup>77</sup> thought that *C. elegans* was the best candidate to use as a starting point. But there appeared to be an unending amount of possible data to collect, even on such a small animal, and so I wanted to find more general principles by which to simulate the worm. I didn't know what these general principles might be, but I did find myself intrigued by the success of deep reinforcement learning agents around the time I started graduate school. Of the three classes of machine learning—supervised, unsupervised, and reinforcement learning—reinforcement learning is the only one that deals with learning from actions taken in an environment. Since any model of an animal would be incomplete without accounting for environmental interaction, I thought that RL was a good starting point for a possible model.

However, I didn't know where to begin with RL: what would make an RL agent more like a living *C. elegans*, as opposed to the machine-based agents that were out there already? It would have

been difficult to build an RL agent and then try to convince myself and others that it modelled a *C. elegans* accurately. I thought (very naïvely) that a more meaningful angle could be to *start* with an existing *C. elegans* and try to add onto it with a computational network. I thought that if I could understand how a living nervous system might grow from  $n$  to  $n + 1$  neurons, then I could infer how it could start from 0 and grow to  $n$ . That was where I began: connecting existing artificial algorithms (reinforcement learning ones) to a living brain.

I spent three years carrying out the idea. I combined artificial and biological neural networks in the *C. elegans* brain in a way such that the artificial networks improved the animal’s ability to find food. It was a biologically relevant task—perhaps *the* relevant task for *C. elegans*, which are mostly hermaphroditic and can clone themselves by laying eggs without mating. But after getting the system to work, I thought that it really didn’t tell me much about how to get from 0 to  $n$  neurons after all, and wasn’t even a good mimic of growing the nervous system from  $n$  to  $n + 1$ . I elaborate on how the RL agent differed from a living nervous system in Section 2.6.

My dissatisfaction with the RL-*C. elegans* hybrid sent me down a much less certain path of trying to understand what it really meant when I said I wanted to build a natural intelligence. It led to the list of features in Section 0.1 and a new appreciation of what it seemed like a nervous system must be: a network of many individual components, where behaviors, goals, and computations *emerge* from the individual neurons; much like how a large flock of birds will stretch, compress, peel apart and come together again. I discuss this further in Chapters 4 and 5, and possible implications of a truly emergent nervous system in A.2.

### 0.3 SUMMARY OF THESIS

This thesis can be split into two main parts, connected by an interlude.

PART 1 Chapters 1 and 2 are about adding to the *C. elegans* nervous system using artificial neural networks, specifically, a deep reinforcement learning agent that could learn to use different sets of neurons in the animal to achieve some task. I describe the system and show that it works in Chapter 1. In Chapter 2, I present assorted applications of the hybrid system, and conclude with an explanation of why I decided to stop working on the project.

INTERLUDE Chapter 3 is the interlude. It consists of a shorter project that gave me ideas about the work in the last two chapters of the thesis. In Chapter 3, I show that in supervised learning tasks using artificial neural networks with branched architectures, the networks can allocate different tasks to branches depending on the shape of the branches. The branch shapes impose inductive biases on the separated branches, lending differently shaped branches to learn tasks that were best suited to their shapes.

Other publications had shown related phenomena,<sup>199</sup> but working on this project made me think that perhaps I didn't have to rely so much on *controlling* what artificial neural networks learned and how they learned it. The work in this chapter was done in collaboration with Arturo Deza, then a postdoctoral researcher in Tomaso Poggio's group, and was based on previous work of his.<sup>46</sup>

PART 2 The last part of the thesis describes my most recent approach to building a small nervous system, this time using emergent principles and the neurophysiologically plausible mechanisms of prediction and noise. The algorithm, named Prediction and Noise (PaN), satisfies the list in Section 0.1 and serves as a new model of how neurons take input and produce behaviors. Chapter 4 describes PaN and shows how it can seek out reward while autonomously balancing exploration and exploitation, which hadn't been accomplished in a purely local prediction-based algorithm before. PaN is also consistent with the "singular success of attractor neural network models" in describing



the brain,<sup>105</sup> and at the end of the chapter I show work done with an outstanding undergraduate student, Jonah Brenner, that explains how PaN implements behaviors through attractor dynamics. Another excellent undergraduate student, Adam Boesky, contributed to the work in this chapter through a class project.

Chapter 5 then goes beyond optimization and shows how the same algorithm can form task preferences and change those preferences over time. I show that these preferences can be biased in biologically relevant ways, like connectivity, past experience, or neuromodulation-like signals. Chapter 5 also includes a section that defines and motivates an accompanying framework for PaN called *self-supervised behavior*, which I present as an alternative to reinforcement learning, as a framework for artificial algorithms that learn actions to take in an interactive environment.

I have reserved some more speculative discussion regarding the work in Chapters 4 and 5 for Appendix A. Section A.1 discusses future directions I see for the model, and Section A.2 is a more philosophical treatment of PaN and how it might relate to certain topics in neuroscience and psychology.

#### 0.4 A BROADER NOTE

It has not been easy for me to support my claims of what a field lacks instead of what it has. It has been even harder to say why those pieces are missing. But I do think it has been important to take stock every once in a while of where I've been going, where the field seems to be going, and try to figure out if the general directions still make sense rather than always forging onward.

*Natural selection almost always builds on what went before, so that a basically simple process becomes encumbered with many subsidiary gadgets. As François Jacob has so aptly put it, “Evolution is a tinkerer.” It is the resulting complexity that makes biological organisms so hard to unscramble.*

Francis Crick, *What Mad Pursuit*

# 1

## Adding to a very small brain

CAN YOU GROW A LIVING BRAIN WITH ARTIFICIAL NEURONS? In this chapter, I discuss my attempt to do exactly that. I didn't succeed (see Chapter 2), but I did make a cool reinforcement learning-based neural interface. This chapter discusses the interface itself and what it can do.

## 1.1 INTRODUCTION

Guiding or improving animal behavior directly through the nervous system is a common goal for neuroscientists and robotics researchers alike<sup>25,164,192</sup>. Previous work has attempted to use direct interventions to affect behavior on a variety of tasks, relying on manual specification for stimulation frequencies, locations, dynamics, and patterns<sup>3,23,51,83,86,93,102,117,133,158,171,172,174,175,191,207,155,209</sup>. But it is impractical to rely on manual tuning for everything—it relies too much on preexisting knowledge of the neural circuits or mechanisms involved. This preexisting knowledge is not always available. We often don't know the right neural activation patterns for given tasks or sets of neurons, and there can be a combinatorial explosion of stimulation parameters to test.

Getting the right patterns for direct neural stimulation, especially when trying to elicit some specific behavior, can be difficult. The patterns depend on the particular neurons that are targeted, and in nervous systems larger than that of a *C. elegans*, it can be impossible to guarantee that the targeted neurons play the same parts in behaviors, animal to animal<sup>17,145</sup>. So even though technologies for precise neuronal modulation exist<sup>90,147</sup>, it is difficult to design algorithms to control them. And if you were to design such an algorithm, it would ideally be able to systematically and automatically learn strategies to activate the set of neurons it connects to in order to improve the specific behavior you had in mind<sup>110,121,201,81,50</sup>.

Here we addressed this challenge using deep reinforcement learning (RL), assessing whether RL can autonomously integrate with an animal's nervous system to improve behavior. In an RL setting, an agent collects rewards through interactions with its environment. By leveraging deep neural networks, RL algorithms have successfully discovered complex sequences of actions to solve a wide set of tasks<sup>179,180,177,144,198,153,206,44,91,78,73</sup>. These past successes relied on reward signals to train algorithms, a framework that can be adapted to biologically relevant goals, such as finding food or mates. While other studies have incorporated machine learning into designing cyborg or bio-

hybrid organisms<sup>209,9,211,212</sup>, they have largely focused on optimizing only one means of interfacing with an animal, which could be difficult to scale up in neural interfaces especially given the highly variable nature of living nervous systems. By using deep RL, we present instead a flexible framework that can, given only a reward signal, observations, and a set of relevant actions, learn different ways of achieving a goal behavior that adapt to the chosen interface.

We tested our ideas on the nematode *C. elegans*, interfacing an RL agent with its nervous system using optogenetic tools<sup>90,121</sup>. This animal has a small and accessible nervous system while still possessing a rich behavioral repertoire<sup>8</sup>, making it a suitable candidate to test deep RL integration. In a natural setting, *C. elegans* must navigate variable environments to avoid danger, or find targets like food. Therefore, we aimed to build an RL agent that could learn how to interface with neurons to assist *C. elegans* in target-finding and food search. We tested the agent by connecting it to different sets of neurons with distinct roles in behavior, where some of these neuronal sets did not have fully understood roles in directed movement. Agents could not only couple with different sets of neurons to perform a target-finding task, but could also generalize to improve food search across novel environments in a zero-shot fashion, that is, without any prior training. We show that our neural-RL interface can be used to investigate the function of neural circuits in task performance, including with sets of neurons whose links to behaviors have not been previously established.

Figure 1.1: **A system that integrates deep RL with the *C. elegans* neural network.**

(A) Concept for combining artificial and biological neural networks for a shared task.

(B) Closed-loop setup using optogenetics. A single nematode was placed in a 4 cm-diameter field and illuminated by a red ring light for imaging. A camera and a high-powered LED (blue or green) were connected to a computer to form a closed-loop system. The LED modulated neurons carrying optogenetic constructs (see main text).

(C) Reward at time  $t$ ,  $r_t^{(15)}$  was defined as the change in distance to target between times  $t$  and  $t + 15$ .

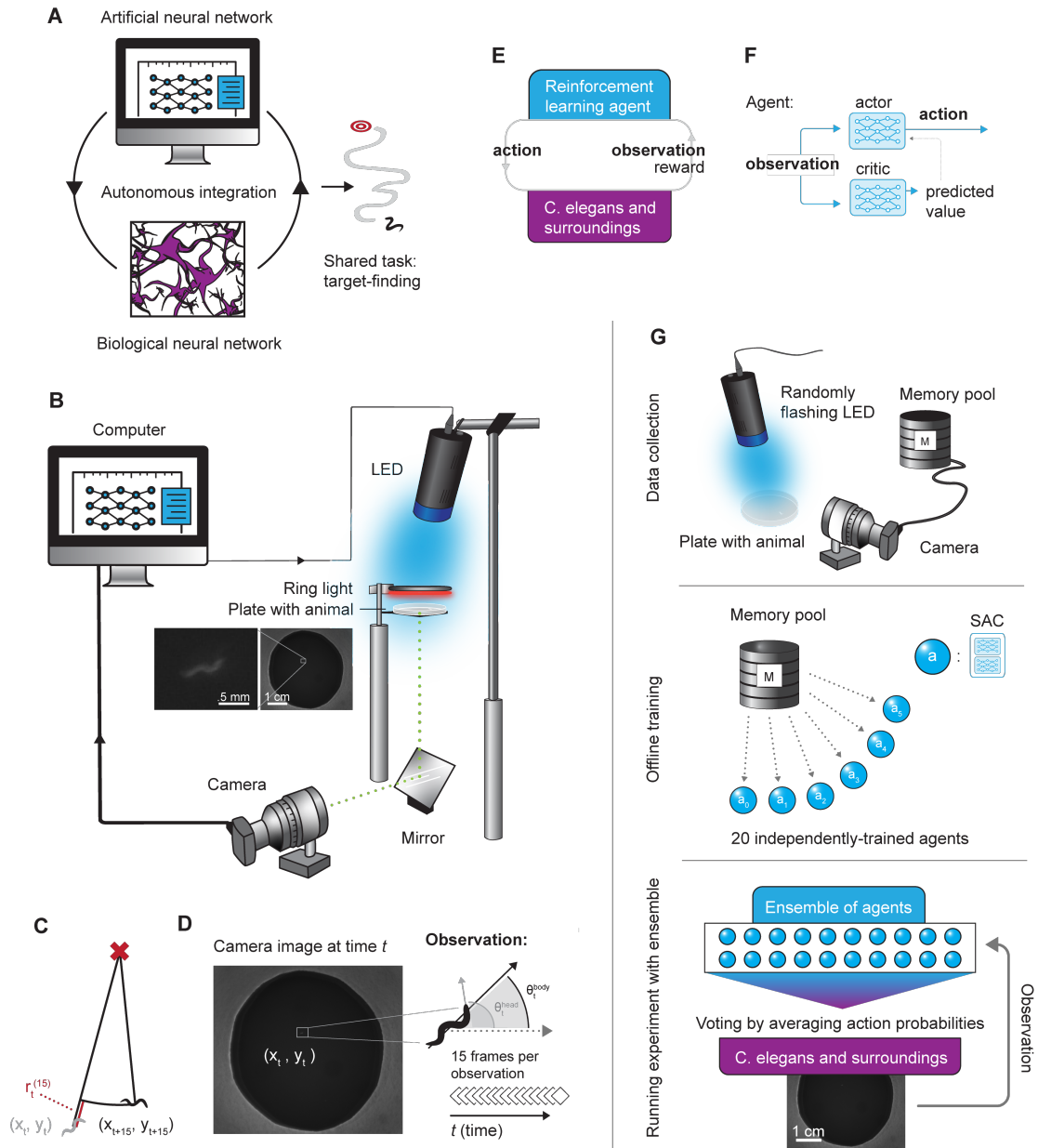
(D) Sample camera image at time  $t$ . An observation was a stack of 6 measurements from 15 frames (5 s at 3 fps) for a total of 90 variables per observation received by the agent at each timestep. Measurements were coordinates of the animal's center of mass at time  $t$   $(x_t, y_t)$ , and the sines and cosines of the head and body angles,  $(\theta_t^{body}, \theta_t^{head})$  of the animal relative to the positive x-axis.

(E) RL loop diagram of the combined system.

(F) Actor-critic architecture used as a deep RL agent.

(G) Pipeline for training and evaluating the RL-animal system. A total of 5 h of data were collected where a light is flashed randomly on an animal, stored in a memory pool. Animals were switched out approximately every 20 minutes. Multiple soft actor-critic agents were independently trained on the memory pool. During evaluation, the agents were put into an ensemble that voted in real time on actions. Each individual agent's decision was based on the observation received from the camera.

Figure 1.1: (continued)



We formulated target-finding as an RL problem by defining a reward value as the negative distance of the animal’s coordinates to a user-specified target (Figure 1.1C; see Section 1.3). The RL agent’s environment consisted of a 1 mm adult animal and a 4 cm-diameter arena on an agar plate. Observations of the environment were given to the agent through a camera at 3 Hz and features were automatically extracted from each camera frame to track the animal’s center of mass. During evaluation, target coordinates were subtracted from the animal’s coordinates before being sent as part of the input to agents,  $(x_t, y_t)$ . Head and body angles  $(\theta_t^{body}, \theta_t^{head})$  were extracted from each frame relative to the  $+x$ -axis, and head angles were measured relative to body angles. We took polar coordinates of the angle measurements so that an observation was defined for every frame  $t$ ,  $(\sin(\theta_t^{body}), \cos(\theta_t^{body}), \sin(\theta_t^{head\ rel.}), \cos(\theta_t^{head\ rel.}), x_t, y_t)$  (Figure 1.1D). Each observation the agent received included these six variables from frames over the past five seconds, making agent inputs 90-dimensional at each timestep (6 variables  $\times$  3 frames per second  $\times$  5 seconds). These variables are relevant for the navigation task, although we note that other tasks may benefit from different sets of task-specific variables.

Given an observation at time  $t$ , the RL agent was trained to learn what action  $a_t$  to take at that time to maximize return, defined as a sum of rewards discounted over time (Figure 1.1E, Section 1.3). To take an action, the agent could decide whether to turn an LED on or off at each timestep. Using optogenetics<sup>90</sup>, the agent could modulate selected neurons that expressed either channelrhodopsin, a light-gated ion channel that can be stimulated by blue light (480 nm) to activate neurons<sup>147</sup>, or archaerhodopsin, a light-sensitive proton pump that can be stimulated with green light (540 nm) to inhibit neurons.

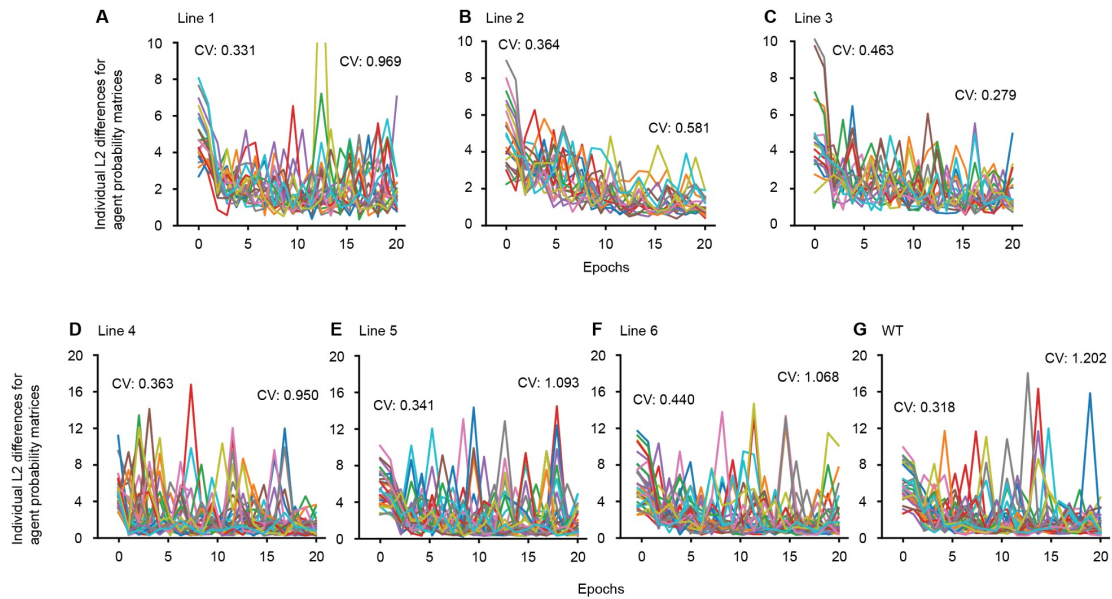
We chose the soft actor-critic (SAC) algorithm for the RL agent because of its successes in simulated and real-world RL environments<sup>206,73,35,204</sup>. SAC has separate neural networks for a critic that learns to evaluate observations and an actor that learns to optimize actions based on critic evalu-

ations for return maximization (Figure 1.1F). Both networks take observations as input and consist of two layers with 64 units per layer. The actor network outputs probabilities of turning the light on at time  $t$ ,  $P(a_t = 1)$ . We assigned the agent’s action for that observation as “light on” if the actor’s output  $P(a_t = 1) \geq 0.5$ .

Deep RL tends to require large amounts of data. For instance, agents learning to play Atari can require thousands of hours of gameplay to achieve good performance<sup>177,144</sup>. It was infeasible to collect thousands of hours of recordings in our environment, and unlike videogames or physical systems with reliable dynamics, adequate computer simulations of the *C. elegans* nervous system and its behaviors are not available to generate training data<sup>173</sup>. Therefore, agents were trained offline on pre-recorded data, collected for 20 min per animal for a total of 5 h. During training data collection, the light was turned on randomly with a probability of 0.1 every second (Figure 1.1G, top). Following approaches in supervised learning<sup>178</sup>, the data were then augmented during training by randomly translating and rotating the animal in a virtual arena approximately the size of the 4 cm-diameter evaluation arena (Section 1.3).

During training, deep RL agents were unstable and prone to sudden performance drops (Figure 1.2), similar to previous work<sup>149, noa</sup>. In simulated environments, performance crashes can be monitored with evaluation episodes in the exact environment used for testing. In our environment, evaluation episodes were impractical because they would have required many more times the amount of data than were used to train agents. So we tested several regularization methods to help with stability, and found that ensembles of agents were effective for our environment (Figure 1.3-1.5). The final deep RL agents were ensembles of SAC agents, with the collection, training, and evaluation pipeline shown in Figure 1.2G. For Lines 1-3 described in Table 1.1, ensembles consisted of 20 agents. For Lines 4-6, which exhibited less stable training dynamics, ensembles consisted of 30 agents (see Section 1.3 for training protocol). Figures 1.5-1.6 show examples of variation between independently trained agents, and how ensembles stabilized agent policies.





**Figure 1.2: Ensemble training stabilized performance.** Individual agents during training were compared to ensembles after 20 epochs of training using a makeshift scoring metric. The metric was defined as the L2 difference between action probability matrices of the tested agent and the fully trained ensemble for each genetic line (as in Figure 1.7L-M).

**(A-F)** The individual L2 differences for each agent in the ensembles for each transgenic line and **(G)** wild type animals. Each color denotes a different agent. The variability of agents is high and tends to increase from their initializations, reflected by coefficients of variation (CVs) printed for first and last epochs of each plot. Note the instabilities, reflected by occasional sharp increases in error.

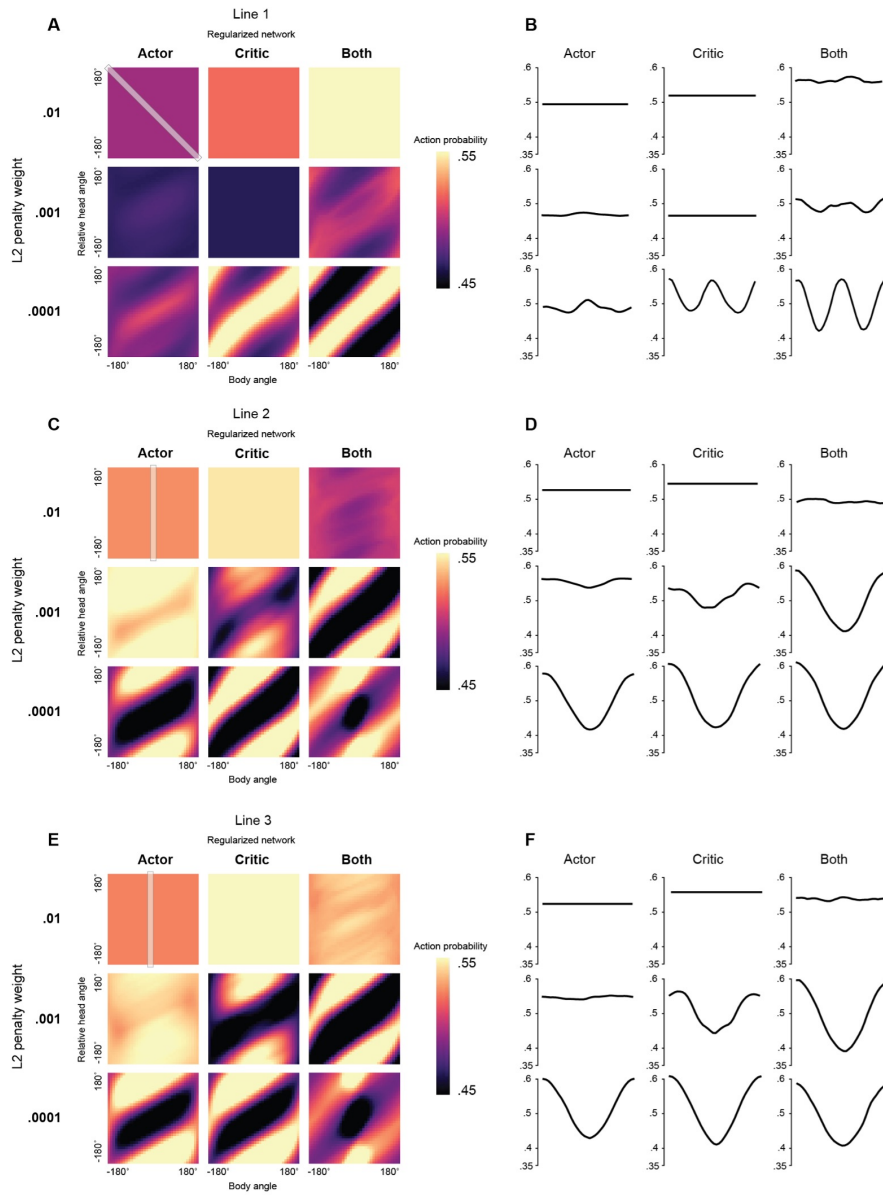
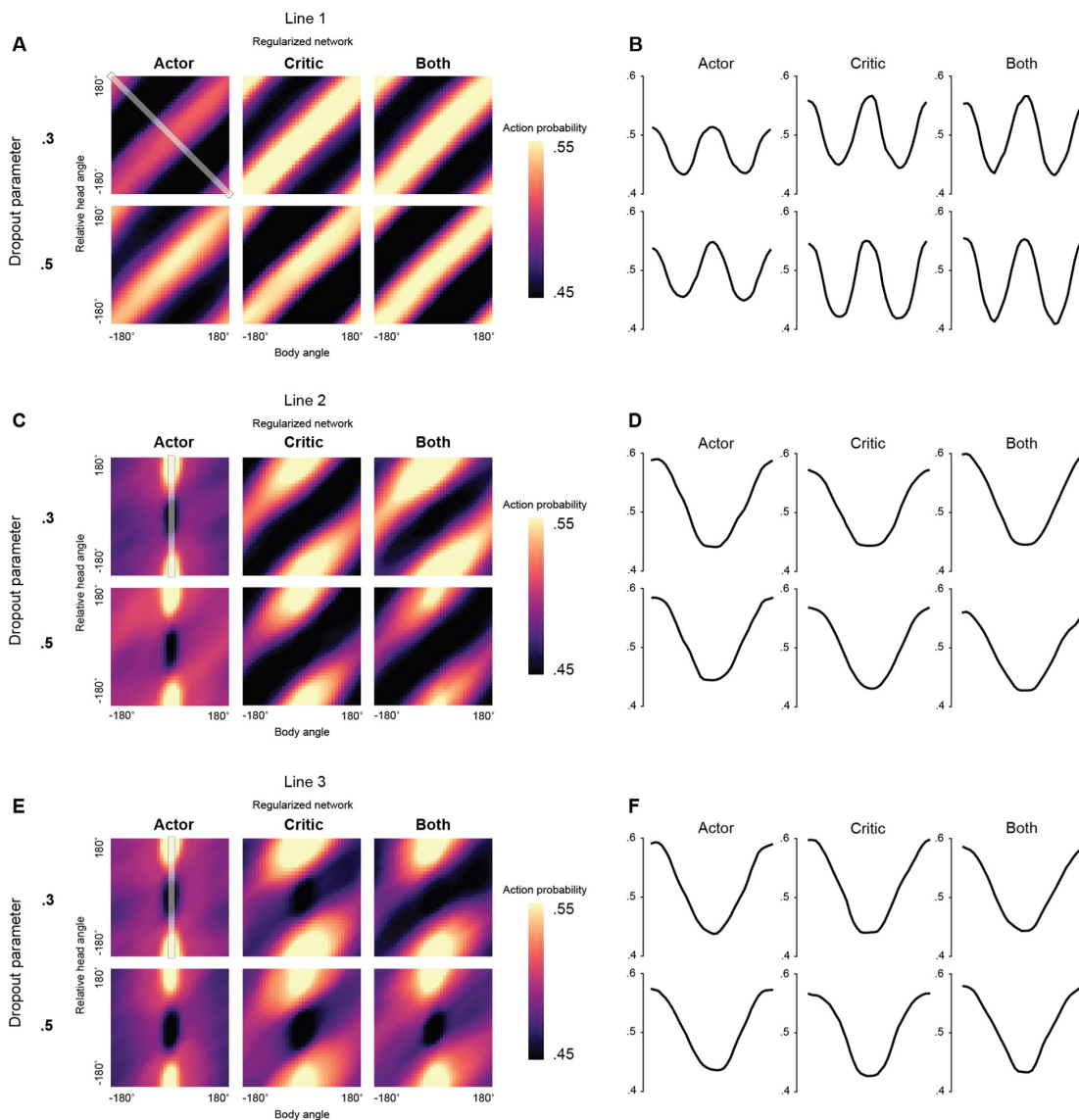


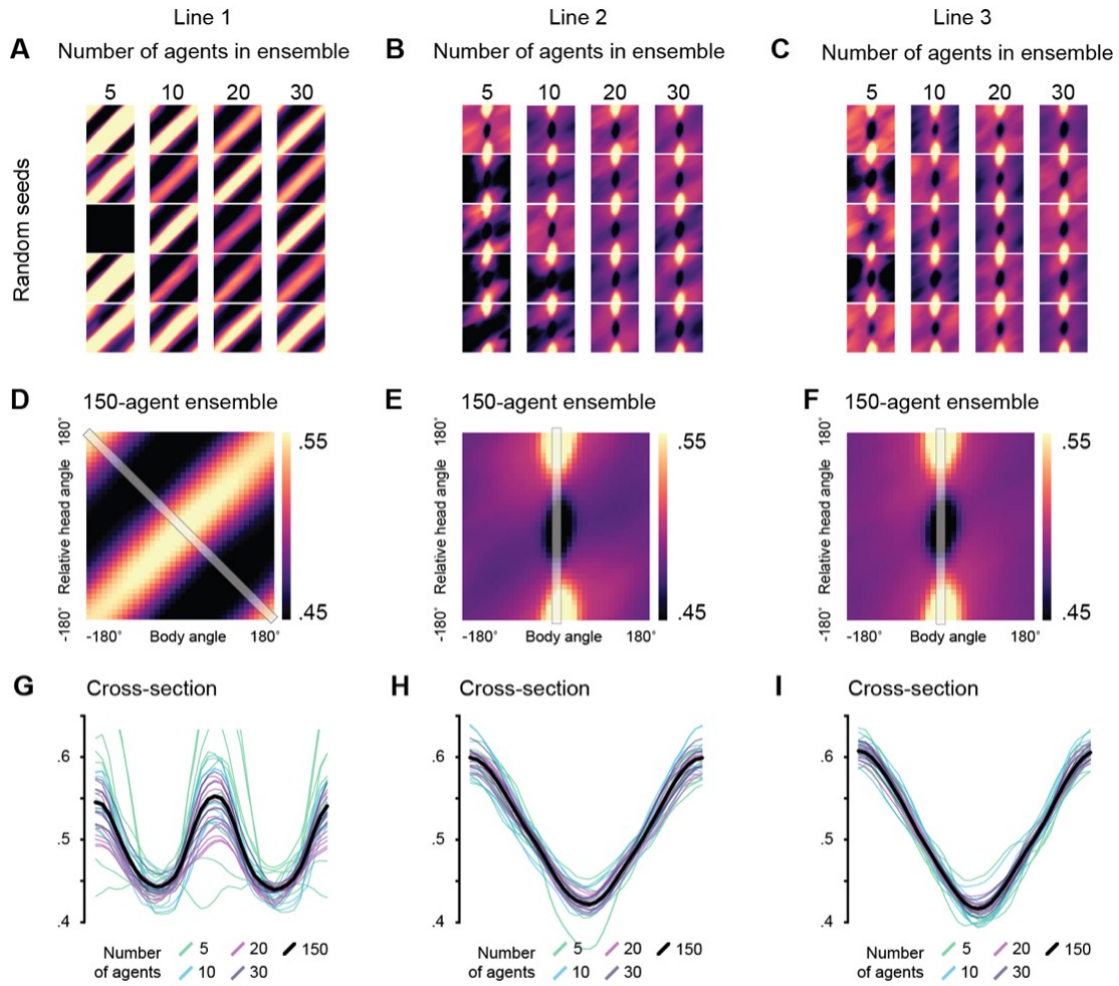
Figure 1.3: L2 regularization did not noticeably improve learning and often made it worse. (A, C, E) Ensemble action probability matrices (Figure 1.7, 4D) using the same training methods as in agents used for evaluation (Section 1.3), but for L2 weight penalties of 0.01, 0.001, 0.0001 over the whole network that the regularizer was applied to. These penalties are proportional to squared weight magnitudes. (B, D, F) Traces of cross-sections highlighted in the top left matrix of (A, C, E). Traces follow cross-sections top-to-bottom along the x-axis.



**Figure 1.4: Dropout did not noticeably improve performance when applied to actor networks, critic networks, or both.**

(A, C, E) Ensemble action probability matrices (following Figures 1.7) using the same training methods as in agents used for evaluation (Section 1.3), but for dropout rates of 0.3 or 0.5 over the whole network that dropout was applied to. These are rates at which neurons are randomly selected to not be part of the feedforward pass.

(B, D, F) show cross-sections highlighted in the top left matrix of (A, C, E). The x-axis for all plots follows the cross-sections top-to-bottom.



**Figure 1.5: Higher numbers of agents in an ensemble led to more consistent learned action probability matrices.** 150 agents were trained on the dataset for each genetic line. (A-C) 5, 10, and 20 agents were drawn randomly without replacement from the 150-agent pool and their action probabilities averaged to form an ensemble. Data are for Lines 1-3. For the 30-agent column, the 150-agent pool was split into 5 equal parts to form ensembles. (D-F) The entire pool of 150 agents was averaged to form one ensemble. (G-I) Cross-sections of ensembles to show variability, taken from the highlighted strip in (D-F).

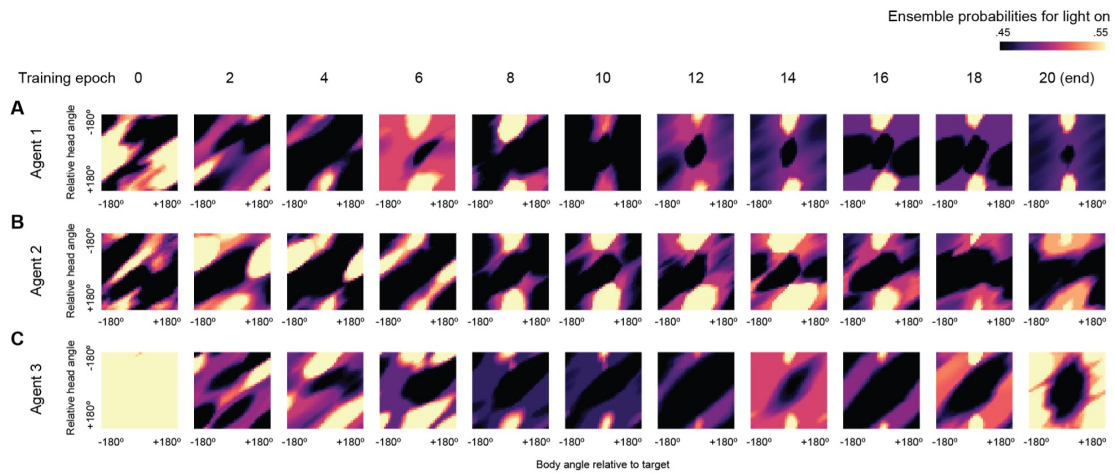


Figure 1.6: Agents are independently trained and actor policies display variation through randomness in data augmentation, weight initialization, and batch selection.

(A-C) Three independently trained agents on animal Line 2, as in the tutorial in the online repository (<https://github.com/ccli3896/RLWorms.git>). Policies are plotted for a location 1 cm to the left of the target. Agents vary substantially in their policies, plotted in A-C here over the course of 20 epochs of training. See methods for additional training details. Averaging over agents stabilized policies, as in Figure 1.5.

## 1.2 TESTING THE AI-ANIMAL SYSTEM

### 1.2.1 AGENTS COULD NAVIGATE ANIMALS TO TARGETS

We first tested our system on the transgenic line *Pttx-3::ChR2*, referred to as Line 1 in the text (Figure 1.7A, Table 1.1). In Line 1, the *ttx-3* promoter drives expression of channelrhodopsin in AIY interneurons, which are known to be involved in chemotaxis. Prior work has established a deterministic strategy for navigating animals using optogenetic activation of AIY<sup>110</sup>, used here as a “human expert” standard to see whether our agent could achieve similar performance.

**Figure 1.7: The system learned to navigate the *C. elegans* Line 1 to a target.**

**(A)** Optogenetically modified neurons AIY (black arrow) in Line 1.

**(B)** Evaluation setup. The animal was placed in the center (purple circle) of a filter paper circle with diameter 4 cm. In each 10 min episode, agents were tested on their ability to navigate the animal to one of the four target locations shown (red).

**(C-F)** Sample tracks with agent, without agent, with random light, and with a “human expert” policy from literature 26, respectively.

**(G)** Closest distance to target achieved by animals for trials with and without an agent as well as with random light stimulations (n=10 for each condition). Animals with agents moved significantly closer to targets than animals without agents. Plots show means +/- SEM. One-sided Mann-Whitney U Test, with agent vs. with control conditions indicated by asterisks, \*\*P<.01, \*\*\*P<.001. (Learned policy: p=.00054, no agent; p=.00019, random light. Known policy: p=.0011, no agent; p=.00017, random light.) Times to reach within 0.5 cm of target for animals with learned and known policies were comparable, shown in inset (n.s., p=.36, one-sided Mann-Whitney U test).

**(H)** Weights of the first 64-neuron layer in all actor (top) and critic (bottom) networks in the agent ensemble. For angle-related variables, the most recent frames (black arrows) had the largest weights.

**(I)** Reference for the policy plots in J-K, showing example animal conformations.

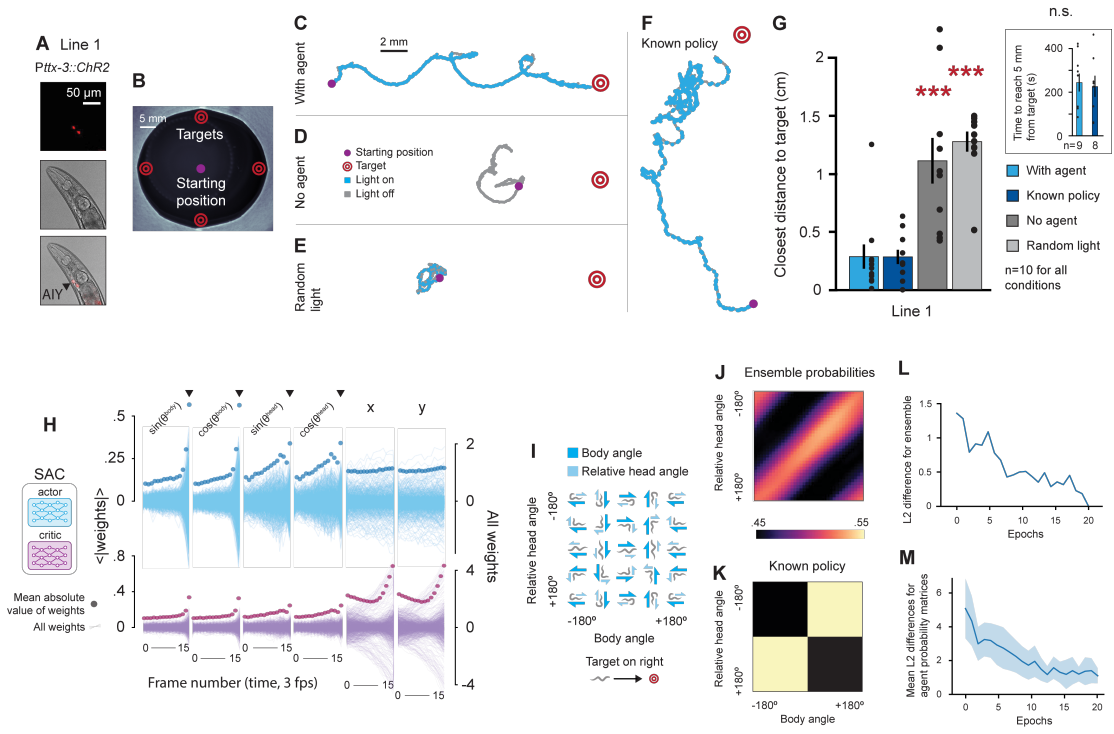
**(J)** Trained agent probabilities for simulated inputs.

**(K)** The human expert policy plotted for comparison. It is similar to the learned agent policy, but not identical.

**(L)** The L2 difference in the policy matrix between the final ensemble and ensembles at each epoch during training. By definition, the difference is 0 at epoch 20.

**(M)** Mean L2 differences between individual agents and the final ensemble, with standard deviation shaded in blue.

Figure 1.7: (continued)







| Name   | Promoter            | Expression  | Genotype  |
|--------|---------------------|---|---|
| Line 1 | <i>Pttx-3::ChR2</i> | AIY   | sraEx281[Pttx-3::chop-2(H134R)::TagRFP; pBX]; pha-1(e2123)III; lite-1(ce314)x                 |
| Line 2 | <i>Pstr-2::ChR2</i> | AWC(ON), [ASI] <sup>*194</sup>                        | sraEx301[Pstr-2::chop-2(H134R)::TagRFP; Pstr-2::TagRFP; pBX]; pha-1(e2123)III; lite-1(ce314)x |
| Line 3 | <i>Pnpr-4::Arch</i> | SIA; SIB; RIC; AVA; RMD; AIY; AVK; BAG <sup>120</sup> | sraEx352[Pnpr-4::Arch-GFP; Pnpr-4::mKO; pBX]; pha-1(e2123)III; lite-1(ce314)x                 |
| Line 4 | <i>PF25B3.3</i>     | All neurons   | sraEx446[pF25B3.3::Arch-tagRFP]; pha-1(e2123)III; lite-1(ce314)x                              |
| Line 5 | <i>Pflp-3::ChR2</i> | IL1; PQR <sup>106</sup>                               | sraEx336[Pflp-3::ChR2-EYFP; Pflp-3::mKO; pBX]; pha-1(e2123)III; lite-1(ce314)x                |
| Line 6 | <i>Pacr-2::ChR2</i> | Cholinergic ventral cord motor neurons <sup>96</sup>  | sraEx437[Pacr-2::ChR2-EYFP; Pacr-2::Arch(D95N)-mKO]; pha-1(e2123)III; lite-1(ce314)x          |

Table 1.1: **Transgenic line names in text with their genotypes and expression.** Neurons in brackets indicate weak or unstable expression in both the reporter lines in literature and the transgenic lines that we generated. Promoters in italics.

After training an RL agent on Line 1, the agent was evaluated by placing an animal in the center of a 4 cm-diameter arena and entering target coordinates as input to the agent (Figure 1.7B). The agent was set to navigate the animal over a 10 min episode to a target placed in one of four possible locations. The agent learned a pattern of light activation (blue points) to maneuver the animal toward the target. A sample track of an animal driven by the agent to a target is in Figure 1.7C (see also Video S1). In contrast, when the light was off all the time (Figure 1.7D), or turned on randomly (Figure 1.7E, Video S2), the animal fails to reach the target. For comparison, we considered the case where the light was turned on according to the known “human expert” policy, which was also successful in driving the animal to the target (Figure 1.7F). Figure 1.7G shows statistics for each condition: the closer the distance to the target, the better the performance. The agent’s learned policy performed as well as the known policy, and both of those performed significantly better than controls (learned policy:  $p=0.00054$ , no agent;  $p=0.00019$ , random light. Known policy:  $p=0.0011$ , no agent;  $p=0.00017$ , random light). There was no significant difference in the time taken to reach within 0.5 cm of the target between the learned and known policies; Figure 1.7G inset ( $p=0.36$ , one-sided Mann-Whitney U test).

To understand what the agent trained on Line 1 had learned<sup>110</sup>, we sought a representative subspace of the 90-dimensional observation space in which to plot agent decisions. For every SAC agent in the ensemble, we plotted weights of the first layer of the actor network as a function of frame number to assess which input variables were associated with large weights (Figure 1.7H). Head and body angles corresponding to the most recent frame in an observation (black arrows in Figure 1.7H) had larger weight magnitudes than in earlier frames. Therefore, to visualize agent strategies, we fixed the 30 coordinate variables  $((x_{t'}, y_{t'}); t - 5s < t' < t)$  in each observation to a position left of the target (Figure 1.7I, Section 1.3) and plotted the probability that the ensemble turned the light on as a function of body and head angles at the latest time in the observation  $(\theta_t^{body}, \theta_t^{head})$  (Figure 1.7J).

The human expert policy is plotted in Figure 1.7K using the same projection.

To interpret the policies, it is useful to compare Figures 1.7I and J. The high-probability diagonal band in Figure 1.7J corresponds to the same diagonal in Figure 1.7I where the animal's head points toward the target. Interestingly, the agent's learned policy was conceptually similar but quantitatively different from the known expert policy in Figure 1.7K, which placed greater emphasis on turning animals in the correct direction. Nonetheless, both policies were effective in the targeted navigation task.

The projection in Figure 1.7J provided a way to plot agent training progress, with Figure 1.7L-M showing the change in agent policies over 20 epochs of training. Figure 1.7L is the difference between the policy of full ensembles during and after training, while Figure 1.7M takes differences between individual agent policies and compares them to the trained ensemble, plotting average differences with standard deviations. We saw that individual agents, even after training, could be quite far from the final policy, which highlighted the importance of ensembling.

### 1.2.2 AGENTS WORKED WITH A VARIETY OF NEURONAL SETS

We aimed to build a robust and flexible algorithm that could adapt to its connected neurons. We next tested whether the RL agent could learn appropriate rules for a variety of neural connections without explicit prior knowledge about them. New agents were trained on five additional transgenic lines that were functionally distinct from Line 1 and did not have associated human expert policies (Figure 1.8). These lines are ordered in the text by agent performance compared to no light and random matched-frequency light controls. See Table 1.1 and Figure 1.8A for line genotypes and neuron expression.

Figure 1.8: **The system could successfully navigate different optogenetic lines to targets.**

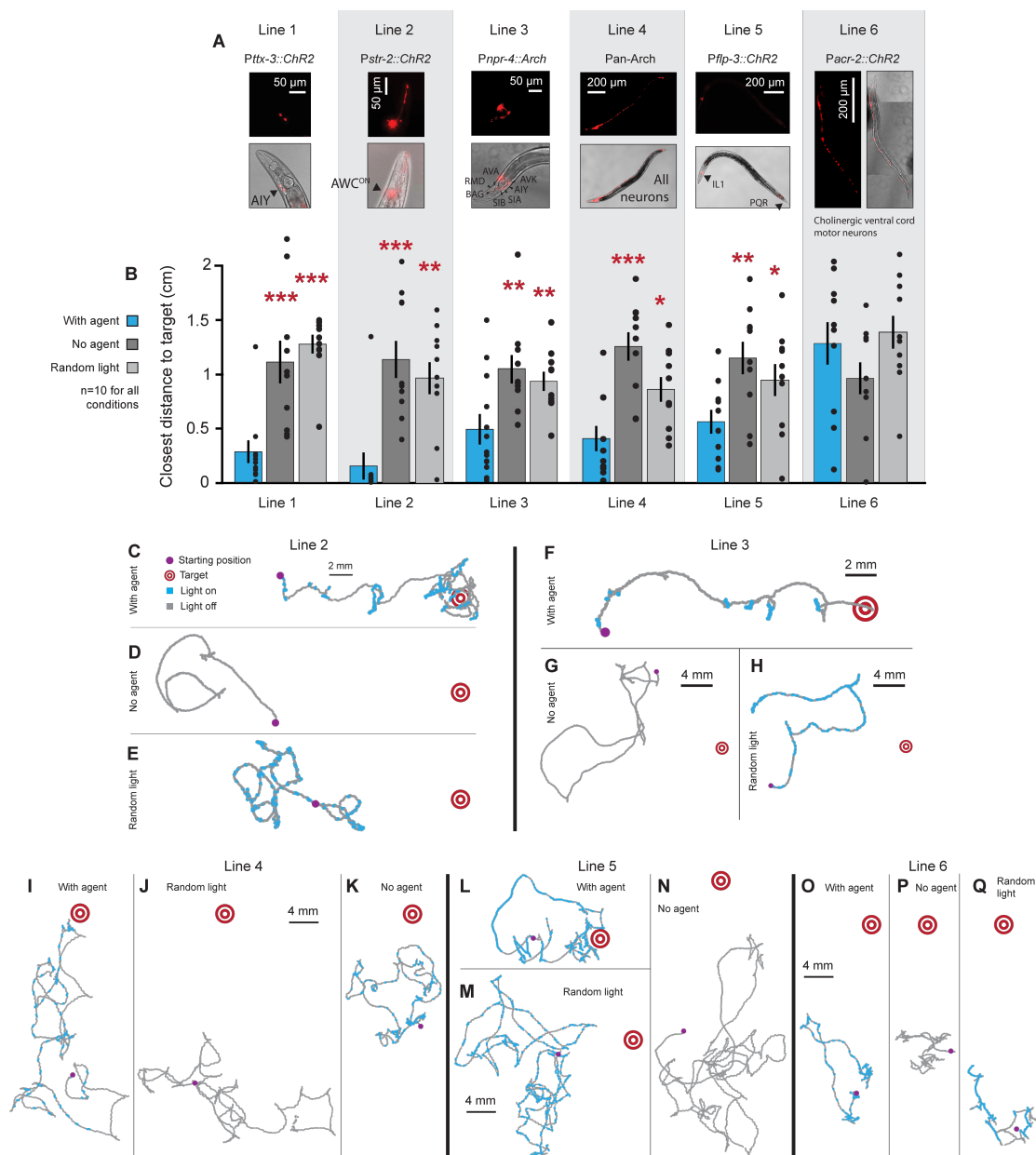
**(A)** Images of optogenetic lines with promoters and modified neurons.

**(B)** Statistics for each line (n=10) comparing performance with agents, without agents, and with frequency-matched random light controls, plotted as means +/- SEM. One-sided Mann-Whitney U Test, with agent vs. with control conditions indicated by asterisks, \*P<.05, \*\*P<.01, \*\*\*P<.001. Lines 1-5 were successful. Line 1: p=.00054, no agent; p=.00019, random light. Line 2: p=.0005, no agent; p=.0029, random light. Line 3: p=.0060, no agent; p=.0071, random light. Line 4: p=.0008, no agent; p=.0104, random light. Line 5: p=.0057, no agent; p=.03216, random light. Line 6: p=.9192, no agent; p=.4841, random light.

**(C-E)** Following the format in Figure 1.7, example tracks for Line 2 with positions of light activation along the trajectory highlighted in blue for animals C, with the agent, D, without any optogenetic activation, and E, with randomly flashing light.

**(F-Q)** Example tracks for Lines 3-6 for each experimental condition in C. Variability in starting positions for controls can be explained by free movement in the time between placing animals on the plate and starting the experiment, approximately 1 min.

Figure 1.8: (continued)



Lines 3-6 expressed light-sensitive channels in multiple neuron types. Line 3 and 4 animals expressed archaerhodopsin, which inhibits neurons upon stimulation with green light (540 nm). Due to concerns about phototoxicity, agents for Line 4 were restricted to short pulses during evaluation (Section 1.3). These lines tested the abilities of the RL agent with different sets of neuronal connections and different means of modulation.

In Lines 1-5, animals with trained agents moved closer to targets than control animals did (Figure 1.8B). Example tracks showing agent activity during evaluation and controls are shown in Figure 1.8C-Q. Videos are available at <https://www.biorxiv.org/content/10.1101/2022.09.19.508590v2> in Supplementary Material; Videos 1-6 show agent performance and controls for Lines 1-3, which performed best. Given that policies for goal-directed movement using optogenetic modulation of these lines were previously unknown, it was remarkable that agents still learned to direct these animals towards a target (for Line 3, see Bhardwaj et al., 2018<sup>19</sup> for *npr-4* mutant behavior and for Line 5, see<sup>163</sup> for IL1 involvement in head withdrawal).

## 1.3 METHODS

### 1.3.1 REINFORCEMENT LEARNING DETAILS

RL is a framework in which an agent interacts with an environment and attempts to maximize a reward signal. The agent receives observations from the environment, giving it an idea of the environment's current state, and learns what actions to take that will be most likely to maximize the reward signal received from the environment. The RL agent learns through experience an action probability distribution,  $\pi(a_t|s_t)$ , where  $a_t$  is the action taken at time  $t$ ,  $s_t$  is the state received from the environment corresponding to time  $t$ , and the maximized reward  $r_t$  is received at time  $t$ . Each of these variables is defined below. We used a discrete soft actor-critic (SAC) algorithm for all agents<sup>73,35</sup>.

For each genetic line, 20 SAC agents were independently trained offline on the same data pool.

## VARIABLE DEFINITIONS

**OBSERVATIONS.** Every camera image was preprocessed into features known to be relevant in *C. elegans* behavior<sup>110</sup>. We used pixel coordinates  $(x, y)$  of the animal’s centroid location in the image, with target coordinates subtracted from the centroid. During training the target was always assumed to be  $(0, 0)$ , with coverage over the plate provided by random translations and rotations. Body angles were measured relative to the  $+x$ -axis and head angles relative to the body angle. Body angles were computed by fitting a line to a skeletonized worm image and head angles were computed through template matching.

We performed head/tail identification by assigning the head label to the endpoint that was closest to the head endpoint in a previous frame. To handle reversals, a common behavior in freely moving animals, the overall movement vector over 10 s was compared to tail-to-head vectors during the same window of time. If the vectors pointed in different directions, head and tail labels were switched. Before each evaluation episode, 5 s of frames were collected to assign the first head label again by comparing movement vectors to tail-to-head vectors. Angles were converted to sine and cosine pairs to avoid angle wraparound issues. 15 frames (5 s at 3 fps) were concatenated together for a single observation. Coordinates were normalized so their means in each 15-frame observation was within  $[-0.5, 0.5]$ . An observation  $s_t$  corresponding to time  $t$  was thus comprised of  $6 * 15 = 90$  variables:

$$f_t = (\sin \theta_t^{body}, \sin \theta_t^{head}, \cos \theta_t^{body}, \cos \theta_t^{head}, x_t, y_t) \quad (1.1)$$

$$s_t = (f_{t-14}, f_{t-13}, \dots, f_t) \quad (1.2)$$

Above,  $f_t$  denotes the tuple of variables for the frame at time  $t$ . See Figure 1.1D for a diagram



defining the head and body angles.

**ACTIONS.** An action at time  $t$ ,  $a_t$ , was defined as a choice between the options “light on” or “light off,” denoted by a binary 0 or 1 signal.

$$a_t \in \{0, 1\} \quad (1.3)$$

We did not place any constraints on actions, as all ensembles learned policies with overall light exposure that was under 50% of the time.

**REWARDS.** Reward  $r_t$  was based on the target-finding task and defined as the distance moved toward the target between the time of the action  $t$  and 15 frames (5 s) after the action (Figure 1.1C).

$$r_t = \sqrt{(x_t - x_{target})^2 + (y_t - y_{target})^2} - \sqrt{(x_{t+15} - x_{target})^2 + (y_{t+15} - y_{target})^2} \quad (1.4)$$

A target region was defined as a circle of radius 30 pixels (625  $\mu\text{m}$ ). If the animal was within the target region, the calculated reward was replaced by a constant reward of 2. All other rewards were scaled by a factor of 2 to normalize values and facilitate training.

## TRAINING

As in standard reinforcement learning, SAC searches for a policy  $\pi(a_t|s_t)$  for an environment with a transition distribution  $\rho_\pi$ .  $\pi(a_t|s_t)$  is the probability of taking an action  $a_t$  given an observation  $s_t$ . Here we also make explicit the dependence of  $r_t$  on  $s_t$  and  $a_t$ . SAC deviates from the standard goal of maximizing the return, or expected sum of rewards over time,

$$\sum_t \mathbb{E}_{(s_t, a_t) \sim \rho_\pi} [\gamma^t r_t(s_t, a_t)] \quad (1.5)$$

Here,  $\gamma$  (fixed at 0.95) is a temporal discount factor that diminishes rewards far into the future. SAC maximizes not only the expected sum of rewards, but also an entropy term weighted by a temperature parameter  $\alpha$ :

$$\sum_t \mathbb{E}_{(s_t, a_t) \sim \rho_\pi} [\gamma^t r_t(s_t, a_t) + \alpha H(\pi(\cdot|s_t))] \quad (1.6)$$

The sum now contains an added entropy term  $H$  of the policy  $\pi(\cdot|s_t)$ , scaled by a temperature parameter  $\alpha$ .  $\pi(\cdot|s_t)$  signifies the policy function  $\pi$  over all possible events. We used a discrete version of SAC with automatic entropy tuning (see code for implementation<sup>124</sup>).

**DATA AUGMENTATION.** Once data were collected, they were stored in a memory buffer as tuples:

$$m_t = (s_t, a_t, r_t, s_{t+15}) \quad (1.7)$$

At each training step, a batch of 64 memory tuples were randomly drawn from the buffer and independently augmented by a random translation and rotation. First, the tuple was centered such that the average of the location coordinates were at the origin, (0, 0) pixels. Then a location within a  $\pm 900$ -pixel square (comparable to the size of the evaluation arena) was drawn from a uniform distribution and the coordinates recentered around that location. An angle was likewise chosen from a uniform distribution [0360 and added to the measured angles in the memory tuple.

| <b>Agent parameters</b>       | <b>Value</b> |
|-------------------------------|--------------|
| Temporal discount factor      | 0.99         |
| Learning rate                 | 0.001        |
| Neurons per hidden layer      | 64           |
| Layers per network            | 2            |
| <b>Critic-only parameters</b> | <b>Value</b> |
| Target smoothing coefficient  | 0.005        |
| Target update interval        | 500          |

Table 1.2: **Agent parameters.** Adam was used as an optimizer for actor and critic networks as well as automatic temperature tuning.

TRAINING DETAILS. See Table 1.2 for architecture and hyperparameter choices. 20 agents per genetic line for Lines 1-3 were trained independently on the same memory buffer for 20 epochs of 5000 steps each. See Figure S5 for an example of training progression for individual agents on Line 2. For Lines 4-6, we found greater agent policy instability during training (Figure 4B). In these cases, the animals’ responses to optogenetic modulation were less tightly coupled to target navigation. We therefore trained 30 agents for Lines 4-6. Each ensemble was trained for a minimum of 20 epochs of 5000 steps. We then inspected policies visually to check that they satisfied two conditions. First, the ensemble policy needed to be non-trivial, or not always-on or always-off. Second, the policies needed to be fairly symmetric about the origin when plotted with body angles relative to target, as they should have been given the uniform random translations and rotations during training.

Minibatch size for all agents was 64. Weights were initialized using Xavier uniform initialization and biases were initialized at 0. We tried dropout and weight decay on actors, critics, or both, and found that none of these regularizers helped enough to compensate for the need to choose more hyperparameters (see Figures 1.3-1.5).

Independent agents were trained such that the randomly taken action  $a_t$ , reward  $r_t$ , and the associated states  $s_t$  and  $s_{t+15}$  were used to learn a state-action value function. This is called a Q-function and was learned by the critic network. The actor network then learned a policy that was the expo-

mental of the Q-function. See Haarnoja et al., 2018<sup>73</sup> for details.

ENSEMBLES. Once the 20 agents for one ensemble were trained, they were combined by taking the average of their action probabilities and setting a threshold at 0.5. That is,

$$\pi_{ensemble}(a_t|s_t) = \frac{1}{N} \sum_n^N \pi_n(a_t|s_t) \quad (1.8)$$

where  $N = 20$ . If the average probability  $\pi_{ensemble}(a_t|s_t) \geq 0.5$ , then the light was on at that timestep. 3-5 random seeds were run for each genetic line, and the final ensemble was chosen based on inspection of visualized agent strategies.

#### COMPUTE RESOURCES.

All training was done on the FASRC Cannon cluster supported by the FAS Division of Science Research Computing Group at Harvard University. Every agent was trained on a compute node with one of the GPUs available on the cluster: Nvidia TitanX, K20m, K40m, K80, P100, A40, V100, or A100.

#### 1.3.2 EVALUATION

All experiments involved a single animal placed on a 10 cm-diameter NGM plate with a 4cm-diameter filter paper barrier soaked in copper (II) chloride. All animals were cultured on food with ATR and were thus sensitive to optogenetic perturbation. Animals were switched out for a new one after each evaluation episode.

STANDARD EVALUATION. Animals were placed in the center of the field. A target was randomly chosen among top, bottom, left, and right options (Figure 2B). The experiment with agents were run for 10 minutes each at 3 fps. At the end of the experiment, animals were switched out.

For controls without the agent, animals freely moved on the plate and were recorded for 10 min. A random target was assigned to compare controls to trials with agents.

For controls with random light exposure, the idea was to make sure that light exposure alone was not responsible for more movement, which could lead to an increased rate of success. Once all trials with agents had been run, the proportion of time where the light was on was calculated for each genetic line. These proportions were 0.4647 for Line 1, 0.2896 for Line 2, and 0.3844 for Line 3. Animals were recorded while light decisions were made every 1 s, with the probability of light on according to the genetic lines listed.

For Line 4 (Pan-Arch), due to concerns about phototoxicity, the evaluation was restricted to 1 s light pulses with 4 s rest periods between them.

#### 1.4 ADDITIONAL NOTES

##### 1.4.1 DATA AVAILABILITY

All processed animal tracks used to generate figures are available at <https://github.com/ccli3896/RLWorms.git>.

##### 1.4.2 CODE AVAILABILITY

Analysis code and training code are available at <https://github.com/ccli3896/RLWorms.git>.

DOI: 10.5281/zenodo.11002033

##### 1.4.3 ACKNOWLEDGMENTS

We thank Surya Bhupatiraju for discussions about reinforcement learning and comments on the manuscript. We thank Timothy Hallacy and Abdullah Yonar for guidance in *C. elegans* experiments and Cory McCartan for input on statistical analyses. We thank Kenneth Blum, Cengiz Pehle-

van, Giri Anand, Alexandru Bacanu, Benjamin Brissette, Dianna Hidalgo, Roya Huang, Heitor Megale, William Weiter, Yusuf Ilker Yaman, Vincent Zhuang, and Steven Zwick for comments on the manuscript.

This work was supported in part by NIGMS grant 1R01NS117908-01 (SR), Dean's Competitive Fund from Harvard University (SR, CL), NIH R01EY026025 (GK), the Fetzer Foundation (GK), and an NSF GFRP fellowship (CL).

Figure 1.9: **Poster presented at the Harvard Molecular and Cellular Biology retreat in 2021.**  
I include the poster here because I had a lot of fun making it.

Figure 1.9: (continued)

CHENGLIANG LI | GABRIEL KREIMAN | SHARAD RAMANATHAN

# DEEP REINFORCEMENT LEARNING ON C. ELEGANS

We want an algorithm  
to learn how to control the  
C. elegans nervous system.  
We aim to combine  
biological and artificial  
neural networks.

## SETUP

First, we test the system using a key interneuron and a known deterministic policy from previous work (Kocabaş et al., 2012).

REAL DATA, 5 MINUTES EACH

WE GENETICALLY EXPRESS LIGHT-SENSITIVE CHANNELS IN SUBSETS OF C. ELEGANS NEURONS. A COMPUTER RECEIVES SIDES (LEFT OR RIGHT) OF THE ANIMAL AND CAN TURN A LIGHT ABOVE THE ANIMAL ON OR OFF. THEN, THE COMPUTER OPERATES A CLOSED-LOOP SYSTEM THAT ACTIVATES THE ANIMAL'S NEURONS BASED ON ITS STATE.

## THEORY

WE USE AN ENSEMBLE OF OFFLINE, OFF-POLICY, ACTOR-CRITIC AGENTS WITH DOUBLE Q-LEARNING AND REGULARIZED ACTOR NETWORKS.

## PROGRESS

WHAT DOES A GOOD ANSWER LOOK LIKE?

OUR AGENTS GET GOOD ANSWERS

& HANDLE NEW NEURONS.

## PLANS

COMBINED COMPUTATION

MULTIPLE-ANIMAL TASKS

### A REINFORCEMENT LEARNING PRIMER

Reinforcement learning is a type of machine learning where an agent takes actions in an environment to maximize some reward.

The goal is to learn a good policy, or good actions for every state.

In chess, a state would be a board configuration. An action would be a move. The full strategy would be the policy.

In our case, a state is the head and body angle of C. elegans. The action is turning a neuron on or off. Since our response is small, we can visualize the policy in chess performance.

Our agent plays with actions while observing the animal, collects reward, and gradually learns better policies.



## 1.5 CONCLUSION

Overall, in this chapter, I've discussed how I built the reinforcement learning/*C. elegans* system and shown that it works on a few genetic lines with various sets of optogenetically modified neurons.

Research in this chapter is on bioRxiv (Li et al <sup>1,28</sup>). After I got it to work, my advisors and I came up with a few ideas on how we could use and study this new system. These ideas are the subject of the next chapter.





*Experience is the name everyone gives to their mistakes.*

Oscar Wilde

# 2

## Questions answered and unanswered

RECALL THAT THE GOAL OF THE WHOLE ENDEAVOR with the cyborg worms was to help me figure out how to build a small but fully functional brain. As hundreds of hours passed watching my machines at work, juxtaposed with the animals wriggling about on their agar plates, I began to realize the system I had built would not help me achieve my original goal. However, my advisors and

I did find a few use-cases for the RL-animal system I had built, which I think were still interesting, if not the intended outcome. These use-cases are the topic of Sections 2.1-2.5, and the reasons I did not find these use-cases very satisfying in Section 2.6.

## 2.1 MAPPING NEURONAL POLICIES

The first use case for the RL-animal system was as a technique to probe neurons. We found that we could train agents on a set of neurons, and after training, send whatever simulated animal configurations we wanted through the agent. We could read out the probabilities of taking actions (light on or off, corresponding to the activity of the connected neurons) and make policy maps that told us something about how the neurons were involved in the tested behavior.

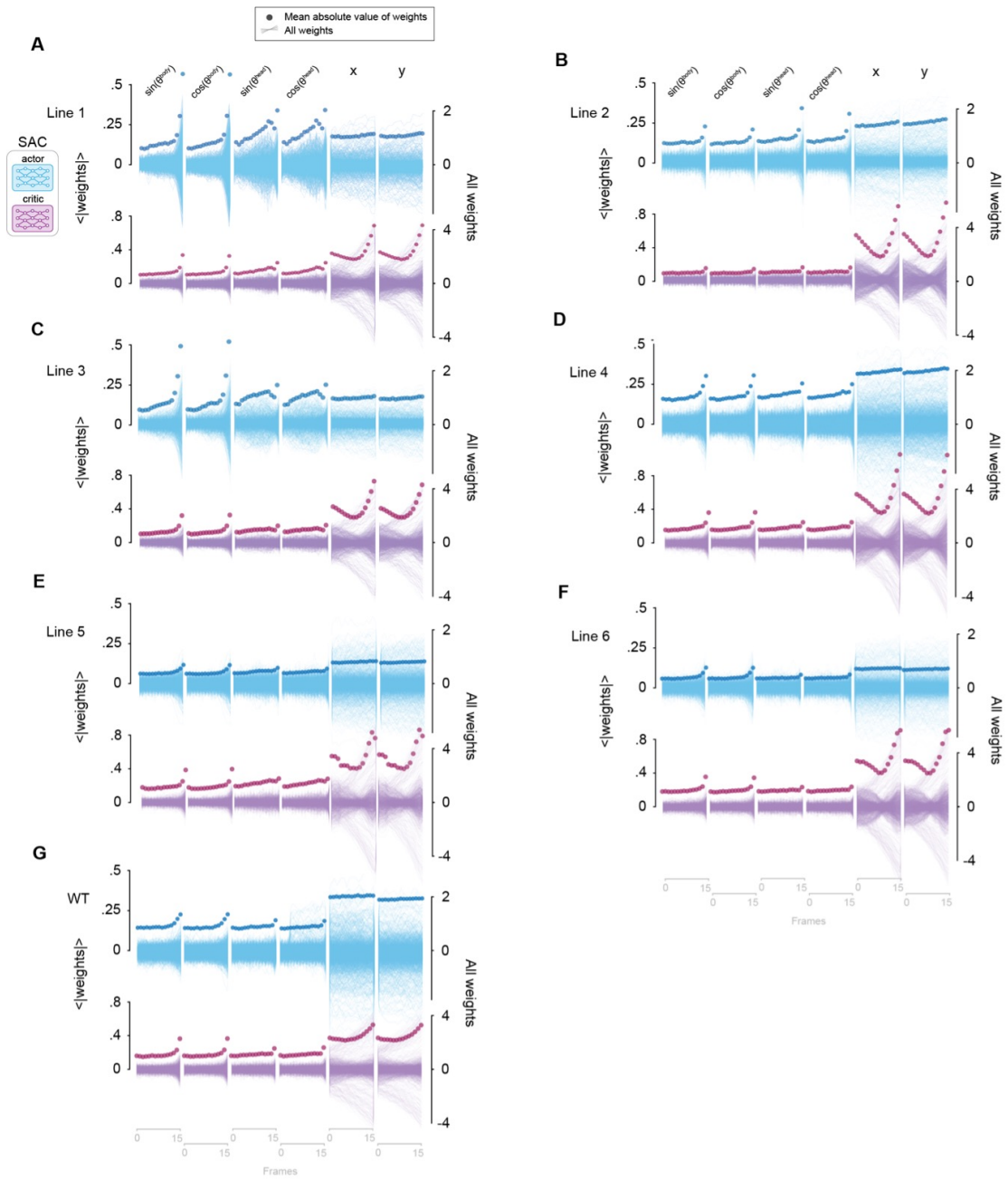
### 2.1.1 AGENT STRATEGY VISUALIZATION.

To visualize agent decisions, we simulated animal states in a smaller space than the full 90-dimensional inputs based on input weight magnitudes. Because the final timesteps of all angle measurements had larger magnitudes than previous timesteps (Figure 1.7H, Figure 2.1.1), we chose to keep input angles constant within each observation and explored the full range of angle possibilities  $[-180, 180]$  in increments of  $10^\circ$  for  $\theta_t^{body}$  and  $\theta_t^{head}$  (36 values each). The 30 coordinate variables  $(x_{t'}, y_{t'}); t - 5 < t' < t$  were always fixed to 0.94 cm to the left of the target, which was exactly half the maximum distance used for random translations during training. In total, 36 head angle values  $\times$  36 body angle values gave rise to 1296 different input observations, each of which were given to an agent ensemble that then output the decision probabilities recorded in the resultant action probability matrix.

Figure 2.1: **Final timesteps for angle variables have larger average magnitudes for agents trained on all lines.**

(A-G) Weights in the first layer of the actor (top) and critic (bottom) networks in all 20 SAC agents of the ensemble trained on Lines 1-6 and wild type animals (G). Plotted as in Figure 1.7H. Raw weights are shown in lighter traces while the mean absolute values are plotted in darker circles to show weight trends.

Figure 2.1.1: (continued)



### 2.1.2 AGENTS LEARNED STRATEGIES SPECIFIC TO THEIR CONNECTED NEURONS

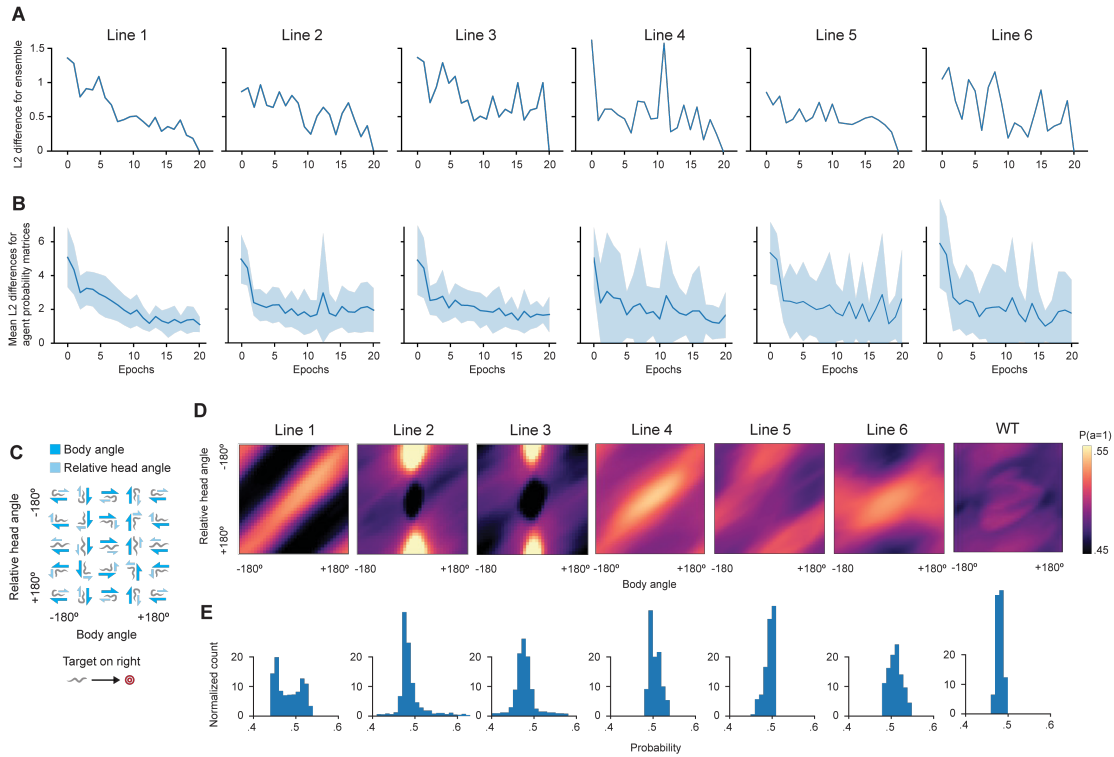
The agent successfully interacted with Lines 3-5, which all involved multiple neurons (Figure 1.8B), including Line 4, which used the entire nervous system<sup>26</sup>. In this instance, the agents took advantage of increased movement after a period of freezing, in contrast to the Line 3 policy that relied on slowing or turning during neuron inhibition. However, the agent failed to find an effective policy for Line 6, where it was coupled to cholinergic muscle excitation in the ventral cord<sup>96</sup>. The standard deviation in the learned policy between agents in the ensembles was noticeably greater for Lines 4-6 (Figure 2.2A-B), which had poorer performance than Lines 1-3 (Figure 1.8B). Together these results show that the choice of sites of integration impact the performance of the animal-agent system.

We visualized policies using the metrics from Figure 1.7I-J to understand how interfaced neurons were involved in target navigation. For reference, figure 2.2C shows animal postures used in mapping agent policies. Policies are plotted in Figure 2.2D. Ensemble action certainty is also visible in Figure 2.2D-E, in which Lines 1-3 have probability values with a wider range than Lines 4-6. This indicates agents are more certain about when to turn the light on or off in Lines 1-3. For comparison, we show an agent trained on wild type animals (Figure 2.2D) with no response to optogenetic modulation. The policies in Figure 2.2D show that agents learned strategies tailored to the neurons they interfaced.

### 2.2 AGENTS PREDICTED SIMILARITIES BETWEEN NEURAL CIRCUITS

Broadly, there were three strategies represented by Lines 1 and 4, Lines 2 and 3, and Line 5 (Figure 2.2D). To understand how agent policies interacted with the nervous system, we focused on the most successful lines: 1, 2, and 3. Although the behavior of Line 1 in response to blue light is mostly to move forward and Line 2 is mostly to reverse, policies were not merely inverses of each





**Figure 2.2: The system learned to navigate different optogenetic lines to a target with neuron-specific strategies.**

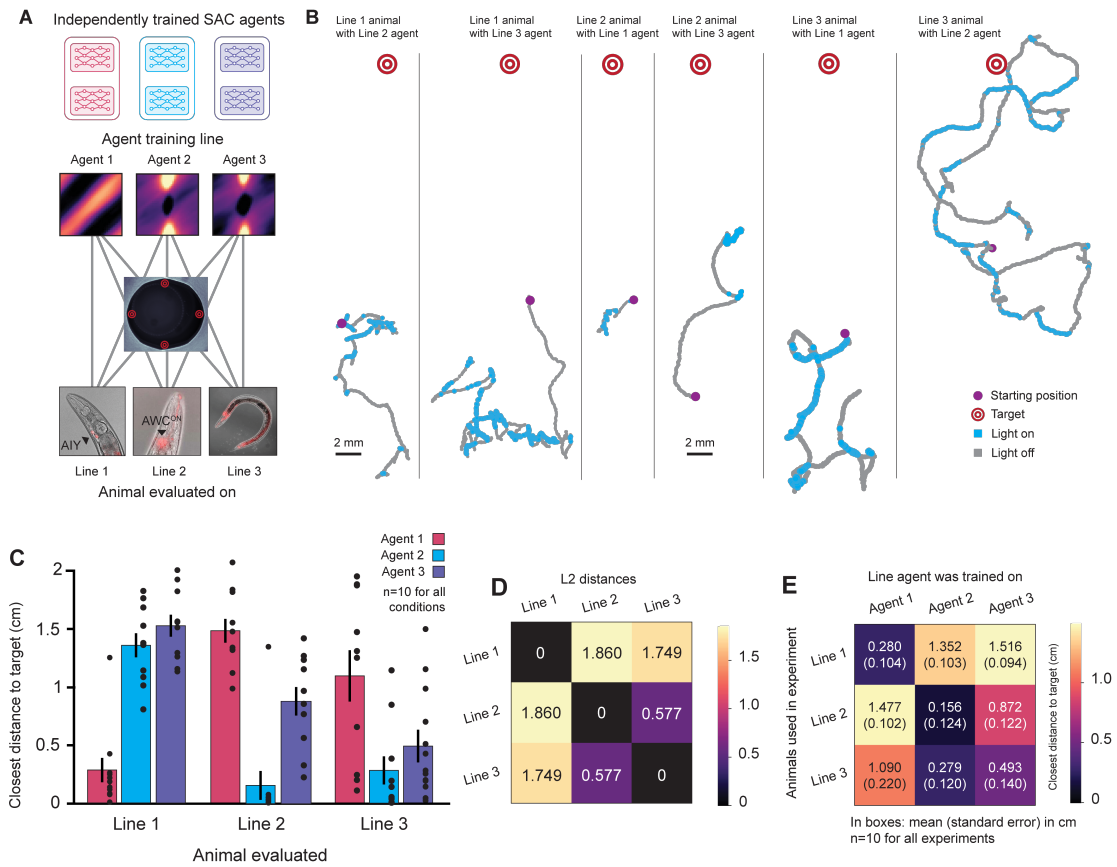
- (A) L2 differences between final ensembles and ensembles at each epoch during training.
- (B) Mean L2 differences between individual agents in the ensemble and the final trained ensemble; shaded regions denote SD. Within an ensemble, agents for Lines 4-6 varied more than in Lines 1-3, which is reflected in the narrower range of probabilities for Line 4-6 in B.
- (C) The animal conformation reference plot for agent policies in B (repeated from Figure 1.7).
- (D) All agent policies for Lines 1-6, and an agent trained on wild type data where there was no possible successful policy. Lines 1 and 4, as well as 2 and 3, had similar agent policies.
- (E) Probabilities in D plotted as a histogram. Lines 1-3 had larger ranges, suggesting greater certainty.

other. Rather, agents learned that Line 1 control was dependent on the animal's head angle relative to the target while Line 2 and 3 control depended on specific head and body angle combinations. Despite large differences in Lines 2 and 3 (excitation of a single neuron in Line 2 versus inhibition of multiple neurons in Line 3), training on Line 3 resulted in an action probability matrix that was remarkably similar to Line 2.

To quantify these similarities in learned actions and to assess generalization across different sites of integration, we ran experiments where each agent was tested on each line (Figure 2.3A). The experiments were conducted identically to standard target-finding evaluations. 10 trials of 10 min each were performed for every agent-genetic line combination.

Sample tracks from combinations of agents and animals are shown in Figure 2.3B with average results in Figure 2.3C. To evaluate whether agent policies were predictive of cross-evaluation performance, we measured L2 norm differences of the action probability matrices (Figure 2.3D). As intuitively observed in Figure 2.3D, the policies from Lines 2 and 3 are most similar. The corresponding plot using experimental data from Figure 2.3C is shown in Figure 2.3E. As expected, diagonal entries have low distances to targets; Line 3 animals tested with Line 2 agents also showed low distances.

Results in Figure 2.3E correlated well with predictions based on the similarity of the action probability matrices in Figure 2.3D ( $r^2 = .8578, p = .000334$ ). As expected from the contrast in action probabilities in Figure 2.2D, Line 1 versus Lines 2 and 3, Line 1 did not respond well to agents trained on Lines 2 or 3. For example, when the agent trained on Line 1 was tested with an animal from Line 2, the closest distance reached from the target was about  $1.477 \pm 0.102$  cm, much larger than when tested on Line 1,  $0.280 \pm 0.104$  cm (Figure 2.3E). The closest distance was also comparable to or greater than the control conditions for Line 2 (Figure 1.8B), as the Line 1 agent tended to drive Line 2 animals away from rather than toward targets (p-value<.08, no agent; p-value<.009, random light; one-sided Mann-Whitney U Test). Likewise, neither Line 2 nor 3 animals performed



**Figure 2.3: Agent policies can predict agent performance on other lines.**

(A) An illustration of cross-evaluation experiments, in which agents trained on each of the 3 best-performing lines were evaluated on every other line.

(B) Sample tracks with agent actions for each combination of agent and animal not shown in Figure 1.7, 1.8, or 1.8.

(C) Statistics of closest distance to target for each combination of agent and animal with  $n=10$  per condition. Data are presented as mean values  $\pm$  SEM.

(D) L2 distances between ensemble action probability matrices for each genetic line.

(E) Mean closest distances (cm) to the target in a 10-min evaluation episode is shown with standard error in parentheses. Distances between the ensemble action probability matrices in D correlate with the closest distances achieved in cross-policy evaluation experiments (Pearson's  $R, r^2=.8578, p=.000334$ ).

well on the task when paired with the Line 1 agent. In summary, by comparing action probabilities learned by agents that were trained to couple to specific sets of neurons, we could make accurate predictions about the behavior of these lines under optogenetic control in the target-finding task.

Another interesting finding was that Line 2 and 3 animals were most successful when paired with the Line 2 agent, even though the Line 3 agent was trained on data from the line itself ( $p < .002$ , Line 2 line with Line 2 vs. Line 3 agent;  $p < .04$ , Line 3 line with Line 2 vs. Line 3 agent, one-sided Mann-Whitney U Test,  $n = 10$ ). These results may be explained by higher data quality from the stronger response of Line 2 to optogenetic stimulation (Supplementary Videos 1, 2, 5, 6), reflected in greater action certainties in Line 2 compared to Line 3 (Figure 2.2D). This suggests that training RL agents with less action noise could improve performance in noisy biological environments<sup>85</sup>. Overall, we demonstrate that our system can generate hypotheses about learning in biological environments, with greater access to internal mechanisms (through the artificial network) than an animal's nervous system alone can provide. And since the completion of this work, others have begun to explore the phenomenon of when RL agents tested in noisy environments benefit from not noisy training data, using simulated cases<sup>24</sup>.

### 2.3 COOPERATIVE ARTIFICIAL AND BIOLOGICAL NEURAL NETWORKS

This section explored a relatively novel use-case for our system. Because this was the first time a deep RL agent had been shown to integrate with a living nervous system to achieve a set task, we wanted to probe what the RL was responsible for in terms of behavior, and what the animal was still responsible for. We wanted to see whether agents and animals could achieve tasks in a general way, integrating information flexibly just as animals can on their own, so we evaluated whether agents and animals could transfer abilities from the target-finding task to food search.

### 2.3.1 ANIMALS CAN CORRECT “ERRORS” MADE BY RL AGENTS

For the food search experiments in Figure 2.3.1A-g, a 10 cm NGM plate was prepared with a 4 cm-diameter filter paper circle soaked in 20 mM copper (II) chloride. 5  $\mu$ L of OP50 bacteria were grown for roughly 24 h before experiments.

Each trial lasted 20 min. An animal was placed on one end of the plate with the OP50 droplet at the opposite end. During the 20 min, the same agents trained on random data as in the standard evaluations were set to navigate animals to targets at 0 cm, 0.5 cm, 1 cm, or 1.5 cm away from the edge of the OP50 droplet. For control trials, agents were left off and the animal roamed freely for 20 min.

Success was defined as a binary outcome as in the obstacle experiments. If an animal reached the food within the 20 min trial, it was counted as a success. Out of 270 trials run across all genetic lines involving OP50 droplets (obstacles and food search), only 1 CH1 animal left food after reaching it during a food search trial when the target was placed 1 cm away from the food edge. This trial was counted as a success.

Using the three best-performing lines, we tested whether animals could correct errors made by agents about the location of food. Targets for the RL agent were placed at increasing distances from the edge of a 5  $\mu$ L patch of food (OP50 *E. coli* bacteria) to mimic errors made by the agent (Figure 2.3.1A). Agents were on throughout the experiment, including after animals had reached the target. Animals were tested whether they could reach food in 20 min trials with or without agents. Agents were identical to those used in Figure 1.7-2.3, each line tested with its own agent. For Lines 1 and 2, when targets were 0.5 cm away from food, animals could leave an agent’s target region (a circle of radius 0.0625 cm) and moved to the food in 8/10 trials ( $p < .0004$ ) (Figure 2.3.1B-C). This was significantly different from trials without agent assistance, in which 0 animals reached food in 10 trials for both lines. Line 3 was not as successful with agent assistance (Figure 2.3.1D), likely due to

less reliable control (Figure 1.8B). This suggests that simultaneous modulation of the neurons in this line is not as strongly linked to directed movement as in Lines 1 and 2. In contrast, Line 1 and 2 animals could switch between making decisions based on their own sensory systems or the agents, which were trained to keep animals at targets. Sample tracks for all experimental conditions are in Figure 2.3.1E-G.

**Figure 2.4: Animals with agents can correct errors and generalize to novel situations.**

**(A)** Error-handling food search experiments. An animal was placed at the opposite end of a plate (large purple circle) as a 5  $\mu$ m drop of OP50 *E. coli* bacteria (orange circle). Trials lasted 20 min and success was defined by whether animals reached food. Agents were directed to navigate animals a distance away from food (target location denoted by concentric red circles).

**(B-D)** Proportion of animals that reached food for Lines 1-3, respectively, n=10 for each condition. For Lines 1 and 2, targets up to 0.5 cm away led to significantly better performance than without agents. One-sided permutation tests; \*\*P<.01, \*\*\*P<.001 (with agent vs. no agent; p=.00034 for Line 1 with target at 0 and 0.5 cm from food and Line 2 with target at 0 cm from food; p=.0053 for Line 2 with target at 0.5 cm from food),

**(E-G)** Sample tracks for Line 1-3 animals with agents based on the majority result of trials. Conditions without agents are shown in the fifth columns.

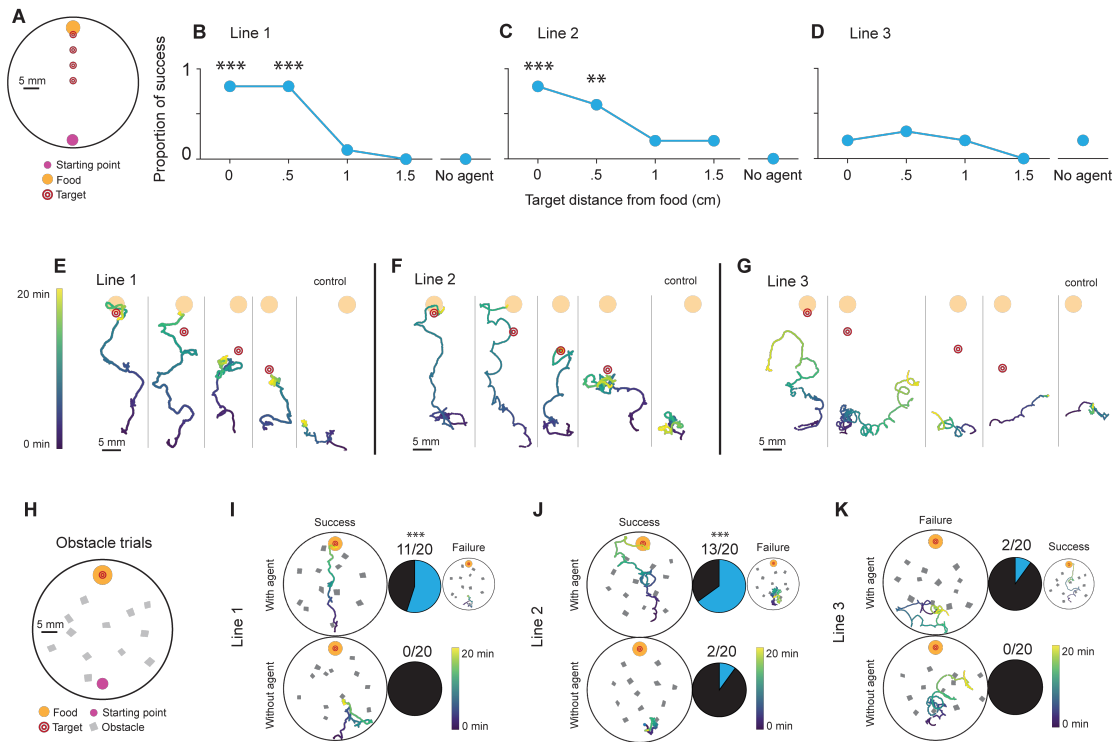
**(H)** Plate used for experiments with obstacles. Twelve paper quadrilaterals with side lengths approximately 2 mm were scattered on the plate. Agents were directed to navigate animals to food and success was determined by whether animals reached food.

**(I)** Sample tracks for Line 1 animals that succeeded (top left) or failed (top right), with control trials without agents (bottom). Success rates shown in pie charts. Animals with agents were significantly more likely to succeed; \*\*\*P<.001; one-sided permutation test; p=.000076.

**(J)** Sample tracks for Line 2 animals. 13/20 animals succeeded with agents and 2/20 without. \*\*\*P<.001; one-sided permutation test; p=.000407.

**(K)** Sample tracks with Line 3 animals. A failed trial in the top left represents the majority outcome. 2/20 animals reached food with agents and 0/20 without (one-sided permutation test, p=.244).

Figure 2.3.1: (continued)





## 2.4 RL AGENTS COULD NAVIGATE ANIMALS IN NOVEL ENVIRONMENTS

We next tested whether the animal and agent could navigate an environment with obstacles to reach food, which represents a novel condition with a biologically relevant goal.

For the obstacle trials in Figure 6H-K, a 10 cm NGM plate was prepared with a 4 cm-diameter filter paper ring soaked in a 20 mM copper (II) chloride solution. We cut 12 pieces of filter paper into quadrilaterals with side lengths 1-3 mm and scattered them on the plate (they were not soaked in copper (II) chloride solution). Obstacle side lengths were comparable to the 1 mm body length of *C. elegans*. Sample arrangements are shown in Figure 2.3.1H-K. Animals cannot cross paper obstacles. Plates were replaced with new obstacle arrangements every 5-10 trials. 5  $\mu$ L of OP50 bacteria were grown on one side of the plate for roughly 24 h before experiments.

Each obstacle experiment was a 20 min trial. A single animal was placed on one end of the plate as in Figure 6H, with the food droplet on the other end and the obstacles in between animal and food. Trained agents (the same agent ensembles used in standard evaluations) were run on the genetic line they were trained on for 20 min. Agents were not retrained to handle obstacles. Control trials had no optogenetic manipulation; that is, the animal was allowed to freely roam the plate with obstacles and food for 20 min. Success was defined as a binary outcome, indicating whether an animal reached food during the trial. This was a challenging task because animals had to use their sensory and motor systems to navigate around obstacles, while agents had to navigate animals to food despite noisier movements.

Line 1 and 2 animals performed well in this new environment (Figure 2.3.1I-J, p-value<.0001, Line 1; p-value<.0004, Line 2; permutation tests). Line 3 was not as successful (Figure 2.3.1K); overall, agents could navigate Line 3 animals closer to targets but could not achieve more difficult food search tasks. For Lines 1 and 2, however, these data provide evidence that our system displays

cooperative computation between artificial and biological neural networks to improve *C. elegans* food search in a zero-shot fashion in novel environments.

## 2.5 SUMMARY OF THE RL-*C. ELEGANS* PROJECT

We presented a hybrid system that used deep RL to interact with an animal’s nervous system to achieve a task, following a reward signal. Agents customized themselves to specific and diverse sites of neural integration, and the combined system retained the animal’s ability to flexibly integrate information in novel environments. Importantly, we could use the same architecture and training process in all lines. Our results did not depend on the number of neurons that agents were interfaced with, nor whether interactions were excitatory or inhibitory, although a failure to learn (as in Line 6) shows the importance of the particular neural circuit under control.

In previous work, brain-machine interfaces have allowed animals to control machines through neural recordings<sup>4,192,187</sup>. Conversely, supervised optogenetic manipulations have taken control of *C. elegans* neurons or muscles to turn the animal into a passive robot<sup>110,49</sup>. In contrast to both of these types of artificial-biological neural interactions, our work integrated a living nervous system with an artificial neural network, automatically discovered tailored neural activation patterns, and did so in a way that allowed computations from both networks to drive animal behavior.

We could then map out patterns of neural activity that were sufficient to drive specific behaviors, enabling us to learn about and compare the roles of different sets of neurons in producing the same behavioral outcome. Mapping out neural policies was possible not only for sets of neurons that were previously well-understood, but also sets that were not. We focused here on navigational tasks, which constitute a central aspect of worm behavior, but our method for learning and visualizing agent policies can broadly be used to learn information about animal behavior using other biologically relevant features besides the particular state space we chose.

It would be interesting for future work to test our method in larger state and action spaces, as

one would find in animals with larger nervous systems than *C. elegans*. Deep RL has already solved complex simulated tasks in high dimensional spaces with large numbers of parameters<sup>179,180,177,198</sup>. That, in addition to our work showing that deep RL is flexible to the site of integration, suggests its potential for use with larger animals whose nervous systems are more variable between individuals. Also, due to broad applicability of the RL framework, the algorithm can be applied to any other behavioral task with a measurable reward function. Overall, our study opens new avenues for using deep RL to understand neural circuits, train in biologically relevant real-world environments, and modulate animal behavior.

## 2.6 QUESTIONS UNANSWERED

BUT SOMETHING WAS MISSING. My goal hadn't been to create some new brain-computer interface; my goal had been to extract some hints as to how I might create a small brain.

At the close of the project, around the end of my third year, my advisors and I began to brainstorm what might be next. We came up with some ideas: maybe we could train agents to get animals to do more complicated movements. Maybe we could go image-to-policy. Maybe we could learn some multi-animal swarming behavior, get the agent to compute ever more complicated functions of locations, posture, whatever, to get the worm to act in a more “intelligent” way than it did with its existing 302 neurons.

But all I saw ahead was a series of engineering projects of increasing complexity. It wasn't clear whether any of them would help me understand how a worm worked, or how I might construct one from scratch using general, fundamental principles. If I truly expected the RL agent to act as a new branch of the *C. elegans* nervous system, what should it look like, based on what we know about living brains?

My discomfort came from the suspicion that, above all, **I should not have had to tell the RL agent what it was responsible for.** The main issue with the RL-animal system—and really, with the use of RL as a model or substitute for animal behavior at all—was that before I connected an RL agent to a nervous system, I had to decide what its reward function was going to be. In Chapter 1, I described how I designed a reward function that increased when animals moved closer to a set target location. But in a biological nervous system, neurons are not assigned tasks when they are born.

I wondered about what I would expect if I'd somehow gotten the worm to grow its own neuron instead of my adding an artificial set of them—what would the neuron do? The addition of neurons happens all the time in real animals. Throughout development, neurons are formed and somehow

coalesce into a working, thinking organ. Even after adulthood, the brain grows new neurons with neurogenesis.<sup>143,53</sup> Somehow, the new neurons figure out their own roles. And whatever they figure out, it must happen via *local* mechanisms, which, when put together with the many also local mechanisms of every other neuron in the nervous system, produce some kind of computation and behavior.

The beauty of natural intelligence is not its ability to achieve the tasks I want it to achieve. Its beauty lies in its capacity to decide for itself. If I built any kind of RL agent, or supervised learning agent, I would have to decide beforehand exactly what it was going to learn, because I would have to first write a loss function for it to optimize. If that is the case, how do neurons allocate themselves to different tasks, and how do they change tasks when the organism requires it, like we see in animals all the time?<sup>15,176</sup>

### 2.6.1 THE PROBLEM WITH BACKPROPAGATION

I began to suspect that the misalignment between natural and artificial intelligences was more than a matter of finding or designing the right reward function. I began to wonder whether the whole problem might stem from the fact that almost all artificial neural networks today, supervised and reinforcement learning algorithms alike, are trained using backpropagation.<sup>118,166</sup>

Artificial intelligence research has pushed the limits of backpropagation. At the same time, we all know it has a few fundamental restrictions. Backpropagation cannot be implemented in a living neural network and is thus biologically implausible.<sup>131</sup> But more importantly, it is inflexible. To train a network using backprop, one must know the answer in the form of a loss function. In contrast, biological intelligence can adapt. It not only learns new solutions to a given problem, it can switch to a new problem or even invent one during play.

Play perhaps embodies the problem with backpropagation more than any other animal trait.

It has proven extremely hard to define and study,<sup>30,202\*</sup> but can be generally defined as an activity done for enjoyment, often without clear utility. We see instances of play across the animal kingdom,<sup>30,48,59</sup> and it is possible that in the ones for which we have not observed play, we also do not know how to characterize enjoyment. Without substantial and targeted modification, it seems impossible for backpropagation-based training to generate play behavior: play requires purposeless or self-invented exploration, and backpropagation requires a purpose (i.e. a loss). I talk about my thoughts on play further in Section A.2.4.

It appeared to me that the problem with backpropagation was never just that it was biologically implausible<sup>131</sup>. I couldn't see how the richness of animal behavior would suddenly pop out if we only found a version of backprop that neurons could feasibly implement. Something bigger had to change, and I thought for a very long time as to what it was. After I reached a tenuous conclusion, I found that many other people had reached it long before I had, and I list a few books related to the topic in Section 2.6.3.

## 2.6.2 CHASING COMPUTATIONAL EMERGENCE

There are clues in the literature as to what kind of thing a nervous system really is. Here I describe what are, in my opinion, two of the most compelling clues: split brain experiments in humans, and the wild reality of nerve nets in the cnidarian *Hydra vulgaris*.

### SPLIT-BRAIN EXPERIMENTS IN HUMANS

In the memoir “Tales From Both Sides of the Brain,”<sup>67</sup> Michael Gazzaniga describes his work with split-brain patients in the ‘70’s. These were patients with epilepsy where, in last-ditch procedures, surgeons had cut their corpus callosa, the thick bridges connecting their neural hemispheres, and

---

\*As the biologist E. O. Wilson wrote about play, “no behavioral concept has proven more ill-defined, elusive, controversial, and even unfashionable.”<sup>202</sup>

sliced their brains cleanly in two. The procedures often cured the patients' epilepsy, and there were few debilitating consequences for them.

Gazzaniga and colleagues found that once the hemispheres were separated, you could communicate to them individually. Each side, left or right, gets information from the opposite visual field, so if you gave written cues to the patient's left visual field, for instance, only the right brain hemisphere would receive them. It was like talking to two separate people in the same body.

In one experiment, Gazzaniga and a colleague, Joseph LeDoux, played a matching game with a patient named P.S. They showed two different pictures to each hemisphere and asked P.S. to point to cards that matched those pictures. First, they showed the left hemisphere a chicken claw, and the right hemisphere a picture of a snowy scene.

"The left hand pointed to a card picturing a chicken and the right hand to a card picturing a shovel," wrote LeDoux in his recollection of the experiment.<sup>119</sup> They asked P.S. to explain.

"P.S. explained his choices saying he saw a chicken claw so he picked the chicken," he wrote, "and you need a shovel to clean out chicken shit in the shed."

In patients with recent operations, only the left hemisphere of the brain can speak. The left hemisphere, then, is left to interpret the right hemisphere's actions with as little context as though they came from a stranger, even though they lived in the same body. The left hemisphere of Patient P.S.'s brain fully believed in its explanation. "The brain was like an old couple who had lived together for years," wrote Gazzaniga in his memoir. It was as though the two hemispheres had "worked out a way to live together and yet be separate." And if you looked at it in reverse, it was like one person had emerged from the joining of the two.

#### THE CURIOUS RECONSTRUCTION OF *HYDRA VULGARIS*

I often wonder what may have happened had Sydney Brenner, back in 1965,<sup>27</sup> chosen *Hydra vul-*  
*garis* as a model organism for neuroscience rather than *C. elegans*. The two organisms seem to teach

radically different doctrines about the nature of living neural networks.

In *C. elegans*, each of the 302 neurons have remarkably consistent roles from animal to animal.<sup>40,203,208</sup> It lends itself to a fixed-circuit type of thinking, where neural functions are outputs of literally hardwired connections, and that is the perspective that neuroscience has largely clung to for the last few decades.<sup>173,208</sup>

*Hydra vulgaris* is a different creature entirely. Hydra are a freshwater polyp, a cylindrical invertebrate with a single opening surrounded by tentacles. They can have a few thousand neurons distributed in a nerve net throughout their body, which provide them with reflexes to touch, temperature changes, chemical stimuli, and light.<sup>107</sup> Their behaviors include contraction along the longitudinal and radial axes, stretching themselves out, nodding, catching food and eating it, and somersaulting to move around, sometimes to move away or toward a light source<sup>107</sup>. This last behavior, phototaxis, is a goal-directed behavior that depends on the hydra's level of satiety.

In 2021, Lovas and Yuste studied the nervous system after a hydra was completely dissociated into individual cells. “We report that *Hydra*'s nervous system synchronizes during its reassembly through a two-step process,” write the authors,<sup>132</sup> “in which the initially uncoordinated activity of small groups of neurons clusters into coactive ensembles, concomitant with pronounced local increases in connectivity and neurite outgrowth, followed by the synchronization of these local ensembles across the entire body of the animal and **re-establishment of behavioral rhythms at 72 h**” (emphasis mine).<sup>†</sup> Essentially, hydra were cultured in high-osmolarity media and then decomposed into individual cells by grinding them down with a glass pipette. Then they were left alone on coverslips for three days, and “animals in all experiments eventually regenerated to yield normal hydranths.”<sup>132</sup>

The authors were able to monitor neural activity during reaggregation, too. They saw that neu-

---

<sup>†</sup>It is unbelievable to me that this paper, released in 2021, has only 13 citations as of June 03, 2024. Several of these are one of the authors. It's rather telling about where attention is directed in the field.



ral activity increased at first and then decreased after neurons had restructured themselves. The strength of functional connections increased over time, and so did the “burstiness” of neural firing patterns. These regeneration abilities were astounding to learn about, as was the ability of a hydra to survive through the gradual removal of its neurons, and then finally *without a nervous system at all*.<sup>76</sup> It seems you can pick the nerve net of a hydra apart and then build it back up like Legos. These attributes also push against the fixed-circuitry view of a nervous system. How might neuroscience be different if we had focused on these animals instead of the nematode? What would our perspectives on nervous systems be if we had taken this organism with its entirely decomposable and rebuildable nerve net as our foundation model?<sup>‡</sup>

#### WHERE TO GO FROM HERE?

Both of these examples, I believe, suggest that a nervous system is not a computational unit with a global reward function. Instead, behaviors emerge from the interaction of many small connected units. There are several somewhat disconnected fields that have explored how to build an emergent computational network, which I think are largely the bodies of research in Hopfield networks,<sup>13,88,116,160</sup> autopoietic computational systems,<sup>28,94,162</sup> and the free-energy principle and active inference.<sup>62,156</sup> I’m sure there is more that I’m unaware of. But so far, none of these fields have managed to produce actively behaving algorithms that appear to even have the capacity to approach the deep learning models I mention at the start of Chapter 0.

After looking around for a while, I almost accidentally stumbled upon a body of work on predictive coding,<sup>22,141,140,182</sup> related to the free-energy principle literature. I became acutely interested in these models of predictive coding because I hadn’t yet seen any comparable algorithms that em-

---

<sup>‡</sup>It is admittedly not so simple as I’ve presented: there is evidence that neurons in regenerating hydra differentiate into neuronal cell types before activity is reestablished.<sup>54</sup> But for the case where this was shown, it was many neurons of the same type that were reintegrated into a functional circuit, in contrast to *C. elegans* where every neuron has a fixed label and many (not all!) neurons have fixed roles.

phasized locality, prediction,<sup>§</sup> and a possible connection to action at the same time. As a bonus, Millidge<sup>142</sup> managed to show that their version of predictive coding could approximate backpropagation. Even though I have been saying my goal is not to rehash backpropagation, this showed that predictive coding could be mapped to it under certain conditions. I wondered whether there were ways to change predictive coding so it could take actions, be autonomous, and still take advantage of the computational efficacy of backprop. For all my qualms, backprop remains the most successful learning method today.

But before I began on that research, presented in Chapters 4 and 5, I worked on another idea: automatic task allocation in branched network architectures. This is the topic of the next chapter, and it helped convince me that I really could relinquish control over the artificial neural networks I was training, like I realized I would have to do in a model of a brain that was truly based on emergence.

### 2.6.3 BRIEF LIST OF BOOKS

This is a very short but useful list of books related to the idea of the brain as an emergent system. There is much more excellent writing on this topic than the books I've listed, but these could be good places to start.

Buzsáki, G. M.D., 2019. *The brain from inside out*. Oxford University Press.

Hawkins, J., 2021. *A thousand brains: A new theory of intelligence*. Basic Books.

Pessoa, L., 2022. *The entangled brain: How perception, cognition, and emotion are woven together*. MIT Press.

---

<sup>§</sup>Neuroscientists generally believe prediction to be how neurons operate.<sup>31,57,74,134</sup>





I'm standing there, my sorry human eyes overwhelmed by light and detail, while the hawk watches everything with the greedy intensity of a child filling in a colouring book, scribbling joyously, blocking in colour, making the pages its own. And all I can think is, *I want to go back inside.*

Helen Macdonald, *H is for Hawk*

# 3

## How I learned to stop worrying and trust the networks

IN THE SUMMER OF 2021, I MADE A BET with someone, a brilliant software engineer who worked in reinforcement learning. I was at Woods Hole, a tiny town on the very tip of the cape of Mas-

sachusetts for a summer course run by the Center for Brains, Minds, and Machines. I was still in the middle of the *C. elegans* project, but the course gave me a month-long respite and the opportunity to think about other ideas for a while.

At the course I met Arturo Deza, a teaching assistant who was at the time a postdoc with Tomaso Poggio at MIT. He had recently published a paper on how network architectures embedded priors for tasks, namely how convolutional networks embedded useful priors for image recognition. After hearing his tutorial, I wondered how the brain might use this principle to allocate tasks without having to try too hard; that is, how the vertebrate brain could allocate tasks to different brain regions by only specifying some general architectural principles instead of hardwiring in the tasks.\*

My bet with the RL engineer was that if I gave a network with two distinct branches two tasks at the same time, those branches would specialize based on how well their architecture was suited for each task. His prediction was that if one of the branches had a better overall prior, like a convolutional architecture, then both tasks would go to the convolutional side, or that both tasks would spread themselves out over the network to utilize the full computational capacity of the network.

I stayed up late one evening at Woods Hole running the first few tests because I wanted to win the bet.† Over the next few months, Arturo helped me turn the quick experiments I ran that night into this chapter, also available online as a NeurIPS workshop paper.<sup>126</sup>

### 3.1 INTRODUCTION

What motivates the brain to allocate tasks to different regions and what distinguishes multiple-demand brain regions and the tasks they perform from ones in highly specialized areas? Here we explore these neuroscientific questions using a purely computational framework and theoretical

---

\*I don't think that hardwiring connections is impossible—see the example of zebrafish vision<sup>11</sup>—but it seems impractical to do that for the entire brain because of its combinatorial complexity. Genes are compact, but not that compact. Also it would be a really inflexible system.

†Which I did, although I've forgotten what I won.

insights. In particular, we focus on how branches of a neural network learn representations contingent on their architecture and optimization task. We train branched neural networks on families of Gabor filters as the input training distribution and optimize them to perform combinations of angle, average color, and size approximation tasks. We find that networks predictably allocate tasks to the branches with appropriate inductive biases. However, this task-to-branch matching is not required for branch specialization, as even identical branches in a network tend to specialize. Finally, we show that branch specialization can be controlled by a curriculum in which tasks are alternated instead of jointly trained. Longer training between alternation corresponds to more even task distribution among branches, providing a possible model for multiple-demand regions in the brain.

Figure 3.1:

**An excerpt from a poster presented at SVRHM 2021 (Shared Visual Representations in Human and Machine Intelligence), a NeurIPS workshop.** <sup>126</sup>.

References in the excerpt:

1. Kanwisher, N. and Yovel, G., 2006. The fusiform face area: a cortical region specialized for the perception of faces. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 361(1476), pp.2109-2128. <sup>101</sup>
2. Kanwisher, N., 2010. Functional specificity in the human brain: a window into the functional architecture of the mind. *Proceedings of the national academy of sciences*, 107(25), pp.11163-11170. <sup>99</sup>
3. Deza, A., Liao, Q., Banburski, A. and Poggio, T., 2020. Hierarchically compositional tasks and deep convolutional networks. *arXiv preprint arXiv:2006.13915*. <sup>46</sup>
4. Fedorenko, E., Duncan, J. and Kanwisher, N., 2013. Broad domain generality in focal regions of frontal and parietal cortex. *Proceedings of the National Academy of Sciences*, 110(41), pp.16616-16621. <sup>58</sup>
5. Garner, K.G. and Dux, P.E., 2015. Training conquers multitasking costs by dividing task representations in the frontoparietal-subcortical system. *Proceedings of the National Academy of Sciences*, 112(46), pp.14372-14377. <sup>65</sup>





Figure 3.1: (continued)

# OUR QUESTION


What might motivate the brain to allocate tasks to different regions?

Certain brain regions are specific for certain tasks. From motor cortex to face perception to written character recognition [1,2],




Studies have provided evidence that the human brain can not only be highly specialized, but be specialized in ways that are consistent between different people. What might lead to these consistencies?

One natural explanation is that local inputs can guide task allocation. But given the same inputs with different branch architectures, how are two tasks distributed?

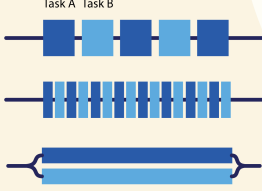


Theoretically, certain neural network architectures are suited to different kinds of tasks [3]. So we hypothesized that the specific architectures of branches can predict how tasks are allocated. See [1 architectural bias].

Specialized brain areas exist. But at the same time, “multiple-demand” regions are involved in many tasks [4]. What separates these areas from the specialized ones?



Specialized regions      Multiple-demand



A clue to this question lies in how humans learn to multitask. While we are generally quite bad at multitasking, it can be done when the two tasks are trained together. The simultaneous training leads to separation in neural substrate in the brain (a functional branching) [5]. Thus, we hypothesized that training protocols would affect the existence of functionally branched networks. See [2 training protocol].

We explore these topics with computational experiments and theoretical insights.

Brain specialization has been an active topic in neuroscience for decades, and has helped us discover many brain regions dedicated to particular tasks across humans despite low-level differences in plasticity (e.g. the Fusiform Face Area<sup>100</sup> that serves a pivotal role for face recognition; though also see<sup>66,7,6,82</sup>). However, we have yet to build a full explanation of why some "multiple-demand" brain systems<sup>58</sup> are involved in a variety of tasks while others are extremely narrow in scope. We do not currently know all the factors that distinguish general and specialized brain regions, nor what causes them to emerge or develop; in addition, they are challenging to study *in vivo* because biological network architectures cannot be easily modified and paired with the proper controls. Consequently, we turn to deep learning, where computational cognitive neuroscientists may now test their ideas on computer models rather than living organisms and can also play the role of a "virtual neurophysiologist" by inspecting artificial neural network activations<sup>210,115,150,75</sup>.

Indeed, the field of machine learning has shown interest in neural network specialization:<sup>199</sup> at OpenAI showed that branching still occurs implicitly in neural networks even when the network architecture does not explicitly bifurcate. On the neuroscience side, examples of previous works on branching in audition include<sup>104</sup> where a branched neural network jointly trained to learn speech and music learned to correlate well with human auditive behaviour. And in vision,<sup>47</sup> found that jointly training networks on a face and object classification task resulted in specialized branching along the hierarchy of the CNNs.

Here we hope to complement observations from the previous studies, and begin identifying the factors involved in branch specialization. In particular we focus on specialization in a precise visual task where the stimuli are Gabor patches (Fig. 3.2) rather than natural images such as those of ImageNet<sup>167</sup>, VGGFace2<sup>33</sup>, THINGS<sup>79</sup> or Places<sup>213</sup>. The benefit of Gabors is that we can design tasks that need not be compositional or hierarchically-local such as object recognition, but can be global (e.g. average color/luminance) and/or order-1 hierarchically local (e.g. orientation)<sup>46</sup>. And

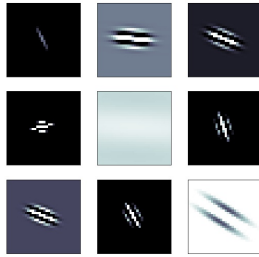


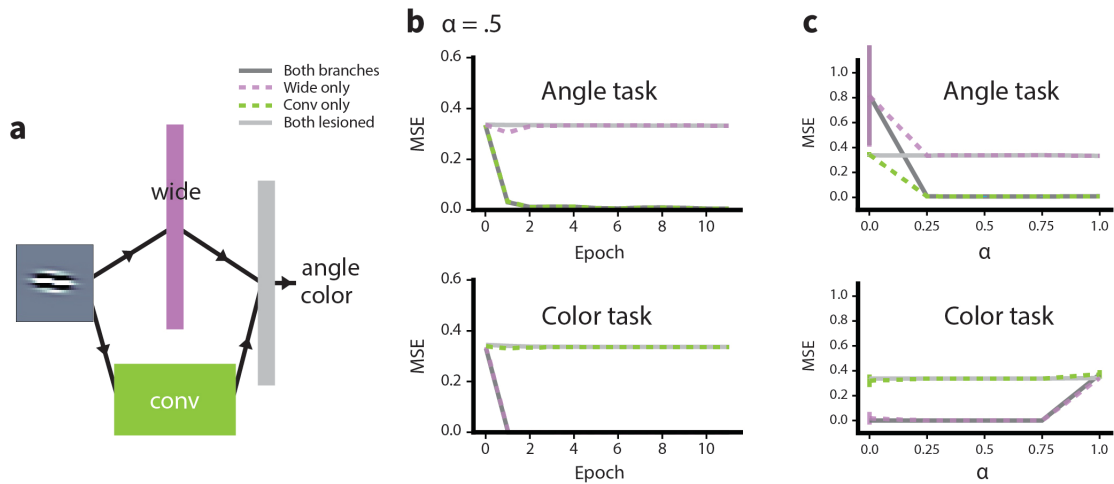
Figure 3.2: **Sample images generated for the Gabor dataset.** See Section 3.5.1 for details.

in contrast to most prior work, we change our tasks, architectures, and training protocols to try to unravel the *causal factors* for branches to specialize rather than only observing this phenomena after training.

The rest of the paper is organized as follows. In section 3.2, we show that branch specialization is robust and happens with diverse or identical branches. In the case of branches with different architectures trained on two simultaneous tasks where each is better suited for only one of the branches, a network specializes in predictable ways that align with the branches' inductive biases. In section 3.3, we show that branch specialization can be controlled by a curriculum learning scheme that alternates task training, and that the faster the alternation rate, the more likely specialization is to occur. In the discussion, we note some neuroscientific implications of our results and describe planned future work on both the mechanisms of branch specialization and its consequences.

### 3.2 TASK ALLOCATION IN BRANCHES CAN BE PREDICTED BY INDUCTIVE BIASES

This section asks how consistently branch specialization occurs, and how much architectural biases affect it. In Figure 3.3, we used a Gabor filter dataset and asked our networks to simultaneously output the images' angle and average color. Our network architecture consisted of two branches (see Figure 3.3a). One was a convolutional network with two convolutional layers followed by two fully



**Figure 3.3: In networks with differently shaped branches, tasks are allocated based on predicted inductive biases.**

**a.** A branched network with one fully connected wide branch and a convolutional branch. Outputs are the angle and color of the  $32 \times 32$  Gabor filter image.

**b.** Mean squared errors (MSEs) of branched network during training in four conditions: intact, with the wide branch only, with the convolutional branch only, and with both branches lesioned. Branches specialize early on in training and do not appear to change afterward.

**c.** Statistics of MSEs for ten random seeds for the experiment in **b**, for different values of  $\alpha$  (see text). Standard deviations shown in error bars. Task weights as defined by  $\alpha$  do not affect branch specialization.

connected layers (see Section 3.5 for details). The other branch was a fully connected network with two fully connected layers. The outputs of both branches are then fed into a linear output layer, from which the two output values for the dual task (angle and color) are read. See the associated code for implementation.<sup>‡</sup> We expected the angle task to be better suited to the inductive biases of the convolutional branch, perhaps requiring edge detection or similar computations that convolutions can more easily learn. Conversely, mean color estimation is a simple average that is better suited to the fully connected branch. In this way, we have designed each branch of the network to have an inductive bias that matches only one of the tasks. In our experiments, however, the fully connected branch has almost three times fewer parameters than the convolutional branch (see Section 3.5), so in one sense, it may be surprising if the fully connected branch learns a task at all.

Results are shown in Figure 3.3b and c. We train the entire network on the dual task but then evaluate branches individually by zeroing out (*i.e.* lesioning) the final outputs of the other branch before it is fed into the linear output layer. We compare those results to ones where both branches were lesioned and where the network is intact. Figure 3.3b shows that training converges quickly and each task is entirely localized to one branch. Over ten random seeds for the training in Figure 3.3b, an intact network had an average MSE of .0074(.0044) on the angle task (standard deviation in parentheses). The fully connected branch alone had an average MSE of .3367(.0045) on the angle task, whereas the convolutional branch alone had an average MSE of .0074(.0044). With both branches lesioned, the average MSE was .3367(.0045). We can conclude, then, that the convolutional branch is responsible for the angle task. These data are shown in Figure 3.3c. Figure 3.3c also tells a similar story for the color estimation task, except that the fully connected network is now responsible for it instead. In ten random seeds, the angle task was always localized to the convolutional branch while color estimation was always localized to the fully connected branch. These allocations align precisely with the inductive biases we initially described.

---

<sup>‡</sup>Code is available at <https://github.com/ccli3896/branches-svrhm>.

We then wanted to see if branch specialization was robust to the two tasks’ relative contributions to loss, so we scaled the losses for both tasks with a convex-combination parameter  $\alpha$  to define a new loss  $\mathcal{L} = \alpha\mathcal{L}_{angle} + (1 - \alpha)\mathcal{L}_{color}$ . We tried a range of  $\alpha$  values from 0 to 1 with the same network architectures in Figure ??a and the same dual task. One might expect that if one task’s relative importance were to increase, a network may allocate more resources to it rather than continuing to split resources evenly. However, we did not see any gradual change in resource allocation. Rather, we saw the same branch specializations as before regardless of task importance, even for losses that were heavily biased toward one task. Furthermore, even when the branched network was trained on only the color or only the angle task (corresponding to  $\alpha = 0$  and  $\alpha = 1$  respectively), one branch was left unused while other shouldered the entire task burden. Thus, relative contributions of each task to the loss did not affect task allocation to branches.

Next, we wanted to know what would happen with two identical branches and two more similar tasks. If branches didn’t have asymmetrical biases for tasks, would they still exhibit branch specialization? We used the architecture described in Figure 3.4a with two convolutional network branches this time, with all else the same as the network in Figure 3.3a. For training we used another simultaneous dual task setup where the input was a Gabor filter and the output was two values: angle and size, where size was set by the parameter  $\omega$  in the Gabor filter generation function (see Section 3.5).

We plot training progress in Figure 3.4b. As before, branches quickly specialize to one task. Because the branches were identical, we used their effect on the size task to label them ”conv 1” or ”conv 2” in all of Figure 3.4 and ordered the branches by their performance on the size task to maintain functional identity. Tasks are consistently allocated to different branches over five random initializations, although in the angle task, both branches were occasionally involved. However, one branch was always more important: in an intact network with  $\alpha = .5$ , the average MSE was .0079(.0036) for the angle task and with ”conv 1” it was .0498(.0832). ”Conv 2” had a significantly larger MSE of .2724(.1229) while lesioning both branches only increased that number to

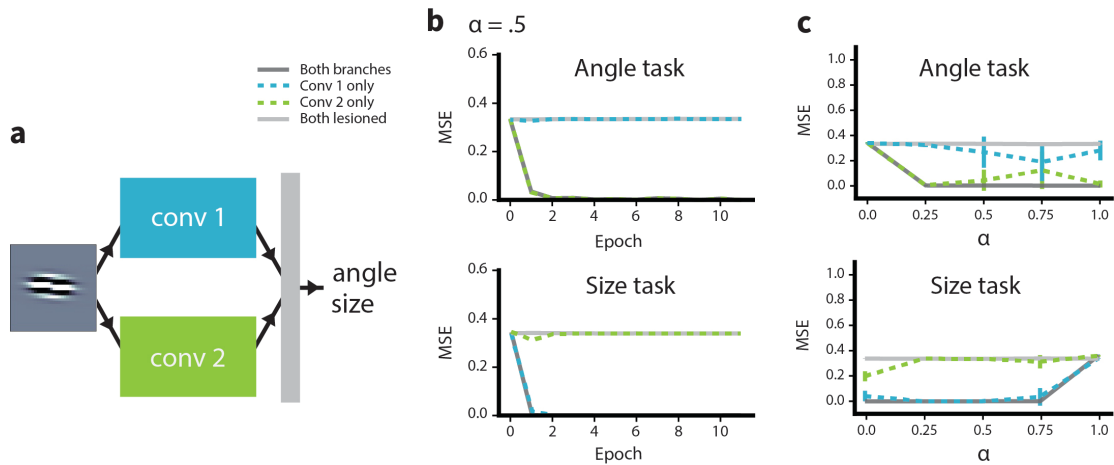


Figure 3.4: **Even with identical branch architectures, networks tend to specialize.**

**a.** Branched network architecture with two identical convolutional branches. In this figure branches are distinguished by functionality after training. “Conv 1” is more important for performance in the size task than “conv 2.” Outputs are now angle and size of Gabor filters.

**b.** Training progress on the simultaneous angle/size task. Plots show performance during training of MSEs for the intact network, “conv 1 only”, “conv 2” only, or both branches lesioned. Again, specialization happens early.

**c.** MSE statistics over five random seeds for different values of  $\alpha$  (see text), with standard deviations in error bars. Complete task separation does not always happen as in Figure ??, but there is clear specialization particularly for the size task.

.3381(.0022). We tried different relative task weightings as in Figure 3.3c, again showing that our results were not too sensitive to the loss function. Overall, branch specialization happened even with more similar tasks and identical branch architectures, so although specialization can be predicted by matching inductive biases to tasks, it is not dependent on task-branch asymmetry.

### 3.3 BRANCH SPECIALIZATION CAN BE CONTROLLED BY CURRICULUM LEARNING

In this section we ask whether branch specialization can be controlled. Looking at individual networks' branch allocations from the experiments for Figure 3.4c (not shown), we noticed that branched networks trained on one task were more likely to use both branches for it. We hypothesized that alternating between tasks could lead to task sharing between branches rather than specialization.<sup>§</sup> To test the hypothesis, we trained the branched architecture in Figure 3.5a on the same Gabor angle and size task as in Figure 3.4. This time, losses came entirely from one task for  $n$  epochs and switched to the other task for the next  $n$  epochs, alternating for 500 epochs. We tested 1, 5, 10, and 20 for values of  $n$ .

For small values of  $n$ , training is more like the simultaneous dual task and we expect branch specialization to happen. Our results for  $n = 1$ , or task alternation between every epoch, are shown in Figure 3.5a. Aside from some epochs where EWC (see footnote) fails to maintain task performance as seen in the spikes in Figure 3.5a-bottom, we observe consistent branch specialization for the angle and size tasks. The MSEs over five random seeds for the angle task, for instance, were .0065(.0017) for the intact network and .0065(.0017) when we lesioned the branch with worse performance on the angle task. When the branch with better performance was lesioned, MSEs were .3392(.0038),

---

<sup>§</sup>To prevent catastrophic forgetting, we used Elastic Weight Consolidation<sup>108</sup>, or EWC during training. EWC is inspired by synaptic weight consolidation in the brain, where synapses that are involved in long-term memories are strengthened and stay strong for days, weeks, or years<sup>38</sup>. In a similar way, when a network is trained on a new task after having already learned another, EWC prevents weights that are important to previous tasks from changing as much as unimportant weights.



compared to both branches lesioned at  $.3392(.0037)$ . Data for the size task are in Figure 3.5c.

When  $n$  is increased to 10, we see more distribution of both tasks over both branches. Figure 3.5b displays the errors from trained networks with all combinations of branch lesions. Now, MSEs for five random seeds for the angle task were  $.0835(.0189)$  for the intact network and  $.1247(.0332)$  with the worse branch lesioned. With the better branch lesioned MSEs were  $.2453(.0330)$ , compared to both lesioned with an average MSE of  $.3397(.0034)$ . On both tasks, both branches have non-negligible contribution to task performance. Results for all tested values of  $n$ , the epochs between task alternation, are in Figure 3.5c. Faster alternation (corresponding to smaller values of  $n$ ) more often led to branches that were entirely responsible for one of the two tasks. For slower alternation, both branches affect both tasks. However, one branch tended to be more important in each task, and this did not change as alternation times grew. Nonetheless, we could partially control the degree of branch specialization by changing the learning curriculum.

### 3.4 DISCUSSION

Understanding how tasks distribute themselves within a neural network has relevance to machine learning research, perhaps to aid in architecture design, and to neuroscientists, to understand why parts of the brain can be highly specialized or be involved in a broad set of functions. We use small networks and a simple task set to try to gain intuitions for larger models and more complex tasks. As a consequence, our conclusions may serve best as guides for larger experiments in the future.

Using a toy Gabor filter dataset and dual task experiments, we show that branch specialization is a robust phenomenon with either identical or different branches. In the case of branches with different architectures that are better suited to one of a set of tasks, task allocation to branches can be predicted based on branches' inductive biases. We also demonstrate that specialization is sensitive to training curricula and if multiple tasks alternate during training, branch specialization can be reduced.

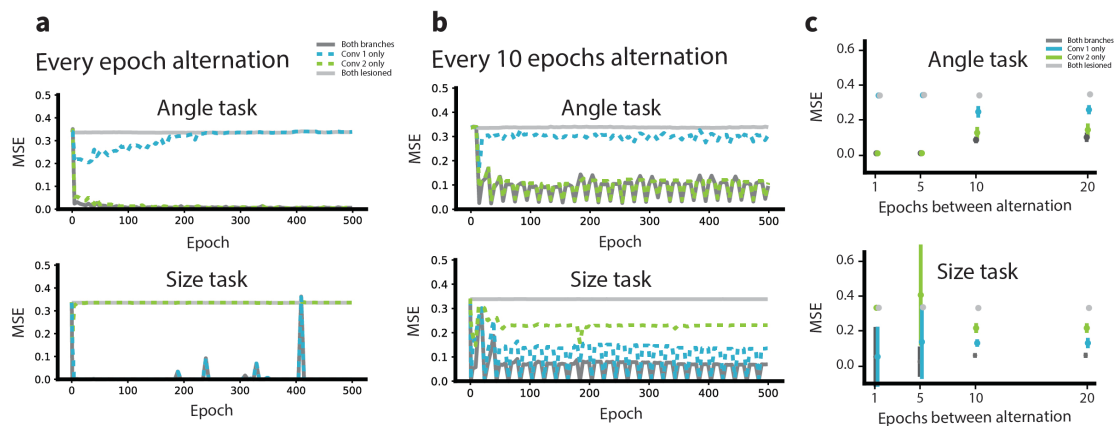


Figure 3.5: **When tasks alternate quickly, network branches tend to specialize. When they alternate slower, tasks are more distributed across networks.**

**a.** and **b.** show training progress with intact networks, one branch only, or both branches lesioned. **a.** is data from alternating between angle and size tasks every epoch and **b.** alternates them every ten epochs. Complete branch specialization happens as usual in **a.** but not in **b.**, which distributes tasks more between the branches.

**c.** shows MSEs for intact networks, one branch, and both branches lesioned over five seeds at the end of training, with standard deviations in error bars. More epochs between task switching decreases specialization, but further decreasing alternation frequency does not further reduce specialization in these experiments.

These results suggest some neuroscientific hypotheses about how and when branch specialization occurs. One can predict that brain regions that are consistently allocated to the same tasks across individuals have architectural inductive biases for those tasks, as mentioned by previous studies such as<sup>47</sup>. Another prediction is that the difference between multiple-demand and specialized brain regions may be dependent on the statistics of task presentation throughout an animal’s life—that is, perhaps more functions in multiple-demand regions are blocked and infrequent compared to a task like visual perception, which the highly specialized human visual cortex must perform almost every waking hour.

For future work, we would like to better understand both the mechanisms leading to branch specialization and the consequences of it. In terms of mechanisms, training dynamics would be interesting to study, since based on how reliably we see branch specialization in small models and large vision models<sup>199</sup> alike, we expect the loss surfaces of branches within one architecture to interact with each other in a feedback loop that pushes branch functions away from each other during training. With regards to the consequences of branch specialization, we want to know when specialization is or is not useful. If both very specific and very broadly-used regions exist in the brain, does the distribution of neural substrate that these tasks are computed on affect performance? In the future we would like to understand this task localization trade-off, and its broader implications to generalization in human and machine visual intelligence.

## 3.5 METHODS

### 3.5.1 CREATION OF GABOR DATASET

Gabor filters are defined by the following sets of equations.  $\theta$  is the angle parameter mentioned in Figures 3.3, 3.4, and 3.5.  $\omega$  is the size parameter in Figures 3.4 and 3.5.

$$g(x', y', \omega) = \frac{\omega^2}{4\pi^3} \exp\left(\frac{-\omega^2}{8\pi^2} * (4x'^2 + y'^2)\right) \exp(\pi^2/2) * \cos(\omega x')$$

$$x' = x \cos \theta + y \sin \theta$$

$$y' = -x \sin \theta + y \cos \theta$$

We then added a random number to the image from a uniform distribution  $\in [-1, 1]$  to vary the image colors. For all experiments,  $\theta \in [0, \pi]$  and  $\omega \in [1, 3]$ , both drawn randomly from uniform distributions. Image sizes were  $32 \times 32$ . The training set was 20k images and the test set was 10k images.

### 3.5.2 TRAINING NETWORKS

All code is available online at <https://github.com/ccli3896/branches-svrhm>.

#### HARDWARE

All experiments use neural networks written in Pytorch<sup>157</sup>. Experiments were run on shared GPUs in Google Colab (with GPU models allocated from NVIDIA K80, T4, P4, and P100) and the FASRC cluster, supported by the FAS Division of Science Research Computing Group at Harvard University (containing automatically allocated GPU models from among NVIDIA K20m, K40m, K80, M40, 1080, TITAN X, TITAN V, P100, V100, and RTX2080TI).

#### NETWORK AND TRAINING PARAMETERS

**NETWORK SIZES** All branched networks had the following traits: they took  $32 \times 32$  image inputs fed immediately into two branches. The two branches' outputs were concatenated and fed into a

linear readout layer, which always output two values.

Convolutional branches had two convolutional layers, max pooling, and two fully connected layers. The first convolutional layer took in one channel and output 32 channels with a kernel size of 3 and padding of 1. The second layer was the same except that the input was 32 channels as well. The max pooling layer had a kernel size of  $2 \times 2$ . It was followed by two dense layers. The first took an input size of 2048 (the flattened output of the max pool layer). Its output size was 120. The second dense layer took an input size of 120 and output of 84. All layers used relu as an activation function. In total the convolutional branch had 265612 trainable parameters.

The fully connected branch used in Figure 3.3 was a two-layer fully connected network with an input size of  $32 \times 32$  (the flattened image). The first layer had an output size of 86; the second layer took an input of size 86 and output 10. Both layers used relu for an activation function. In total there were 89020 trainable parameters.

Thus, the networks with fully connected and convolutional branches had  $265612 \text{ conv branch} + 89020 \text{ dense branch} + (84 \text{ conv outputs} + 10 \text{ dense outputs}) * 2 \text{ outputs} + 2 \text{ biases for outputs} = 354822$  parameters. Networks with two convolutional branches had  $265612 * 2 + (84 * 2) * 2 + 2 = 531562$  parameters.

**TRAINING** For training all models, we used Adam optimizer with a learning rate of .001. For experiments in Figure 3.5, we used dropout with a rate of .5 and elastic weight consolidation with an importance (see <sup>108</sup>)  $1e4$  for the size task and  $1e7$  for the angle task, both chosen after a period of tuning.

### 3.6 CONCLUSION AND NEXT STEPS

The lesson I took away from this project was not the specifics of the branched architecture setup, but rather the more general notion of autonomy in artificial neural networks. I developed a sense

that control was not necessarily the best way to build a useful algorithm, and that sometimes you could just let the network figure things out. With that in mind, I went back to pursuing my goal of a model of a small nervous system based on emergence. The results of that pursuit fill out the rest of this thesis.







*To try to make some meaning out of all this seems unbelievably quaint. Maybe I only see a pattern because I've been staring too long. But then again, to paraphrase Boris, maybe I see a pattern because it's there.*

Donna Tartt, *The Goldfinch*

# 4

## Building up a very small brain: Optimization

THE NEXT TWO CHAPTERS ARE MY ATTEMPT AT BUILDING AN EMERGENT BRAIN. A very, very small emergent brain, and an incomplete one (Appendix A.1), but it is a first step. I present

here the first biologically plausible algorithm I've seen that can explain animal autonomy, reward-seeking, and self-governed task-switching at the same time. These results are available as a preprint on bioRxiv.<sup>125</sup>

In this chapter, I show that neural networks can implement reward-seeking behavior using only local predictive updates and internal noise. These networks are capable of autonomous interaction with an environment and can switch between explore and exploit behavior, which we show is governed by attractor dynamics. This is a very important outcome, because attractor models represent a “singular success” in modelling and understanding many key features of the brain, such as memory, error correction, and signal integration (Khona and Fiete, 2022<sup>105</sup>). The fact that our model uses attractors to carry out behavioral computations makes it very compelling as a true model of a working nervous system.

In the next chapter, I show that networks can adapt to changes in their architectures, environments, or motor interfaces without any external control signals. When networks have a choice between different tasks, they can form preferences that depend on patterns of noise and initialization, and I show that these preferences can be biased by network architectures or by changing learning rates. The algorithm presents a flexible, biologically plausible way of interacting with environments without requiring an explicit environmental reward function, allowing for behavior that is both highly adaptable and autonomous. Code is available at <https://github.com/cc1i3896/PaN>.

#### 4.1 INTRODUCTION

There is a growing consensus that nervous systems are emergent<sup>193,159</sup>, meaning that behaviors and computations emerge from the interactions of many small components (neurons) rather than the independent actions of larger modules. Neuron-level prediction is a biologically plausible theory of how neural networks form useful sensory representations<sup>146,61</sup>, and together with the emergent view of the brain, it is likely that neurons update their activities and connection strengths with a

small and consistent set of rules that include a predictive component<sup>57,74,89?</sup>.

However, predictive agents are susceptible to the “dark room problem,” where agents minimize predictive errors by either reducing their activity to zero or staying in places where nothing happens<sup>186</sup>. Predictive agents that explore and act in environments usually need additional components to work, such as separate action selection modules<sup>57,161</sup> or curiosity drives<sup>70</sup>.

At the same time, biophysical studies have long established the presence of noise in nervous systems. Sources of noise include ion channel fluctuations, synaptic failure, or inconsistent vesicular release at synapses<sup>45,56,111</sup>. The role of noise in the brain is still not agreed upon, especially considering that some forms of it—such as spontaneous neural activity—can be energetically expensive to maintain<sup>123</sup>. Common perspectives are that nervous systems must somehow compensate for noise, average over it, or use noise as a means of regularization<sup>56,127</sup>.

Here we propose a new role for internal noise. We show that noise itself can overcome the dark room problem while adhering to the predictive and emergent view of behavior. We combine noise with predictive coding models<sup>140,170,151,183</sup> to form our algorithm PaN (Prediction and Noise), and find that it can interact with environments in interesting ways. Our main contributions are to show that:

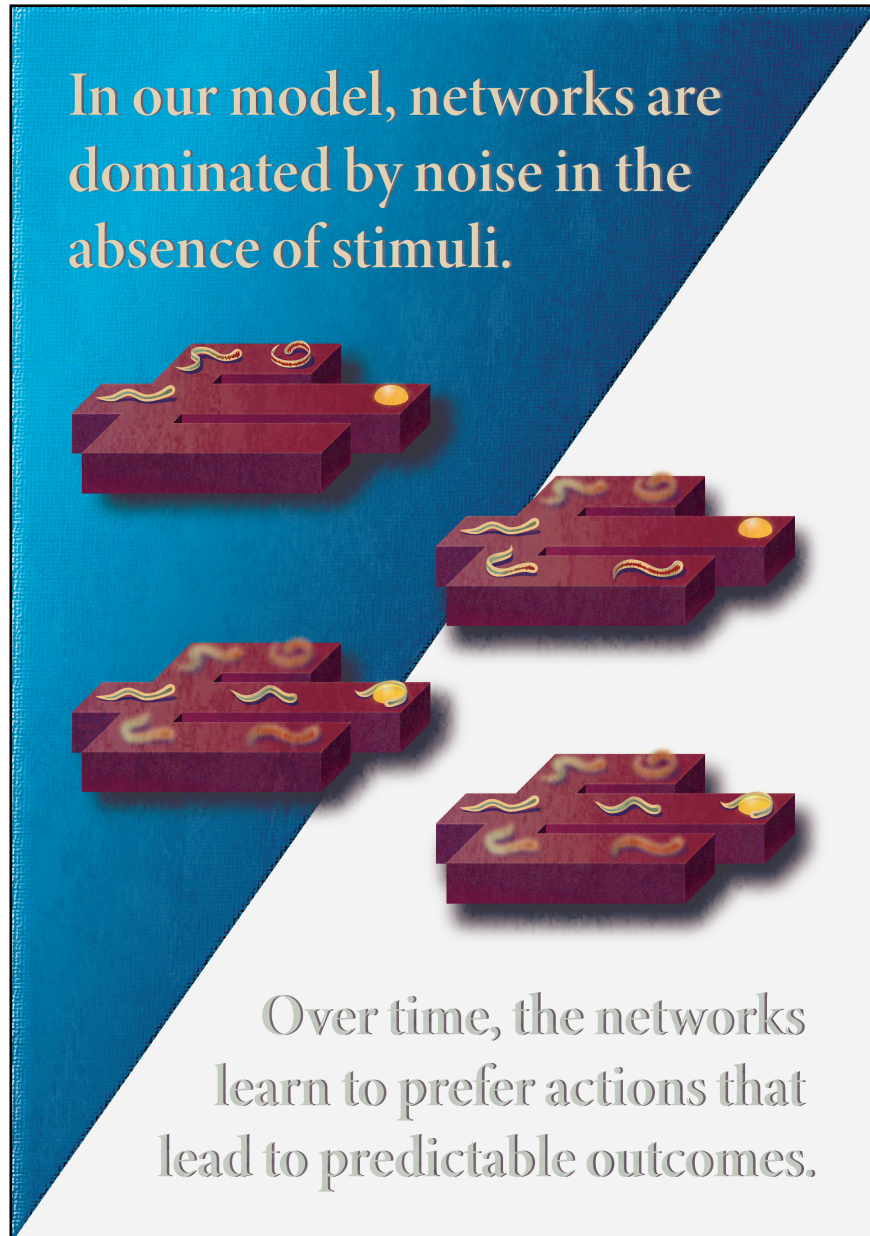
- Local prediction and noise alone can implement reward-seeking behavior.
- Networks switch between exploration and exploitation phases without prompting. These phases can be explained by attractor dynamics.
- Networks adapt to both internal and external changes. Internal change refers to the addition or removal of neurons; external change refers to environments with dynamic reward structures or motor interfaces.
- Networks vary in their goal preferences, which can be biased in biologically relevant ways: initialization, experience, architecture, and modulated learning rates.

PaN behavior is emergent rather than governed by optimization of an environmental reward. It therefore differs from the classical reinforcement learning framework, and so we place PaN under the novel framework of *self-supervised behavior* (SSB), motivated and defined in Section 5.4.

Figure 4.1:

**Excerpt from Cosyne 2024 poster.** I illustrate how in the prediction and noise model, possible actions have some probability of occurring, but only actions that lead to signal persist. Images progress temporally from top to bottom, and each path in a fork represents one of three possible action sequences. In the top two images, when there are no sensory stimuli at the end of the action sequence, noise dominates the network and prediction error is high. These are undesirable states, represented by red in the cartoons, and are not reinforced, represented by the blurring. When networks do receive stimuli, such as the gold droplet, there is a predictable signal and the actions that led to that signal are good (represented by green) and reinforced (represented by the lack of blurring). The fourth and final image shows that the presence of noise ensures that networks maintain exploration over the action space, preventing optimality but allowing for flexibility of behavior.

Figure 4.1: (continued)



## 4.2 RELATED WORK

### 4.2.1 PREDICTION AS ACTION

Earlier work in active inference<sup>80,64</sup> has suggested that actions can be seen as a way to minimize predictive error, illustrated in Figure 4.2. Parr and Pezzulo<sup>156</sup> state that “active inference is a normative framework to characterize Bayes-optimal behavior and cognition in living organisms,” suggesting that “living organisms follow a unique imperative: minimizing the surprise of their sensory observations.”

However, most active inference models require components with separate roles<sup>139,156,64,60</sup> and thus do not operate on emergent principles. Active inference models are computationally costly and critics have remarked that they can be difficult to build or understand<sup>32,63,21</sup>. Although these models have been successful in certain environments<sup>92,139,196</sup>, one must still specify learned target distributions, and scaling them up is an ongoing problem<sup>195</sup>.

We use the idea from active inference that actions are a way to minimize predictive errors. But we do not aim to explain or generate Bayes-optimal behavior—nor do we aim to produce behavior that is optimal with respect to any predefined utility functions. Moreover, the continuous injection of internal noise in our model both *prevents* optimality in static environments and *is necessary* for most of our listed contributions (Section 4.4). Authors of active inference and the related free energy principle literature, on the other hand, assume that one can average over internal noise to arrive at long-term behavior<sup>63</sup>.

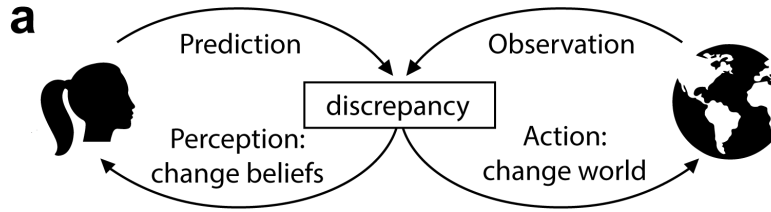


Figure 4.2: a. Active inference, adapted from Figure 2.3 of Parr and Pezzulo<sup>156</sup>. When an organism makes predictions about its environment, it may experience discrepancies with its observations. To reduce discrepancies, the organism can either change its beliefs about the environment or change the environment to match its predictions.

#### 4.2.2 RELEVANT MODELS

We build on the existing body of work in predictive coding (PC), especially Monte Carlo Predictive Coding (MCPC)<sup>151</sup> and Incremental Predictive Coding (iPC)<sup>170</sup>, both reviewed briefly here. While previous work has focused on PC’s role in perception or the learning of fixed targets, we demonstrate novel capabilities of these algorithms in actively behaving contexts.

**BASE PREDICTIVE CODING ALGORITHM** In PC as defined by Bogacz and colleagues<sup>140,183</sup>, networks minimize a predictive energy (Equation 1)<sup>140</sup> through activity and weight updates. To learn an input-output mapping, a PC network’s first and last layer’s activities are fixed to target values; then, network activities are updated to minimize prediction error. After activity updates are run to convergence, weights are updated for a single step to minimize the same prediction error. All updates use local information, and activity and weight updates alternate until the network has converged. Song et al. (2024)<sup>183</sup> show that this procedure may enable more efficient and biologically plausible learning than backpropagation, where loss-minimizing updates are restricted to weights.

**MONTE CARLO PREDICTIVE CODING** MCPC adds noise to the activity updates from PC<sup>151</sup>. Given images from the MNIST dataset, MCPC can infer and sample posterior distributions of la-



tent variables. The algorithm is robust to a range of noise settings and can account for experimental findings; for example, it explains a reduction in neural variability after stimulus presentation, which has been observed across a variety of conditions in animals<sup>36</sup>.

**ACTIVITY AND WEIGHT COUPLING** We also couple our activity and weight updates as in Incremental predictive coding (iPC)<sup>170</sup>. iPC modifies PC by updating activities and weights in alternation, so activities are no longer updated to convergence. Coupling eliminates the need in PC for an external control signal to switch the network from activity updates to the weight update step. iPC is faster than PC and has convergence guarantees, whereas convergence in PC can be unpredictable<sup>170</sup>.

Activity and weight coupling has been studied in other contexts, such as recurrent neural network (RNN) training. While our models do not have recurrent weights, an effective recurrence can be introduced with environmental feedback. RNNs are normally trained to produce desired activity dynamics with frozen weights, but alternative training schemes using coupled activity and weight updates can impart computational benefits<sup>138,137,84,10</sup>—in one example, Clark and Abbott (2024)<sup>37</sup> describe how such coupling can serve as a mechanism for working memory.

### 4.3 METHODS

Figure 4.3a shows an example PaN network with an argmax-based environment function and fixed action-to-signal mappings. For a network of  $L$  layers, let the neuron activities for layer  $l$  be denoted  $\mathbf{x}_l$  and the weights connecting layer  $l$  to  $l + 1$  be  $\mathbf{W}_l$ . Let  $\mathbf{s}$  be a scalar or vector of sensory feedback. We define a predictive energy for each timestep  $t$ :

---

**Algorithm 1** Prediction and Noise (PaN).
 

---

**Require:**  $L$  layers, activities  $\mathbf{x}_0$  to  $\mathbf{x}_{L-1}$ , sensory input  $\mathbf{s}$ , weights  $\mathbf{W}_0$  to  $\mathbf{W}_{L-2}$ .  
 Activity and weight learning rates  $\alpha$  and  $\omega$ , respectively. Default  $\alpha = \omega = 0.01$ .  
 Activity settling steps  $J$ ; standard deviation of noise  $\eta_x$  and  $\eta_W$ .  
 $\text{Env}(\text{argmax}(\mathbf{x}_{L-1}))$ , an environment interaction function.  
 Predictive energy  $E(\mathbf{s}, \mathbf{x}_0, \dots, \mathbf{x}_{L-1}, \mathbf{W}_0, \dots, \mathbf{W}_{L-2})$   
 (Equation 4.1 unless specified)

**for**  $t = 1, 2, \dots$ , Simulation length, **do**  
 $\mathbf{s} \leftarrow \text{Env}(\text{argmax}(\mathbf{x}_{L-1}))$   
**for**  $j = 1, 2, \dots, J$ , **do**  
 $\mathbf{x}_l \leftarrow \mathbf{x}_l - \alpha \frac{\partial E}{\partial \mathbf{x}_l}$ ,  $0 \leq l \leq L-1$   
 $\mathbf{x}_l \leftarrow \mathbf{x}_l + \mathbf{n}_x$ ,  $\mathbf{n}_x \sim \mathcal{N}(0, \eta_x)$ ,  $0 \leq l \leq L-1$   
**end for**  
 $\mathbf{W}_l \leftarrow \mathbf{W}_l - \omega \frac{\partial E}{\partial \mathbf{W}_l}$ ,  $0 \leq l \leq L-2$   
 $\mathbf{W}_l \leftarrow \mathbf{W}_l + \mathbf{n}_W$ ,  $\mathbf{n}_W \sim \mathcal{N}(0, \eta_W)$ ,  $0 \leq l \leq L-2$   
 $\mathbf{W}_l \leftarrow \text{clip}(\mathbf{W}_l, -2, 2)$ ,  $0 \leq l \leq L-2$   
**end for**

---

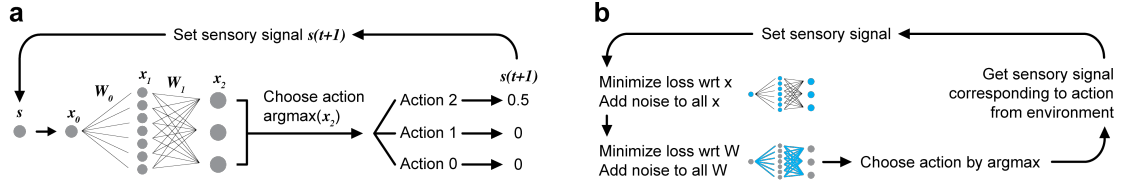


Figure 4.3: **a.** Flow of information in a Prediction and Noise (PaN) network, using an example set of possible action-to-signal mappings.  $\mathbf{s}$  is an input,  $\mathbf{x}_l$  denotes the activity vector for each layer  $l$ , and  $\mathbf{W}_l$  denotes the weight matrix connecting layer  $l$  to  $l + 1$ . **b.** Update loop (see Algorithm 1).

$$E(t) = \frac{1}{2}(\mathbf{s}(t) - \mathbf{x}_0(t))^2 + \frac{1}{2} \sum_{l=1}^{L-1} (\mathbf{x}_l(t) - \text{Relu}(\mathbf{W}_{l-1}(t)\mathbf{x}_{l-1}(t)))^2 \quad (4.1)$$

Relu is the rectified linear activation function. The sensory feedback  $\mathbf{s}(t)$  is determined by some environment function  $\text{Env}(\mathbf{x}_{L-1}(t-1)) = \mathbf{s}(t)$ ; we specify this environment function in each experiment. Networks are updated using Algorithm 1, illustrated in Figure 4.3b. Activities and weights are updated in alternation to minimize the predictive energy (Equation 1). Every update is accompanied by the addition of white noise, independent at every timestep and normally distributed with a standard deviation of  $\eta_x$  for activities and  $\eta_W$  for weights. All experiments in the paper were run on Intel Cascade Lake or Sapphire Lake CPUs.

## 4.4 RESULTS

### 4.4.1 TWO-NEURON NETWORKS SEEK REWARD AND SWITCH BETWEEN EXPLORATION AND EXPLOITATION

We start with two neurons connected by a weight, Figure 4.4a, with no nonlinearity in the energy for ease of analysis:

$$E(t) = \frac{1}{2}(s(t) - x_0(t))^2 + \frac{1}{2}(x_1(t) - W_0(t)x_0(t))^2 \quad (4.2)$$

PaN networks are run for 500k timesteps per trial in the closed-loop setup in Figure 4.4a. Without noise,  $\eta_x = \eta_W = 0$ , networks randomly fixate on one action (Figure 4.4b). With noise, networks learn to prefer Action 1, which is associated with nonzero signal and in this context looks like reward-seeking behavior (Figure 4.4c-d). Looking more closely at an example noise setting

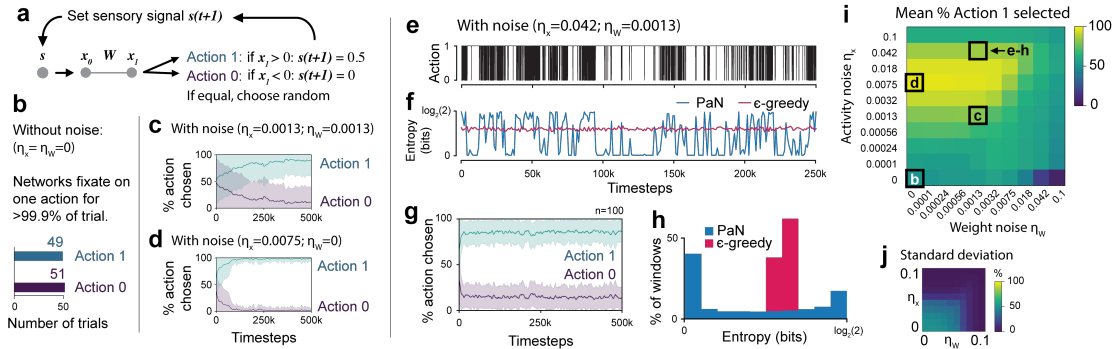


Figure 4.4: **a.** A two-neuron network that can take two actions. Updates follow Algorithm 1 without Relu nonlinearity for ease of analysis;  $J=1$ . **b.** Without noise, networks randomly fixate on one action for  $>99.9\%$  of trial. **c-d.** With noise, networks choose the rewarding action at a rate dependent on activity and weight noise values. Mean plotted over 100 seeds. Shaded regions denote standard deviation. **e.** Sample actions chosen over 250k timesteps for  $\eta_x = 0.042, \eta_W = 0.0013$  where  $\eta$  values define the width of noise distributions. **f.** Entropy of rolling 1000-timestep windows for PaN as well as an  $\epsilon$ -greedy algorithm with  $\epsilon$  set to 0.3 for matched reward 85%. **g.** Same as (c-d) but for  $\eta_x = 0.042, \eta_W = 0.0013$ . **h.** Histogram of entropy values over 100 seeds for 500k timesteps each. In this noise setting, PaN is bimodal with peaks corresponding to exploration (entropy close to  $\log_2(2) = 1$ ) and exploitation (entropy close to 0). An  $\epsilon$ -greedy agent, in contrast, maintains a consistently random exploration strategy. Section 4.5 shows that bimodality is not strongly dependent on window size. **i.** Different noise scales were tested for 100 seeds, 500k timesteps each. The mean percentage of time networks selected Action 1, the rewarding action, with standard deviations in **j**. See Section 4.6 Table 1 for values. Upper bound for compute for this figure and Figure 4.5 was 500 CPU hours, 55 GB for storage.

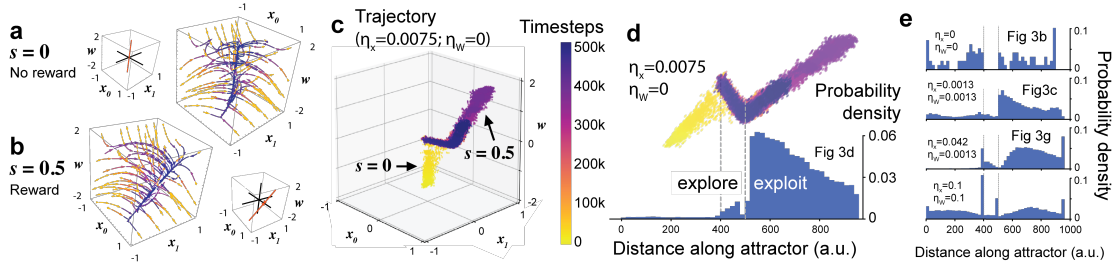


Figure 4.5: **a-b.** Attractors derived in Section 4.7 for the two-neuron system in Figure 4.4a. Small 3D plots are the line attractors for each case and the larger colored plots show how networks move toward attractors, from yellow to purple over time. For  $s = 0$ , the attractor is along the  $W$ -axis (a). For  $s = 0.5$ , the attractor is along  $x_1 = Wx_0$  (b). **c.** A single trajectory in parameter space  $(x_0, x_1, W)$  with movement along and between attractors. **d.** Probability density of 100 trajectories along attractor branches in arbitrary units (a.u.), with reference locations on the attractor marked in gray dashed lines. In this noise setting, networks strongly prefer the exploit branch. We only include timesteps 200k-500k to account for early transients in (d-e). **e.** Probability densities of other noise settings, 100 trials each. Without noise (top), networks stay along the part of the attractor where they began. With high noise (bottom), where networks distribute themselves widely and do not seek reward. Gray dashed lines correspond to the same locations as in (d).

$\eta_x = 0.042, \eta_W = 0.0013$  in Figure 4.4e-h, we see that action choices (Figure 4.4e) alternate between high entropy exploratory states and low entropy exploitative states. In contrast, trained  $\epsilon$ -greedy agents have uniform patterns of exploration (Figure 4.4f,h). Section 4.5 shows that the separation of high and low entropy states is not strongly dependent on the window size used. Activity and weight noise scales impact the amount of reward collected (Figure 4.4i-j) with a maximum at  $\eta_x = 0.0075, \eta_W = 0$  (see Section 4.6 Table 1) and different qualitative behaviors for different noise settings. See Section 4.6 for further examples.

#### ATTRACTOR DYNAMICS EXPLAIN TWO-NEURON BEHAVIOR

For intuition as to why PaN selects the rewarding action, consider the case where Action o is chosen. Then  $s(t+1) = 0$ , and using labels from Figure 4.4a,  $x_0$  and  $x_1$  will both quickly converge to o. When  $x_1$  is close to o, the next action is mostly determined by noise, leading to high prediction error.

But when Action 1 is chosen,  $s(t + 1) = 0.5$ , and  $x_1$  will instead converge to  $0.5W$ . As long as  $W$  is not close to 0, the network is likely to fixate on a single action, leading to low prediction error.

We can explain the explore/exploit behavior of the PaN network by studying the underlying dynamical system. When  $s$  in Equation 2 is fixed to either 0 or 0.5, we can derive an associated attractor in the parameter space,  $(x_0, x_1, W)$ . Figure 4.5a shows the line attractor associated with no reward and Figure 4.5b the attractor with reward. Small plots in Figure 4.5a-b are the lines that networks move toward, and colored plots show how networks move, from yellow to purple.

A parameter space trajectory in Figure 4.5c for the noise setting in Figure 4.4d shows that the network moves along and between the attractors. Attractor stabilities are determined by analyzing the Jacobian of the stochastic differential equation approximation of the network, which we explain and empirically validate in Section 4.7. The three segments of the trajectory correspond to  $s = 0$  (Figure 4.5a), which is unstable in the presence of noise;  $s = 0.5$  (Figure 4.5b), which is stable exploitation behavior; and movement between these two states, which is exploration behavior also driven by noise. We refer to the last segment as the attractor “bridge” for convenience. Video 1 shows the correspondence of trajectory location and explore/exploit behavior for the two-neuron system.

Figure 4.5d shows the probability density of 100 trajectories like the one in Figure 4.5c, plotting the densities along the closest point on the attractors and their bridge. In Figure 4.5d, the network is inclined toward exploitation. Changing probability densities in Figure 4.5e show that noise affects the tendency of a network to explore or exploit. Without noise, networks stay where they began and do not prefer one action over another (Figure 4.5e, top). At intermediate levels of noise (Figure 4.5d, e second and third plots), networks occupy the attractor corresponding to exploitation with higher probability. With very high noise (Figure 4.5e, bottom) networks distribute themselves between both attractors and do not show consistent reward-seeking. See Section 4.7 for details.

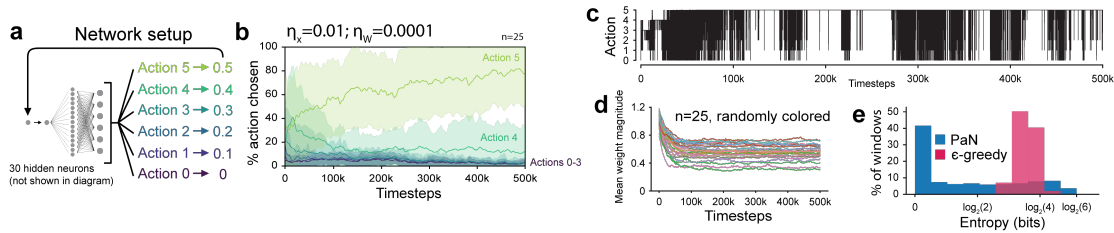


Figure 4.6: **PaN networks learn to maximize sensory signals.** **a.** PaN networks with linearly spaced action choices **b.** learn to optimize signal, not just choose nonzero values. Mean and standard deviation plotted over 25 seeds. **c.** Example trace for (a-b). **d.** Signal optimization can be explained by weight decay, automatically implemented by PaN loss (see text for discussion). Each line is the mean weight magnitude  $\overline{W}_0(t)$  over the course of the trial. 25 random seeds. **e.** Entropy distributions for PaN networks in (a-d), 25 trials, with distributions for 25 trials of matched-reward  $\epsilon$ -greedy agent activity. Experiments required upper bound of 6.25 CPU hours, 10 GB.

#### 4.4.2 LARGER PaN NETWORKS LEARN TO CHOSE MAXIMALLY REWARDING ACTIONS

Previous results show that in the presence of noise, a two-neuron system chooses a rewarding action more often than a zero-reward action. We next test a larger network with a 30-neuron hidden layer that can choose between six actions with linearly spaced rewards (Figure 4.6a). Networks learn to choose the maximal action over time (Figure 4.6b, with c as an example trace). As in Figure 4.4h, Figure 4.6e shows greater variation in entropy distributions as compared to a matched-reward  $\epsilon$ -greedy agent.

To provide intuition for why PaN fixates on actions that maximize signals rather than any nonzero action, note that the predictive energy increases through both fluctuating inputs and internal noise. The contribution through input fluctuation can be eliminated by making a set of weights large enough to fixate on one action, and the contribution through internal noise can be minimized by shrinking all weights. Because larger inputs allow for smaller weights to achieve fixation, networks prefer actions that lead to the largest input signals. Consistent with this intuition, we observe that weight decay is a consequence of minimizing the PaN loss in the presence of noise (Figure 4.6d).

#### 4.5 EXPLORE/EXPLOIT MODES PERSIST IN ENTROPY CALCULATIONS WITH DIFFERENT WINDOW SIZES.

In Figure 4.7 we show that distinct explore/exploit states persist regardless of window sizes for entropy calculations. The data here are for the system in Figure 4.8. First we plot comparisons for PaN (Figure 4.7a) and an  $\epsilon$ -greedy agent with matched reward (Figure 4.7b). Figures 4.7c-d show PaN at different timescales, while Figure 4.7e plots the entropy for a 1000-timestep rolling window for both PaN and  $\epsilon$ -greedy agents. PaN exhibits distinct high-entropy (explore) and low-entropy (exploit) phases, while  $\epsilon$ -greedy agents maintain consistent levels of exploration throughout. Figure 4.7f shows that even when entropy windows are very small, PaN still has a bimodal entropy distribution, while the  $\epsilon$ -greedy agent distributions are unimodal. Figure 4.7g plots exploitation, as defined by the frequency of windows with very low entropy (marked bars in Figure 4.7f) for a range of noise conditions.



## 4.6 THE EFFECT OF NOISE PARAMETERS ON BEHAVIOR.

The scales of activity and weight noise affects how frequently a PaN network chooses the rewarding action in a bandit task. They also affect the amount of exploration, and variability in performance between networks. Here we show, using a network with a 30-neuron hidden layer and 3 possible actions, that different noise settings can produce different qualitative regimes of behavior (Figure 4.8).

In Figure 4.8a we show the network setup and in Figure 4.8b, we plot the percentage of time the rewarding action was selected. We see that there is a range of noise settings that leads to higher reward, around  $\eta_x \in [0.0013, 0.042]$ . Without noise, networks fixate on a random action in Figure 4.8c. With noise, networks choose the rewarding action and learn to exploit it more over time (Figure 4.8d).

An interesting case arises, however, when weight noise is high and activity noise is low. We see in Figure 4.8e that here, the rewarding action is avoided rather than preferred. This is because noise is only propagated through the system when there is signal, and so the network prefers zero reward actions over ones with any signal.

Finally, when weight and activity noise are both high, all actions are random (Figure 4.8f).

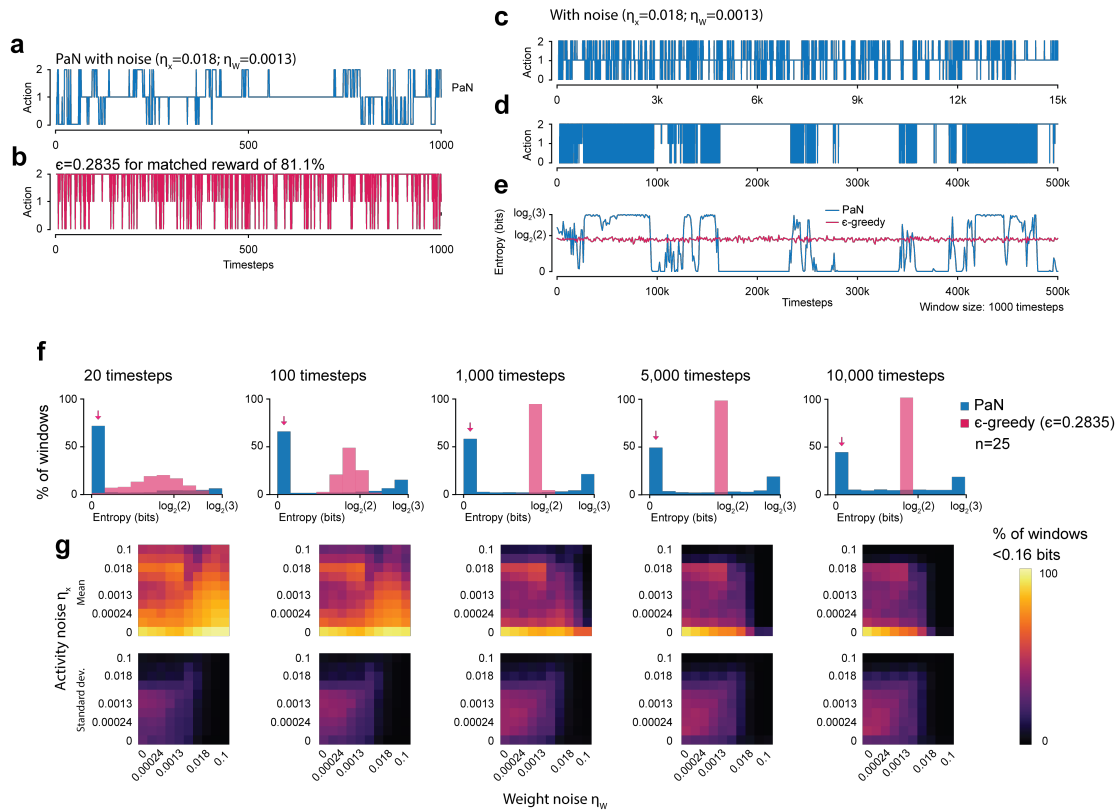
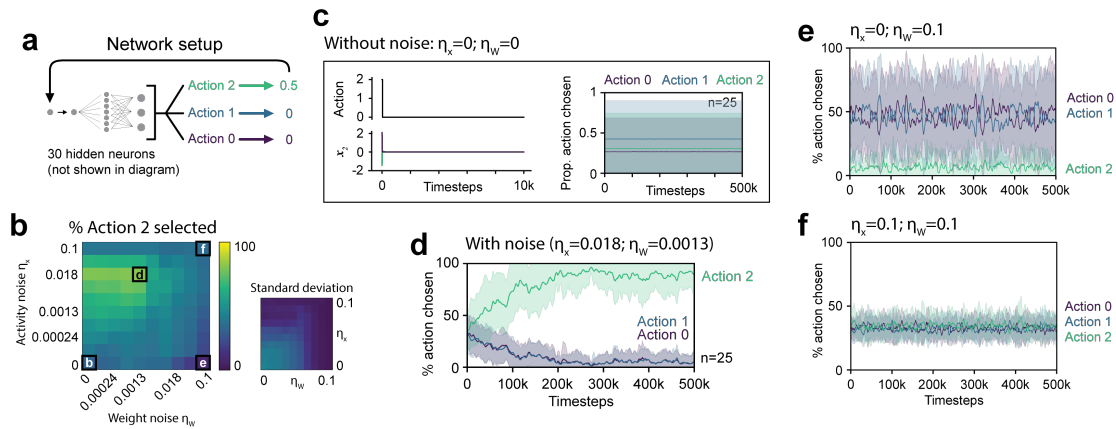


Figure 4.7: **a.** Sample action choices for a PaN network in Figure 4.8a, first 1000 steps. Compare to **b.**, which is an  $\epsilon$ -greedy agent with a matched proportion of reward selected. **c.** The first 15k steps from the same trial as **a.** PaN displays exploration (dense parts of plot) and exploitation (fixation) at multiple timescales. **d.** Entropy of rolling windows of action selections for both PaN and  $\epsilon$ -greedy agents. PaN transitions between high entropy (exploration) and low entropy (exploitation). **e.** Histograms of entropies for PaN and  $\epsilon$ -greedy agents with differently sized entropy windows, showing that bimodality in PaN's entropy does not depend strongly on window size. Here bimodality suggests the presence of two behavioral states, as discussed for Figure 4.4h in main text. **f.** We devise a proxy to visualize the amount of exploitation in a PaN network, which is the proportion of windows across trials and 25 random seeds with  $< 0.16$  bits ( $0.1 * \log_2(3)$ ) of information. Pink arrows mark this value in **e.** for a noise setting of  $\eta_x = 0.018; \eta_w = 0.0013$ . We plot the exploitation metric for all noise settings and a range of window sizes. For a broad range of window sizes (from 1k to 10k timesteps tested here) there remains a region of noise that allows for exploitation, around  $\eta_x = 0.018$ .



**Figure 4.8: PaN behavior in a 3-armed bandit task under different noise settings.** **a.** Network setup with one hidden layer and three possible actions. Actions correspond to  $[0, 0, 0.5]$  reward, respectively. **b.** The percentage of actions chosen that were rewarding (Action 2). Mean (left) and standard deviation (right);  $n=25$  per condition. For exact values see Table 2. **c.** When noise parameters are 0, networks fixate on one randomly chosen action. This and following plots are mean action selections over time, with shaded areas denoting standard deviation,  $n=25$ . **d.** When noise is nonzero and within a certain range (see b.), networks learn to choose the most rewarding action. **e.** When weight noise is very high and activity noise is 0, rewarding actions (Action 2) are *not* preferred, and the network chooses the zero-reward actions preferentially. **f.** In high noise settings, networks are dominated by noisy actions throughout trials.

#### 4.6.1 TABLE VALUES FOR NOISE PARAMETER SWEEPS

We include here the values for heatmaps in Figure 4.4i-j and Figure 4.8b.

Table 4.1: **Values for Figure 4.4i-j.a.** Mean and **b.** standard deviation of proportion of actions taken that were rewarding for an action space with 2 actions corresponding to rewards [0, 0.5]. For two-neuron system in Figure 4.4a. Note that here, random actions would correspond to a mean of 0.5. The maximum means in (a) are bolded (.957).

| $\eta_x \backslash \eta_W$ | 0    | .0001 | .00024 | .00056 | .0013 | .0032 | .0075 | .018 | .042 | .1   |
|----------------------------|------|-------|--------|--------|-------|-------|-------|------|------|------|
| 0                          | .510 | .514  | .529   | .530   | .538  | .572  | .540  | .428 | .186 | .059 |
| .0001                      | .556 | .554  | .552   | .552   | .576  | .623  | .615  | .582 | .555 | .457 |
| .00024                     | .608 | .610  | .604   | .596   | .600  | .632  | .620  | .585 | .560 | .478 |
| .00056                     | .718 | .721  | .723   | .716   | .669  | .657  | .633  | .590 | .565 | .496 |
| .0013                      | .807 | .809  | .811   | .816   | .809  | .725  | .663  | .601 | .573 | .512 |
| .0032                      | .915 | .915  | .915   | .915   | .911  | .877  | .733  | .628 | .588 | .530 |
| .0075                      | .957 | .957  | .957   | .957   | .955  | .943  | .851  | .684 | .614 | .552 |
| .018                       | .947 | .947  | .947   | .947   | .947  | .942  | .914  | .767 | .651 | .579 |
| .042                       | .851 | .851  | .851   | .851   | .851  | .849  | .841  | .778 | .655 | .571 |
| .1                         | .615 | .615  | .615   | .615   | .615  | .615  | .613  | .605 | .576 | .537 |

| $\eta_x \backslash \eta_W$ | 0    | .0001 | .00024 | .00056 | .0013 | .0032 | .0075 | .018 | .042 | .1   |
|----------------------------|------|-------|--------|--------|-------|-------|-------|------|------|------|
| 0                          | .500 | .495  | .488   | .475   | .434  | .379  | .217  | .099 | .036 | .013 |
| .0001                      | .483 | .483  | .471   | .449   | .410  | .349  | .205  | .090 | .043 | .017 |
| .00024                     | .455 | .453  | .447   | .429   | .398  | .345  | .204  | .090 | .043 | .017 |
| .00056                     | .384 | .380  | .374   | .365   | .360  | .332  | .200  | .089 | .043 | .017 |
| .0013                      | .329 | .326  | .321   | .306   | .275  | .281  | .191  | .088 | .042 | .017 |
| .0032                      | .146 | .146  | .146   | .146   | .149  | .142  | .163  | .084 | .041 | .017 |
| .0075                      | .048 | .048  | .048   | .048   | .051  | .055  | .086  | .075 | .038 | .016 |
| .018                       | .022 | .022  | .022   | .022   | .023  | .024  | .032  | .055 | .034 | .015 |
| .042                       | .018 | .018  | .018   | .018   | .018  | .019  | .020  | .029 | .026 | .013 |
| .1                         | .018 | .018  | .018   | .018   | .018  | .018  | .018  | .018 | .016 | .012 |



Table 4.4: Values for Figure 4.8b. **a.** Mean and **b.** standard deviation of proportion of actions taken that were rewarding for an action space with 3 actions corresponding to rewards  $[0.5, 0, 0]$ . For three-neuron system in Figure 4.8a. Note that here, random actions would correspond to a mean of 0.333. The maximum mean in (a) is bolded (.811).

Table 4.5:

| $\eta_x \backslash \eta_W$ | 0    | .0001 | .00024 | .00056 | .0013 | .0032 | .0075 | .018 | .042 | .1   |
|----------------------------|------|-------|--------|--------|-------|-------|-------|------|------|------|
| 0                          | .307 | .319  | .339   | .386   | .359  | .343  | .340  | .245 | .173 | .063 |
| .0001                      | .401 | .359  | .372   | .401   | .381  | .373  | .424  | .366 | .317 | .211 |
| .00024                     | .464 | .450  | .408   | .427   | .401  | .383  | .430  | .373 | .332 | .236 |
| .00056                     | .507 | .534  | .515   | .497   | .454  | .402  | .440  | .382 | .348 | .264 |
| .0013                      | .604 | .621  | .598   | .595   | .547  | .442  | .459  | .397 | .365 | .297 |
| .0032                      | .636 | .651  | .662   | .693   | .670  | .541  | .500  | .423 | .389 | .335 |
| .0075                      | .717 | .732  | .728   | .768   | .757  | .556  | .553  | .460 | .411 | .361 |
| .018                       | .800 | .796  | .789   | .802   | .811  | .587  | .516  | .482 | .416 | .359 |
| .042                       | .645 | .638  | .634   | .635   | .622  | .541  | .420  | .439 | .392 | .347 |
| .1                         | .387 | .386  | .382   | .390   | .388  | .388  | .352  | .352 | .364 | .346 |

Table 4.6:

| $\eta_x \backslash \eta_W$ | 0    | .0001 | .00024 | .00056 | .0013 | .0032 | .0075 | .018 | .042 | .1   |
|----------------------------|------|-------|--------|--------|-------|-------|-------|------|------|------|
| 0                          | .441 | .412  | .401   | .361   | .305  | .264  | .155  | .066 | .030 | .012 |
| .0001                      | .418 | .402  | .397   | .363   | .306  | .260  | .147  | .054 | .026 | .013 |
| .00024                     | .408 | .411  | .391   | .356   | .305  | .256  | .145  | .055 | .026 | .013 |
| .00056                     | .351 | .366  | .348   | .330   | .293  | .242  | .142  | .054 | .026 | .012 |
| .0013                      | .273 | .271  | .269   | .284   | .255  | .208  | .136  | .053 | .025 | .013 |
| .0032                      | .242 | .232  | .220   | .200   | .212  | .188  | .125  | .053 | .025 | .011 |
| .0075                      | .128 | .123  | .133   | .133   | .144  | .168  | .119  | .054 | .024 | .010 |
| .018                       | .049 | .065  | .063   | .071   | .072  | .145  | .114  | .050 | .020 | .010 |
| .042                       | .067 | .070  | .088   | .081   | .090  | .134  | .075  | .040 | .019 | .008 |
| .1                         | .047 | .051  | .045   | .044   | .048  | .040  | .036  | .035 | .024 | .010 |

## 4.7 ANALYSIS OF ATTRACTOR DYNAMICS

The goal of this section is to develop an intuition and explanation for PaN behavior based on a dynamical systems perspective of the analytically tractable two-neuron network. Jonah Brenner and I came up with the ideas in this section together, although he was primarily the one who wrote them up.

We begin by reinterpreting the prediction and noise (PaN) algorithm as a system of stochastic differential equations (SDEs). We use these equations to derive an attractor in the network’s parameter space in the absence of noise. The attractor has three components and we describe their associated stabilities and behaviors. Finally, we simulate the long-term distributions of network parameters along the attractor, and show how these distributions change with different noise magnitudes.

### 4.7.1 THE GENERAL PAN ALGORITHM AS A SET OF DIFFERENCE EQUATIONS

PaN’s dynamics are governed by the predictive energy function (Section 4.3):

$$E(t) = \frac{1}{2}(\mathbf{s}(t) - \mathbf{x}_0(t))^2 + \frac{1}{2} \sum_{l=1}^{L-1} (\mathbf{x}_l(t) - \text{Relu}(\mathbf{W}_{l-1}(t)\mathbf{x}_{l-1}(t)))^2 \quad (4.3)$$

At timestep  $t$ , a network’s activities  $\mathbf{x}_l(t)$  and weights  $\mathbf{W}_l(t)$  are updated according to the following steps, which we rewrite from Algorithm 1. Here,  $\alpha$  and  $\omega$  are the activity and weight learning rates,  $\sigma_x$  and  $\sigma_W$  are the activity and weight noise standard deviations, and  $\eta_{\mathbf{x}/\mathbf{W}}(t)$  are i.i.d. standard normal noise across parameters and time (notation here differs slightly from Algorithm 1 to facilitate analysis).

1. Begin with activities and weights  $(\mathbf{x}_l(t), \mathbf{W}_l(t))$  at time  $t$ .



2. Perform gradient descent on  $E$  with respect to the activities  $\mathbf{x}_l$

$$\mathbf{x}_l \leftarrow \mathbf{x}_l(t) - \alpha \nabla_{\mathbf{x}_l} E(\mathbf{x}(t), \mathbf{W}(t)) \quad (4.4)$$

3. Add activity noise

$$\mathbf{x}_l(t+1) \leftarrow \mathbf{x}_l + \sigma_x \boldsymbol{\eta}_x(t) \quad (4.5)$$

4. Perform gradient descent on  $E$  with respect to the weights  $\mathbf{W}_l$

$$\mathbf{W}_l \leftarrow \mathbf{W}_l(t) - \omega \nabla_{\mathbf{W}_l} E(\mathbf{x}(t+1), \mathbf{W}(t)) \quad (4.6)$$

5. Add weight noise

$$\mathbf{W}_l(t+1) \leftarrow \mathbf{W}_l + \sigma_W \boldsymbol{\eta}_W(t) \quad (4.7)$$

We can summarize the change in the activities and weights between timesteps  $t$  and  $t+1$ ,  $\Delta \mathbf{x}_l(t)$  and  $\Delta \mathbf{W}_l(t)$ , as

$$\Delta \mathbf{x}_l(t) = -\alpha \nabla_{\mathbf{x}_l} E(\mathbf{x}(t), \mathbf{W}(t)) + \sigma_x \boldsymbol{\eta}_{\mathbf{x}}(t) \quad (4.8)$$

$$\Delta \mathbf{W}_l(t) = -\omega \nabla_{\mathbf{W}_l} E(\mathbf{x}(t+1), \mathbf{W}(t)) + \sigma_W \boldsymbol{\eta}_{\mathbf{W}}(t) \quad (4.9)$$

Notice that  $\Delta \mathbf{W}_l(t)$  depends on  $\mathbf{x}(t+1)$  rather than  $\mathbf{x}(t)$ . So the weight update at timestep  $t$ , written in terms of the parameters at that timestep, is given by

$$\Delta \mathbf{W}_l(t) = -\omega \nabla_{\mathbf{W}_l} E[\mathbf{x}(t) - \alpha \nabla_{\mathbf{x}_l} E(\mathbf{x}(t), \mathbf{W}(t)) + \sigma_x \boldsymbol{\eta}_{\mathbf{x}}(t), \mathbf{W}(t)] + \sigma_W \boldsymbol{\eta}_{\mathbf{W}}(t) \quad (4.10)$$

We can summarize the PaN algorithm with the following set of noisy difference equations

$$\Delta \mathbf{x}_l(t) = -\alpha \nabla_{\mathbf{x}_l} E(\mathbf{x}(t), \mathbf{W}(t)) + \sigma_x \boldsymbol{\eta}_{\mathbf{x}}(t) \quad (4.11)$$

$$\Delta \mathbf{W}_l(t) = -\omega \nabla_{\mathbf{W}_l} E[\mathbf{x}(t) - \alpha \nabla_{\mathbf{x}_l} E(\mathbf{x}(t), \mathbf{W}(t)) + \sigma_x \boldsymbol{\eta}_{\mathbf{x}}(t), \mathbf{W}(t)] + \sigma_W \boldsymbol{\eta}_{\mathbf{W}}(t) \quad (4.12)$$

From now on, we drop the  $t$  argument, since it is the same everywhere.

#### 4.7.2 THE TWO-NEURON PAN NETWORK AS A NOISY DYNAMICAL SYSTEM

##### DIFFERENCE EQUATIONS FOR THE TWO-NEURON NETWORK

We now focus on the dynamics of the two-neuron agent in the two-action bandit task. The agent is parameterized by two activities,  $x_0$  and  $x_1$ , and a weight  $W$ . It receives one sensory input  $s$  from the

environment. Its dynamics are governed by the energy function from Equation 2:

$$E = \frac{1}{2}(s - x_0)^2 + \frac{1}{2}(x_1 - Wx_0)^2 \quad (4.13)$$

Here we removed the ReLU nonlinearity from Equation 1 for ease of analysis. Following the previous section, this energy gives the update rules

$$\Delta x_0 = -\alpha(x_0 - s + W^2x_0 - Wx_1) + \sigma_x\eta_{x_0} \quad (4.14)$$

$$\Delta x_1 = -\alpha(x_1 - Wx_0) + \sigma_x\eta_{x_1} \quad (4.15)$$

$$\Delta W = -\omega \frac{\partial E [\mathbf{x} - \alpha \nabla_{\mathbf{x}} E(\mathbf{x}, W) + \sigma_x \boldsymbol{\eta}_{\mathbf{x}}, W]}{\partial W} + \sigma_W \eta_W \quad (4.16)$$

The activity noise now appears in the weight updates, meaning that activity noise and weight updates are correlated. Upon expansion of  $\Delta W$ , one finds that the variance of the noise in the weight update also depends on  $x_0, x_1$ , and  $W$ . Both of these observations will complicate future analysis.

Luckily, we can make a simplification: the complexity of the expression for  $\Delta W$  comes from the additional  $-\alpha \nabla_{\mathbf{x}} E(\mathbf{x}, W) + \sigma_x \boldsymbol{\eta}_{\mathbf{x}}$  term. This term appeared because we updated the weights according to  $\mathbf{x}(t+1)$  rather than  $\mathbf{x}(t)$ . However, terms from this addition must be of  $O(\omega\alpha)$ ,  $O(\omega\sigma_x)$ , or higher in  $\alpha$  or  $\sigma_x$ . In all our simulations, these three parameters are small (on the order of  $10^{-2}$  or less), so we neglect the higher-order terms. In Section 4.7.4, we empirically show the validity of the approximation, which gives  $\Delta W$  the much simpler approximate form

$$\Delta W = -\omega(Wx_0^2 - x_0x_1) + \sigma_W\eta_W \quad (4.17)$$

Because PaN interacts dynamically with its environment, the sensory stimulus it receives is a function of its motor output  $x_1$ . Indeed, as per the definition of the two-lever bandit task,  $s$  is given by

$$s(x_1) = s_0H(x_1) \quad (4.18)$$

where  $H(x)$  is the Heaviside step function and  $s_0$  is the value of the reward lever. \* We can rewrite two-neuron PaN dynamics as the following system of difference equations:

$$\Delta x_0 = -\alpha(x_0 - s_0H(x_1) + W^2x_0 - Wx_1) + \sigma_x\eta_{x_0} \quad (4.19)$$

$$\Delta x_1 = -\alpha(x_1 - Wx_0) + \sigma_x\eta_{x_1} \quad (4.20)$$

$$\Delta W = -\omega(Wx_0^2 - x_0x_1) + \sigma_W\eta_W \quad (4.21)$$

This is the form of a discretized system of stochastic differential equations (SDEs).

---

\*It was important that we made the substitution  $s \rightarrow s(x_1)$  at this step rather than in the energy function, because if we had made the substitution in the energy function before taking derivatives, we would have inadvertently enforced that  $x_1$  updates to change  $s$  so that it is closer to  $x_0$ . That would have been a non-local interaction, which is not permitted.

## STOCHASTIC DIFFERENTIAL EQUATION APPROXIMATION FOR THE TWO-NEURON SYSTEM

The three noise terms are now uncorrelated normal random variables, making the equations above amenable to analysis. We now demonstrate that the PaN algorithm can be interpreted as an SDE, giving insight into the long-term parameter distributions we should expect.

A general SDE takes the form:

$$d\mathbf{X}_t = \mu(\mathbf{X}_t, t) dt + \sigma(\mathbf{X}_t, t) d\mathbf{B}_t \quad (4.22)$$

Here,  $\mathbf{X}_t$  and  $\mu(\mathbf{X}_t, t)$  are vectors in  $\mathbb{R}^N$ .  $\sigma(\mathbf{X}_t, t)$  is an  $N \times M$  matrix, and  $\mathbf{B}_t$  is an  $M$ -dimensional Wiener process. If we discretize time in steps  $\Delta t$ , the Euler scheme implies

$$\Delta\mathbf{X}_t = \mu(\mathbf{X}_t, t) \Delta t + \sqrt{\Delta t} \sigma(\mathbf{X}_t, t) \mathbf{Z}_t \quad (4.23)$$

where  $\Delta\mathbf{X}_t = \mathbf{X}_{t+\Delta t} - \mathbf{X}_t$  and  $\mathbf{Z}_t \sim \mathcal{N}(\vec{0}, I)$ . Letting

$$\Delta \mathbf{X}_t = (\Delta x_0, \Delta x_1, \Delta W)^T \quad (4.24)$$

$$\mu(\mathbf{X}_t, t) = \mathbb{E}[(\Delta x_0, \Delta x_1, \Delta W)^T] \quad (4.25)$$

$$\sigma(\mathbf{X}_t, t) = \begin{pmatrix} \sigma_x & 0 & 0 \\ 0 & \sigma_x & 0 \\ 0 & 0 & \sigma_W \end{pmatrix} \quad (4.26)$$

we see that the PaN is a 3-dimensional SDE discretized by the Euler scheme with  $\Delta t = 1$ . In Section 4.7.4 we support this claim by showing that the system's long term behavior is consistent with the real PaN agent as we send  $\Delta t \rightarrow 0$  in simulations.

This result motivates us to reinterpret the two-neuron PaN agent, and all PaN systems in general, as SDEs with nonlinear drift terms and diagonal  $\sigma$  matrices. Note that the form of the SDE depends both on the architecture of the agent through  $E(t)$  and on the environment function through the form of  $s(\mathbf{x})$ . This interpretation holds even in stochastic environments where the stimulus vector  $\mathbf{s}$  is not completely determined by the network activities  $\mathbf{x}$ , as we can always approximate it as  $s(\mathbf{x})$  plus an additional noise term.

For the linear two-neuron PaN agent in the deterministic bandit task, the SDE that describes the system is given by

$$\frac{dx_0}{dt} = -\alpha(x_0 - s_0 H(x_1) + W^2 x_0 - W x_1) + \sigma_x \eta_{x_0} \quad (4.27)$$

$$\frac{dx_1}{dt} = -\alpha(x_1 - W x_0) + \sigma_x \eta_{x_1} \quad (4.28)$$

$$\frac{dW}{dt} = -\omega(W x_0^2 - x_0 x_1) + \sigma_W \eta_W \quad (4.29)$$

where here the  $\eta$  denote independent white noise processes.

#### 4.7.3 DERIVATION OF ATTRACTORS AND HOW THEY CORRESPOND TO EXPLORE OR EXPLOIT BEHAVIOR

We can gain a geometric intuition for PaN behavior by considering the attractor of the dynamical system (the set of all points that are stable without noise). To find the attractor, we calculate the fixed points and determine stabilities with linear stability analysis.

##### LINEAR STABILITY ANALYSIS OF THE FIXED POINTS

Ignoring noise and setting all derivatives to zero, we find that the fixed points are given by

$$\begin{cases} x_1 = W x_0 \\ x_0 = s_0 H(x_1) \end{cases} \quad (4.30)$$

We perform linear stability analysis to determine which fixed points form an attractor. Letting  $F(x_0, x_1, W) = \frac{dx_0}{dt}$ ,  $G(x_0, x_1, W) = \frac{dx_1}{dt}$ , and  $H(x_0, x_1, W) = \frac{dW}{dt}$ , the Jacobian is given by

$$J = \begin{pmatrix} \frac{\partial F}{\partial x_0} & \frac{\partial F}{\partial x_1} & \frac{\partial F}{\partial W} \\ \frac{\partial G}{\partial x_0} & \frac{\partial G}{\partial x_1} & \frac{\partial G}{\partial W} \\ \frac{\partial H}{\partial x_0} & \frac{\partial H}{\partial x_1} & \frac{\partial H}{\partial W} \end{pmatrix} = \begin{pmatrix} -\alpha(1 + W^2) & \alpha(W + s_0\delta(x_1)) & \alpha(x_1 - 2Wx_0) \\ \alpha W & -\alpha & \alpha x_0 \\ \omega(x_1 - 2Wx_0) & \omega x_0 & -\omega x_0^2 \end{pmatrix} \quad (4.31)$$

since  $\frac{H(x_1)}{dx_1} = \delta(x_1)$ , where  $\delta(x_1)$  is the Dirac delta function. The Dirac delta distinguishes two cases,  $x_1 \neq 0$  and  $x_1 = 0$ .

**STABILITY FOR FIXED POINTS WHERE  $x_1 \neq 0$**  We begin by analyzing the system's stability when  $x_1 \neq 0$ . This occurs if and only if  $x_0 \neq 0$  and  $W \neq 0$ . At the fixed points, if  $x_0 \neq 0$ , then  $x_0 = s_0$ . So, we evaluate the Jacobian where  $x_0 = s_0$  and  $x_1 = Wx_0$  and solve for its eigenvalues  $\lambda$ . We find

$$\lambda_1 = 0 \quad (4.32)$$

$$\lambda_{2,3} = \frac{1}{2} \left( -\omega s_0^2 - \alpha(2 + W^2) \pm \sqrt{\omega^2 s_0^4 + 2\omega\alpha s_0^2 W^2 + \alpha^2 W^2(4 + W^2)} \right) \quad (4.33)$$

The zero eigenvalue corresponds to the line of fixed points. Since  $\alpha > 0$  and  $\omega > 0$ , the second two eigenvalues are negative real numbers for all  $W$ . As such, the fixed points where  $x_1 \neq 0$  are all stable.

**STABILITY FOR FIXED POINTS WHERE  $x_1 = 0$**  Next, we analyze the case where  $x_1 = 0$ . Strictly speaking, the Jacobian is undefined here because  $\delta(x_1 = 0) = \infty$ . However, we momentarily consider the error function approximation of  $H(x_1)$  so that its derivative is not  $\delta(x_1)$  but rather a very sharp Gaussian. This means we can approximate our Jacobian by replacing  $\delta(0)$  with  $\Omega$ , where  $\Omega$  is an arbitrarily large but finite positive number.



When  $x_1 = 0$ , there are two sub-cases. Either  $x_0 = 0$  (a line where  $W$  can take any value) or  $x_0 = s_0$  and  $W = 0$  (a point). With the  $\Omega$  approximation, the eigenvalues of the Jacobian on the line where  $x_0 = 0$  are

$$\lambda_1 = 0 \tag{4.34}$$

$$\lambda_{2,3} = \frac{1}{2}\alpha \left( -2 - W^2 \pm \sqrt{W} \sqrt{4W + W^3 + 4s_0\Omega} \right) \tag{4.35}$$

The eigenvalues at the fixed point characterized by  $x_0 = s_0$  and  $W = 0$  are

$$\lambda_1 = 0 \tag{4.36}$$

$$\lambda_2 = -\omega s_0^2 - \alpha \tag{4.37}$$

$$\lambda_3 = -\alpha \tag{4.38}$$

In Equations 36-38, all nonzero eigenvalues are negative, so this fixed point is stable without noise. However, Equation 35 has an eigenvalue with a positive real part when  $W > 0$ . This is because  $\Omega \gg 1$ , so if the term in the square root is multiplied by a positive real value ( $\sqrt{W}$ ), then one of the eigenvalues will be a very large positive number.

We therefore conclude that the fixed points where  $W > 0$  and  $x_0 = 0$  are not stable. However, all other fixed points where  $x_1 = 0$  are stable without noise.

## THE ATTRACTOR OF THE TWO-NEURON NETWORK

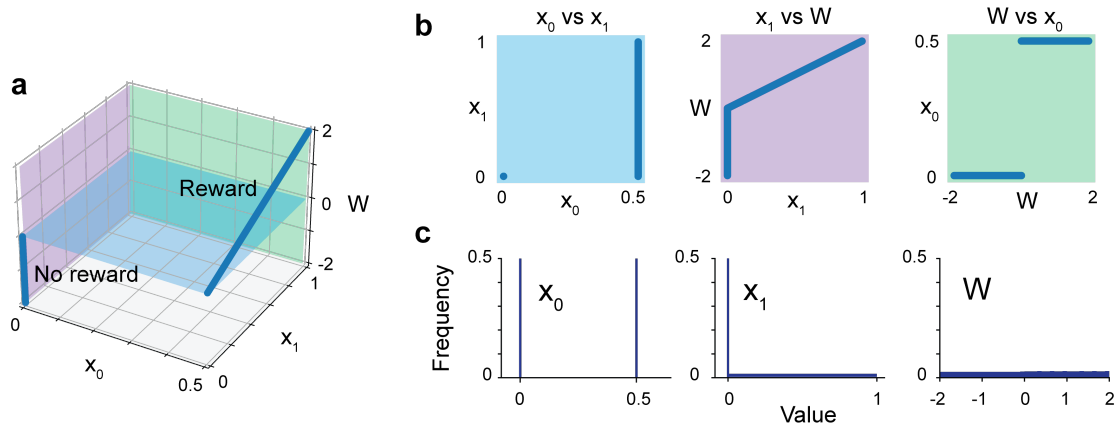


Figure 4.9: **Attractors without noise in the two-neuron network.** **a.** Reward and no-reward attractors plotted in the parameter space of  $(x_0, x_1, W)$ . **b.** Projections of attractors on each labelled plane, color-matched to planes in **a.** **c.** Frequency distributions of each parameter.

Combining the results of Section 4.7.3, we find that there is an attractor in the two-neuron network's parameter space given by

$$\begin{cases} x_1 = Wx_0 \\ x_0 = s_0H(x_1) \end{cases} \quad (4.39)$$

with the constraint that  $W \leq 0$  if  $x_0 = 0$ . We visualize it in Figure 4.9—it looks like two disconnected line attractors in the parameter space of  $(x_0, x_1, W)$ .

### WITHOUT NOISE, ALL PARTS OF THE ATTRACTOR ARE STABLE

Projected into the three planes, the attractor looks as in Figure 4.9b, with distributions of each parameter in Figure 4.9c. The histograms give the expected distribution of the agent's parameters

under the zero noise condition: networks stay on the attractor and do not favor any section of the attractor over another. These plots can be used as references for later figures.

#### WITH MOTOR NOISE, ONLY THE REWARDING PART OF THE ATTRACTOR IS STABLE

If we add noise, any part of the attractor that lies on the agent's "decision boundary,"  $x_1 = 0$ , is unstable. This is because when  $x_1 = 0$ , small amounts of noise added to  $x_1$  can switch the agent's lever choice, which alters whether PaN is driven to the reward or no reward attractor. The entire no reward attractor lies on this decision boundary, which means that it is unstable in the presence of motor noise.

The reward attractor, on the other hand, is almost entirely stable. Only the very edge (where  $x_1 = W = 0$ ) lies on the decision boundary, so only when PaN diffuses along the attractor to the edge can it select the zero reward lever and begin converging to the zero reward attractor.

We show these results in Figure 4.10. When there is no motor noise as in Figure 4.10a-f, networks do not prefer the rewarding action, best seen in the  $x_0$  distributions in Figures 4.10d, f, which are evenly split between inputs 0 and 0.5. When motor noise is added as in Figure 4.10g-j, networks do prefer the rewarding action. This is again best seen in the  $x_0$  distributions in Figures 4.10h, j, where now the 0.5 input is strongly preferred.

#### ATTRACTOR DYNAMICS EXPLAIN OVERALL EXPLORE/EXPLOIT BEHAVIOR.

Given the observations up until now, we expect the following behavior.

1. The agent should in general be converging to either the reward or no reward attractor, depending on its current lever choice.
2. In the presence of noise, the agent should spend more time at the reward attractor because it is more stable.

3. Since the agent can switch from converging towards the reward attractor to converging towards the zero reward attractor when it is near the reward attractor edge,  $x_1 = W = 0$ , there should be an occasionally traversed "bridge" along the  $x_0$  axis between the no reward and reward attractors.

Now we have a geometric explanation for the agent's long-term explore-exploit behavior. We can interpret the time that the agent spends diffusing along the reward attractor as its exploit phase. This phase should be longer than all other phases because the reward attractor is the most stable section of the agent's parameter space. We can interpret the time that the agent spends switching between the two attractors, along the  $x_1 = W = 0$  "bridge", as its explore phase. This phase should happen whenever the agent nears the edge of the reward attractor.

The rate at which the agent oscillates between explore and exploit phases therefore depends on the length of the reward attractor, its rate of diffusion along the attractor (which is proportional to the noise in the system), and the distance between the reward and zero reward attractors, which depends on the magnitude of  $s_0$ . Quantifying these rates will be interesting directions for future work.

With this picture, we can also explain why it takes time for the agent to learn to maximize reward. The learning phase is the transient phase that occurs while the agent converges to an attractor from its random initial point, which one can see in Video 1.

#### 4.7.4 EMPIRICAL VALIDATION OF ATTRACTOR INTERPRETATION UNDER VARIOUS NOISE CONDITIONS.

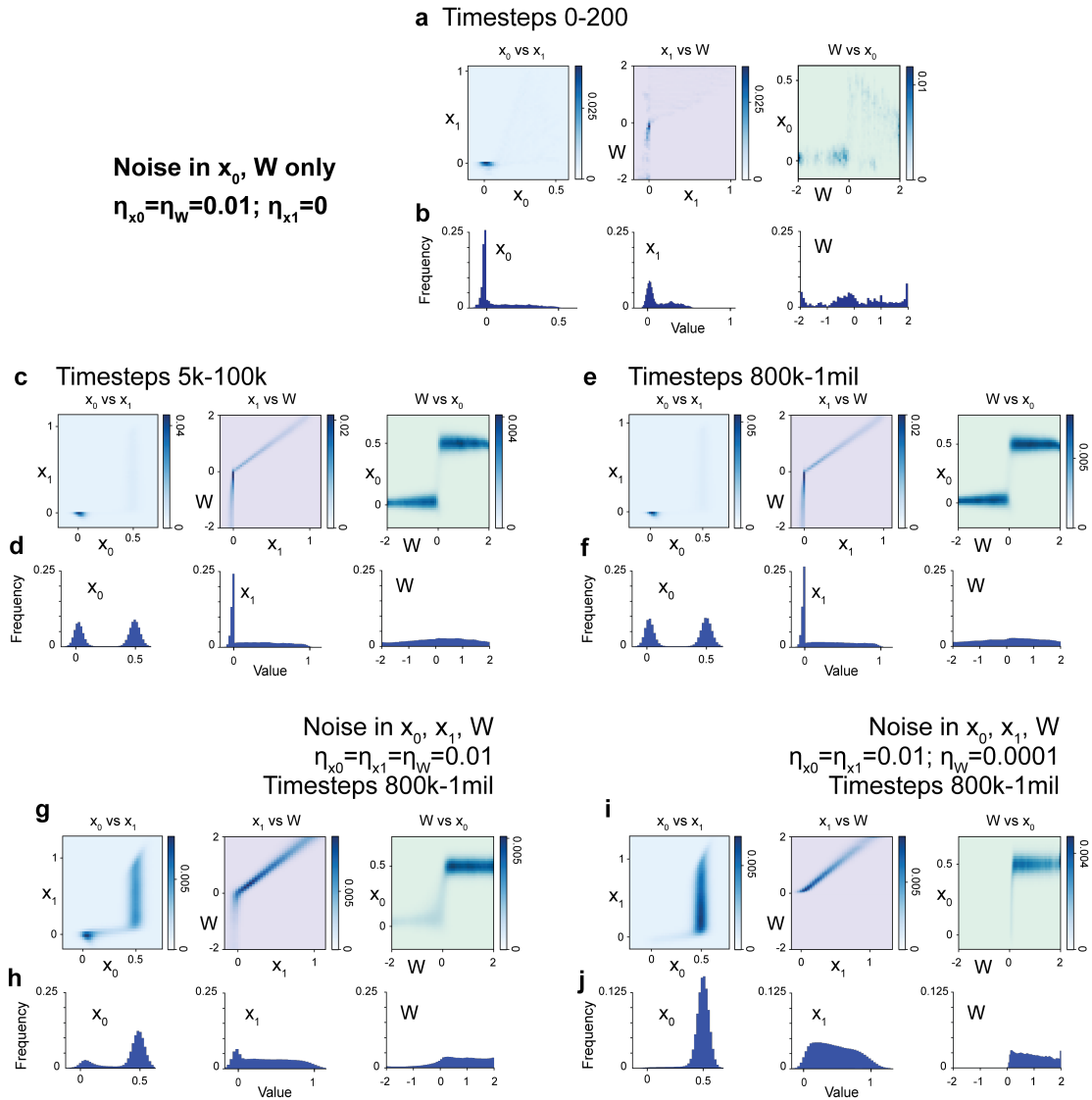
Here we show that the approximations made in Section 4.7.2, where higher-order terms are dropped in the difference equations, do not have a noticeable effect on network distributions. All plots in Figure 4.11 have noise in all three parameters:  $\eta_{x_0} = \eta_{x_1} = 0.01$ ;  $\eta_W = 0.0001$ . Plots are all for

timesteps 300k-500k. Figure 4.11a-b plot parameter projections and distributions when higher-order terms are included. There is little difference between Figure 4.11a-b and Figure 4.11c-d, when higher-order terms are dropped, suggesting that dropping these terms is a valid approximation.

We also support the writing of PaN as a system of SDEs by showing that different step sizes do not noticeably change parameter distributions. Figures 4.11e-f show plots with a timestep of 1 (see Section 4.7.2) and Figures 4.11g-h show plots with a timestep of 0.05. Again, distributions change very little between conditions.

#### 4.7.5 FUTURE WORK

The SDE picture of PaN opens some avenues of future exploration. In particular, future work could focus on solving the Fokker-Planck equation for the agent's equilibrium distribution in parameter space, opening the door to the prediction of long-term behavior with arbitrary architectures in any environment. This would allow us to explore agents' behavior in different tasks with varying number of hidden neurons, changing network connectivities, or with varying learning rates  $\alpha$  or  $\omega$ .



**Figure 4.10: Motor noise (noise at  $x_1$ ) is necessary for reward-seeking behavior.** a-f look at networks without motor noise,  $\eta_{x_1} = 0$ . Networks are still settling in the first 200 timesteps (a-b) but by timesteps 5k-100k (c-d), they have settled into their long-term distributions (e-f). Networks evenly distribute between choosing both actions, most easily seen in the  $x_0$  histograms in (d) and (f). **g.** With motor noise, networks prefer the rewarding action, which is again best seen in the  $x_0$  distribution in **h.** **i-j.** Noise can be tuned to further push networks toward exploitation. With less weight noise, networks almost entirely choose the rewarding action (**j**).

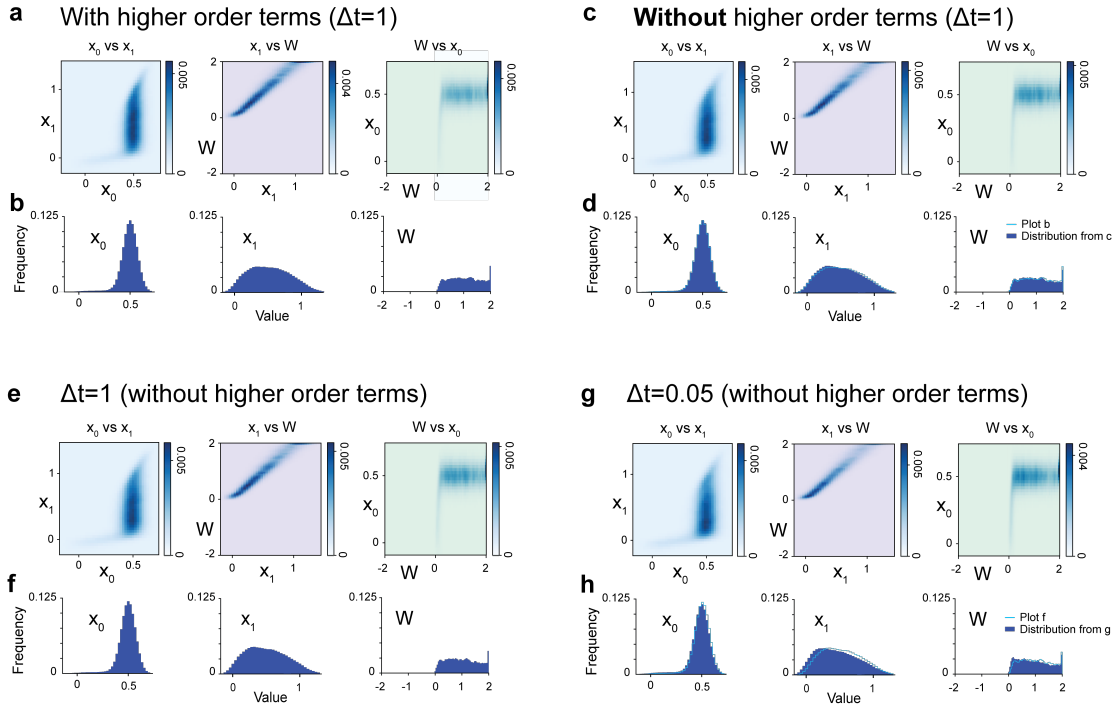


Figure 4.II: **Support for approximations made in Sections 4.7.2 and 4.7.2.** All plots are for two-neuron PaN networks with noise in all three parameters:  $\eta_{x_0} = \eta_{x_1} = 0.01; \eta_W = 0.0001$ . Plots are all for timesteps 300k-500k. **a-b.** Projections and distributions, respectively, for parameters when higher-order terms are included in the difference equations in 4.7.2. **c-d.** Projections and distributions when higher-order terms are dropped. Differences between a-b. and c-d. are very slight. **e-f.** Timestep of 1 as in Section 4.7.2, compared to a timestep of 0.05 as in **g-h.** There is again very little difference, suggesting that PaN can be written and understood as a discretized SDE.







Just because you do it  
doesn't mean you always will.  
Whether you're dancing dust  
or breathing light  
you're never exactly the same,  
twice.

Yrsa Daley-Ward, *bone*

I am not excessively fond either of salads or fruits,  
except melons. My father hated all sorts of sauces;  
I love them all.

Montaigne, *On Experience*

# 5

## Building up a very small brain: Beyond optimization

THE PREVIOUS CHAPTER WAS NECESSARY, BUT NOT SUFFICIENT for the beginnings of a model of a living nervous system. The first thing we had to show that a predictive algorithm could do,

which hadn't been satisfactorily shown before, was that it could autonomously learn to seek out reward and maximize signals.

But reinforcement learning already maximizes signals. It's the stuff that RL can't do that sent me down this path to begin with. So in this chapter, I map out a few of PaN's more unusual capabilities, like adaptability, flexible task-switching, and preference bias. Together, these are unusual only with respect to machine learning algorithms. In animal intelligence, they are everywhere.

## 5.1 PaN ADAPTS TO BOTH INTERNAL AND ENVIRONMENTAL CHANGES

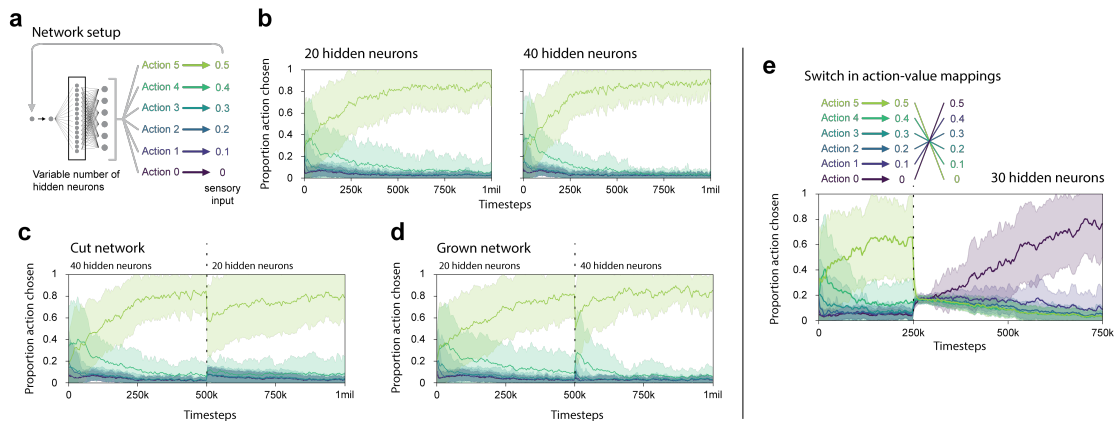
### 5.1.1 PaN AUTOMATICALLY ADJUSTS TO ARCHITECTURAL CHANGES

PaN can adapt itself to changing hidden layer sizes in the bandit task from Figure 4.6a. In Figure 5.1a-b, we first show that networks with 20 or 40 hidden neurons can still learn the optimization task. Then in Figure 5.1c, we run networks with 40 hidden neurons and randomly cut 20 neurons at 500k timesteps, showing that networks adapt and then continue to select the highest-reward action most of the time. Similar adaptation is shown in Figure 5.1d, where networks start with 20 hidden neurons and are then doubled to 40 neurons at 500k timesteps.

### 5.1.2 PaN ADAPTS TO EXTERNAL CHANGES

#### DIRECT ACTION-TO-REWARD MAPPINGS.

We now test PaN's performance as a continual learning agent using dynamic environments<sup>2</sup>. Figure 5.1e again begins with the 6-action setup in Figure 4.6a, now with 30 neurons in the hidden layer. At 250k timesteps, the reward associations are flipped—the action that was previously most rewarding is now the least rewarding, and vice versa. PaN autonomously learns to exploit the new best action over the following 500k timesteps.

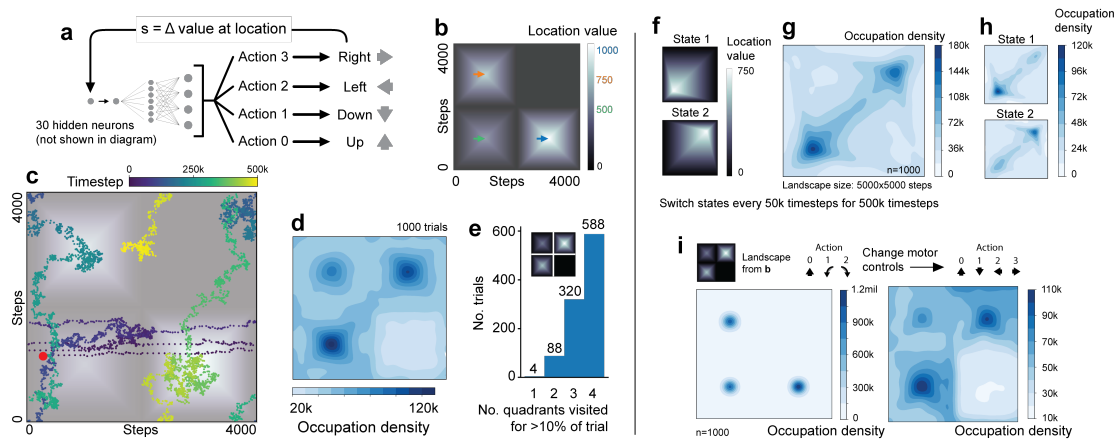


**Figure 5.1: PaN networks adapt to internal and environmental change in bandit environments.**

**a.** PaN networks are put in a 6-action environment with linearly spaced rewards. All plots are for 50 random seeds; means and standard deviations in shaded regions. **b.** Networks with 20 (left) or 40 (right) hidden neurons can learn to select the most rewarding action (Action 5). **c.** If networks begin with 40 hidden neurons and are cut to 20 hidden neurons at 500k timesteps, they continue to learn and adapt to the new architecture without external cues. **d.** Likewise, networks adapt when their 20 hidden neurons are doubled to 40 at 500k timesteps. **e.** A network with 30 hidden neurons can adapt to a bandit task that switches reward values midway through the trial. The reward switch occurs at 250k timesteps and the simulations are for 750k timesteps total.

### OPEN-FIELD SEARCH TASKS.

For a more complex environment, we design a task where actions correspond to movements in a reward landscape (5.2a). Instead of fixed rewards, we set the change in reward between an agent's previous location and its new location as the input  $s$ . This one-timestep difference mimics an immediate "adaptation" at the sensory level to the previous reward value. When agents are placed in the landscape in Figure 5.2b, they spend more time at locations with greater reward values (sample track in Figure 5.2c; occupation density in Figure 5.2d). Agents do not fixate on one local maximum but usually explore multiple peaks during a single trial (Figure 5.2e). In fact, most agents initialized at random locations spent at least 10% of the trial in at least 3 different quadrants of the landscape, and many agents spent time in all 4.



**Figure 5.2: Open-field search task.** **a.** Actions correspond to four discrete movements in the reward landscape **b.** **c.** Every hundredth point for a single trial. Starting point marked in red; see Video 2 for animated track. **d.** Density of occupation for 1000 trials, 500k timesteps each, showing that networks spend more time in higher-value regions. **e.** The number of quadrants occupied for >10% of the trial was counted for each agent. 908 of 1000 agents visited at least 3 quadrants rather than getting stuck at local maxima. **f.** Agents were placed in landscapes that switched between two states every 50k timesteps for 500k timesteps total. **g.** Occupation density over all trials, showing greater occupancy at each maximum. **h.** When (g) is separated by the current landscape state, we see localization at the current maximum. **i.** We change the motor interface of the agent from rotational controls (forward, turn CCW, turn CW) (timesteps 0-500k) to translational controls as in a (timesteps 500k-1mil). Occupation density plots show that agents are able to adapt and continue to seek reward. All experiments in this figure required an upper bound of 2000 CPU hours and 3.5 GB of storage.

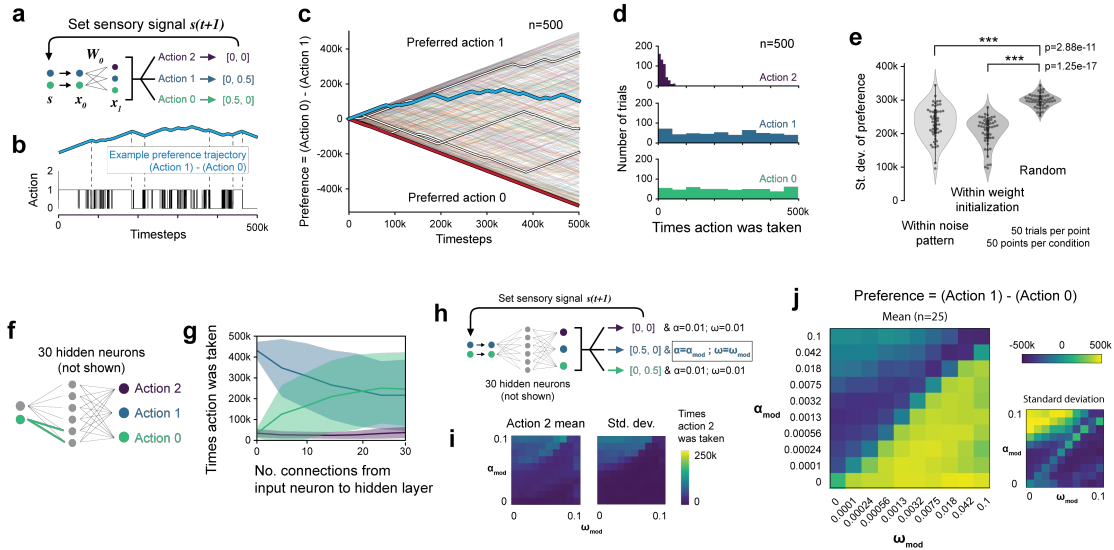
AGENTS ADAPT TO CHANGING LANDSCAPES. Figure 5.2f shows a landscape where the reward peak flips between two states every 50k timesteps. The agents distribute their time between the two locations, 5.2g, and Figure 5.2h shows that agents localize to the current maximum.

PAN CAN ADAPT TO CHANGES IN MOTOR INTERFACES. Figure 5.2i places agents in the reward landscape from Figure 5.2b. For the first 500k timesteps, agents move using a rotation scheme: networks can take three possible actions corresponding to forward movement, 90 degree counterclockwise rotation, or 90 degree clockwise rotation. Then a motor neuron is added and the interface is changed to the standard translational scheme in Figure 5.2a. Agents seek out higher-reward regions of the landscape throughout the simulation.

## 5.2 PAN AUTONOMOUSLY FORMS TASK PREFERENCES

### 5.2.1 TASK PREFERENCES VARY BETWEEN AND WITHIN NETWORKS

PaN displays an autonomy in goal selection that emerges from local interactions. We show this with a network of two input neurons corresponding directly to two output neurons, with one additional output neuron that provides no signal back to the network as a control (Figure 5.3a). Goal preferences vary over time, shown by plotting preference trajectories, defined as the difference between cumulative Action 1 and Action 0 selections at each timestep. Figure 5.3b has an example preference trajectory, top, for the action choices below. Figure 5.3c shows that random seeds can stick to one action for long periods of time, as in the red trace, or they can change preferences often, as in the blue trace. Preferences are distributed between the rewarding actions evenly (Figure 5.3d).



**Figure 5.3: Networks exhibit preferences that can be biased.**

**a.** Network in two-task environments. **b.** Actions can be plotted as a preference trace, defined as the cumulative number of Action 1 selections minus cumulative Action 0 selections. The blue preference trace in **b**, top, is plotted in **c**. as part of 500 random seeds. Seeds in **c**. are randomly colored and each trial is 500k timesteps. Several random tracks are highlighted: some networks strongly prefer one action for extended periods of time (red trace) while others change preference rapidly (blue trace). **d.** Networks evenly distribute between Actions 0 and 1 but avoid the non-rewarding Action 2. **e.** Specific noise patterns and weight initializations significantly bias network preference, as measured by the difference between the number of times a network selected Action 1 and Action 0. P-values for two-sided Wilcoxon rank-sum test. **f-g.** We gradually increase the number of connections from the input neuron corresponding to Action 0 and show that the more dense the connections from an input neuron to the hidden layer, the more likely networks will be biased toward the corresponding action. **h.** We change the learning rate for the entire network if Action 1 is selected. Learning rates are by default 0.01 otherwise. **i.** Networks avoid the non-rewarding Action 2. **j.** Networks strongly prefer Action 1 if the modulated activity learning rate  $\alpha_{mod}$  is lower than the modulated weight learning rate  $\omega_{mod}$ . Experiments in figure together required upper bound of 2100 CPU hours and 150 MB for storage.

## 5.2.2 TASK PREFERENCES CAN BE BIASED IN BIOLOGICALLY RELEVANT WAYS

### NOISE AND INITIALIZATION

We plot the standard deviations of preference within noise patterns for differently initialized networks in Figure 5.3e, left, as well as standard deviations of preference within the same weight initializations using different noise patterns in Figure 5.3e, center. We compare both cases to random noise patterns and initializations in Figure 5.3e, right, showing that preferences are biased by specific patterns of activity and weight noise.

### CONNECTIVITY

In networks with two input neurons and 30 hidden neurons, we increase the number of connections from the input neuron corresponding to Action 0 and keep the other input neuron fully connected (Figure 5.3f). The sparser the connectivity of the input neuron, the less the network prefers its corresponding action (Figure 5.3g).

### MODULATED LEARNING RATES

Next, we change the activity learning rates  $\alpha$  and weight learning rates  $\omega$  when networks choose Action 1 (see Algorithm 1 for variable definitions and Figure 5.3h for illustration). Networks retain their reward-seeking behavior by avoiding the non-rewarding Action 2 (Figure 5.3i). But interestingly, Figure 5.3j shows that when the modulated activity rate  $\alpha_{mod}$  is less than the modulated weight learning rate  $\omega_{mod}$ , networks strongly favor the action responsible for the change (Action 1). In other words, if an action results in faster weight learning than activity learning, then that action is preferred (see <sup>39</sup> for biological plausibility).



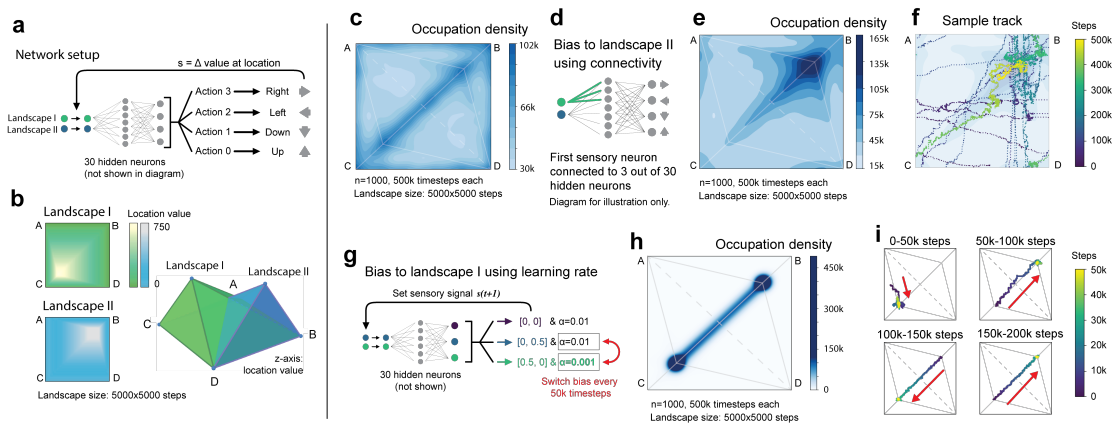


Figure 5.4: PaN networks in open-field search tasks can be biased.

a. Networks can take translational movements in b. a landscape with two overlapping value gradients. c. An unbiased network chooses either of the value maxima. d. When networks are biased by reducing the connections from the input neuron corresponding to Landscape I values, e. networks overwhelmingly prefer Landscape II. f. Sample track for e. g. Next we bias networks by reducing activity learning rates when they receive positive Landscape I signal. Every 50k timesteps, the bias switches to the other landscape for a total of 500k timesteps. h. Networks move between landscape maxima, and a sample track in i. shows that networks move quickly toward the biased landscape, with arrows to emphasize direction of movement. Experiments in this figure required upper bound of 1500 CPU hours and 3 GB.

### 5.2.3 OPEN-FIELD SEARCH TASK WITH TWO TYPES OF REWARD

We then evaluate whether we can bias PaN’s preferences in the open-field search task. We add a second input neuron to the network in Figure 5.2a, shown in Figure 5.4a, and allow it to freely behave in the landscape in Figure 5.4b. Without any bias, networks do not consistently prefer either maximum over the other (Figure 5.4c). When we bias networks toward Landscape II by decreasing connections from the input neuron corresponding to Landscape I (Figure 5.4d), networks prefer the Landscape II maximum (Figure 5.4e-f). When we use the learning rate modulation method to bias the network, decreasing the activity learning rate for signals from the biased landscape (Figure 5.4g), we find that networks can form extremely strong preferences for the currently biased landscape. Preferences were rapidly switched by changing the biased landscape every 50k timesteps for 500k timesteps total (Figure 5.4h), with an example track in Figure 5.4i.

## 5.3 DISCUSSION

We show that neural networks using only local prediction and noise can exhibit flexible, emergent, and autonomous behavior. Our algorithm PaN can switch between exploration and exploitation without external prompting, and these rich behaviors are explained by attractor dynamics. PaN is distinct from classical RL agents in that it uses entirely local predictive updates rather than an environmental reward function; consequently, it has the potential to select and shape its own goals. We show that PaN networks can have preferences and can change them over time, similar to observations in animals (see fruit fly behavior in <sup>181</sup>). And because of how PaN differs from other algorithms that learn through environmental interaction, we define a preliminary framework called *self-supervised behavior* (SSB) and elaborate upon it in Section 5.4.

Our algorithm is biologically plausible, has the potential to account for autonomy in living nervous systems, and shows that emergent behavior is capable of switching between exploration and

exploitation, where reward signals can be maximized during exploitation. In our estimation, these features make PaN a possible starting point for more expressive algorithms that better resemble natural intelligence.

### 5.3.1 LIMITATIONS

We have only tested PaN in simple environments. We have not demonstrated PaN's use of some of the advantages of predictive coding models found in the literature<sup>183</sup>, such as their ability to learn useful sensory representations; nor have we shown any capacity of PaN to form associative memories, which are required in more complicated tasks, or how aversive stimuli might be represented. Moreover, deeper PaN networks, ones with recurrent connections, or networks following Dale's Law<sup>41</sup> may allow for more complex behavioral dynamics that PaN is incapable of in its current form. These are all important missing components and will be interesting directions for future work.

## 5.4 SELF-SUPERVISED BEHAVIOR: A NOVEL FRAMEWORK FOR ENVIRONMENTAL INTER-ACTION

At the moment, reinforcement learning (RL) is the primary framework for artificial agents that learn from and act within an environment. However, critics have pointed out that RL may be limited, in part because it requires the foreknowledge of an appropriate goal<sup>94</sup>. While we do define a single predictive loss for PaN (Equation 4.1), the loss is determined internally and updates include noise. This is in contrast with RL, where learning is set up as a way to maximize expected return from external Markov Decision Processes<sup>189</sup>. We argue that our algorithm is distinct both in its definitions and resultant behaviors, which are never designed to achieve an optimal action trajectory (the authors of GFlowNets make a similar argument to distinguish their work from RL; see<sup>16</sup>).

PaN is only a preliminary model, but we believe it has the potential to be extended to autopoietic computational systems. Most modern machine learning algorithms cannot by definition<sup>94</sup>, because they use backpropagation to learn and backpropagation requires tasks or objectives that are predefined. In self-organized, self-managing, and scalable systems<sup>28</sup>, however, goals should emerge on their own. We therefore suggest that PaN belongs to a new framework, which we call self-supervised behavior (SSB), named because PaN's energy function resembles that of models that use self-supervision over time<sup>152</sup> to learn sensory representations. Here we posit features that an SSB model should possess.

### 5.4.1 FEATURES OF SELF-SUPERVISED BEHAVIOR

- **Updates are based on internal, local loss, with the absence of an environmentally defined global utility function.**

An SSB agent should exhibit emergent goals and behaviors rather than ones that are predefined. Emergence allows for diversity in behavior both between and within individual agents,

as well as a great capacity to adapt to changing circumstances.

- **Autonomous operation that can execute continual learning in dynamic situations.**

SSB should remain autonomous. For an agent to function—much like a living organism—it should not require continual direction from an external entity. An SSB agent’s ability to carry out continual learning should be entirely self-driven.

- **Evidence of apparent goals that are self-selected, and evidence of ability to improve performance on these goals.**

Again, goal selection should emerge from an SSB agent’s internal workings rather than an external definition. The agent should also be able to learn over time to better achieve these goals, whether through increasing a reward signal, which we have shown here, or by learning effective motor sequences on its own, which we leave for future work.

#### 5.4.2 WHY DEFINE AN ALTERNATIVE FRAMEWORK?

Defining SSB allows us to scaffold ideas and results from PaN without relying too much on the details of its implementation. We do not believe PaN’s particular energy function, architecture, and training methods are unique in their ability to produce the results we show in this paper. At the same time, we believe PaN’s use of emergence to generate behaviors makes it fundamentally unlike most other algorithms.

Namely, we feel it is important to define a framework for agents behaving in environments that *does not* rely on achieving some optimal course of action. RL and even active inference begin from the assumption that actions taken in an environment serve to optimize some set task or environmental function (minimizing overall “surprise” in active inference<sup>156</sup>, for example). But one might consider the great diversity, creativity, and self-driven nature of animal behavior, and that all of it might suggest that learning algorithms do not need to optimize an environmental outcome (outside

of fulfilling the basic constraints of survival and reproduction) in order to perform useful, adaptive, or simply interesting behaviors.

Thus we believe that defining SSB as a framework is important and ongoing work, for PaN and other emergent algorithms. Our hope is that suggesting this alternate framework for agents acting in environments will encourage researchers to consider that explicit optimization is only one way of learning from an environment, and to explore other, possibly richer ways of interacting with the world.







# 6

## Conclusion

My goal at the end of graduate school, just like it was at the start, is to understand the brain by building a very small working one. I've come at this problem from several angles over the last five years—really, the last ten, since I first started working with Dr. Allan Gulledge my freshman year of college, and then the year I spent at the University of Cambridge with Dr. David Parker. I can separate my approaches into several phases. None of these phases I explored to an adequate depth, and each approach could have been worth pursuing in its own right. They just mostly didn't answer the

questions I wanted to ask, which was how one might build a nervous system with the complexity, autonomy, and *interestingness* of behavior that anyone can see in the smallest and largest animals alike.

## 6.1 RESEARCH TRAJECTORIES, REFRAMED

First, in undergrad, I thought that maybe biophysical neuron models would reveal something about how behavior as a whole was generated. I studied the properties of specific glutamate receptors and how they combined to integrate information.<sup>127</sup> While these were interesting chemical and electrical systems to study, I did not feel that detailed biophysical models helped me understand general animal behavior.

Next, during my master's degree, I went to the opposite extreme and tried averaging over the properties of neurons. I quickly realized that although there was probably a lot to learn from this statistical physics approach, the required math and physics were beyond me. I also felt, vaguely, that I was interested in how animals carried out specific computations and behaviors that were possibly *too* specific to be captured by averaged quantities.

After that, when I started my Ph.D., I thought that trying to build an artificial nervous system from scratch might be too difficult considering the efforts that had already been made<sup>173</sup> and the data that seemed to be required.<sup>77</sup> Instead I tried to add onto an existing animal's brain, that of *Caenorhabditis elegans*, in the hope that combining artificial and biological neural networks would tell me something about how natural intelligence functions and grows, neuron by neuron.

But instead of truly adding to a living nervous system in the way that would have occurred had the animal grown a new set of neurons, the combination of natural and artificial intelligence emphasized to me the dramatic underlying differences between how animals and how reinforcement learning algorithms operate—reinforcement learning being the overwhelmingly dominant approach (if not the only approach) to AI interaction with an environment. Modern machine learning, I

thought, was not exactly what I wanted either.

After the *C. elegans* project and a brief exploration of task allocation in branched architectures, I thought I had a better sense of what was dissimilar between RL and the behavior of very small animals. I tried to identify a few key principles and put them together to make a new kind of algorithm that I hoped was much more like how animals work, which exhibited the traits listed in the Introduction: autonomy, reward-seeking, adaptability, and the capacity to devise its own goals. Many pieces are still missing from this model (see Appendix A.1), but I am very excited to see where it might go. After sampling a few different angles over the years, I am finally, cautiously optimistic about this one: an approach based on emergent behavior, attractor dynamics,<sup>105</sup> and predictive principles that combine to make a new kind of algorithm that at last, in my opinion, has the first glimmers of natural intelligence.

## 6.2 TECHNICAL SUMMARY OF THESIS

### 6.2.1 CHAPTER ONE

I describe the methods used to integrate a deep reinforcement learning agent with the living nervous system of *C. elegans* using optogenetics. I show that agents can learn appropriate policies to navigate animals depending on their sites of integration.

### 6.2.2 CHAPTER TWO

I present several applications of the technology in Chapter 1: first, probing and mapping the roles of different sets of neurons in a goal behavior; second, studying RL efficacy when control methods are noisy; third, testing the combined capabilities of the animal and agent to generalize to novel environments and food search tasks.

### 6.2.3 CHAPTER THREE

This chapter describes how two tasks that are better solved with different priors (color estimation and frequency labelling of Gabor filters) are allocated to differently shaped branches in branched neural network architectures.

### 6.2.4 CHAPTER FOUR

Here I present a new kind of learning and behavioral algorithm that operates using principles of emergence. I show that with just local prediction and noise, an artificial network is capable of reward-seeking behavior without a global environmental reward. The network can balance exploration and exploitation on its own, and we derive attractor dynamics to explain its behavior.

### 6.2.5 CHAPTER FIVE

In the final chapter, I show that the prediction and noise algorithm is highly flexible with respect to internal, architectural changes, as well as changes in the structure of the environment. It also exhibits individual preference for different tasks, and this preference can be biased in biologically relevant ways (patterns of noise, architecture, neuromodulation).

## 6.3 FINAL THOUGHTS; A VIGNETTE FROM THE HISTORY OF ML

Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton won Fei-Fei Li's ImageNet contest with convolutional neural networks in 2012. In a special 2016 issue of *The Economist*, the journalist Tom Standage wrote that "the ImageNet results showed what deep learning could do. Suddenly people started to pay attention, not just within the AI community but across the technology industry as a whole."<sup>184</sup>

But before this happened, image processing researchers thought they were well on their way to solving the problem, by building bigger and ever more complicated classifiers stacked on features stacked on descriptors.<sup>197</sup> It must have been hard to argue that increasing the complexity of the algorithms *wouldn't* work, because one could always patch any issues that arose.

After I spent some time with reinforcement learning for the work in Chapters 1 and 2, and after talking to people more experienced with RL than I was, I couldn't help but see parallels between how RL research was going and how computer vision research had gone in the pre-convolutional neural network days. RL is *hard*. Nontrivial tasks require teams of experts to coax the algorithms into working,\* and even relatively simple RL algorithms tend to have a lot of specialized modules working in concert.<sup>57,190,190a</sup> If you have unlimited data, then RL is not as hard, but there are only so many domains with perfect simulators or that volume of information available. Richard Sutton's Bitter Lesson<sup>188</sup> has played out time and time again in deep learning<sup>†</sup>, but the reality is that few entities in the world have access to the scale of data required to keep pushing The Bitter Lesson forward. And we're now seeing that even those entities are finding it hard to keep up with the pace of data necessary to continue improving their models.

Just as increasing model complexity took us quite far in the domain of computer vision, increased data has taken us extremely far in artificial intelligence. And yet—large language models often make mistakes when they need to count to five. Neural networks cannot generalize linear regression except with comically large datasets. Vision networks find it very hard to generalize to small out-of-distribution changes in angle, lighting, etc.<sup>169</sup> But bees can do all of the above with much less raw

---

\*This was the language Timothy Lillicrap of DeepMind used in a Kempner Institute talk in May 2023: there is always lots of *coaxing* involved.<sup>130</sup>

†This is an excellent essay in which Richard Sutton, one of the pioneers of RL research, makes an argument/observation about what has led to success in deep learning. The first sentence: “The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin.” In other words, success comes from general methods and vast amounts of computation and data.

data and seemingly fewer parameters.<sup>34,69‡</sup> Small animals with a million neurons, like the bee, or far fewer, like *C. elegans*, strongly suggest there are important concepts we are missing from intelligent algorithms. Perhaps, like in 2012, the question is no longer how to scale the same types of algorithms up, but rather where to look for new kinds of algorithms entirely.

My motivation has never been to make better artificial intelligence. All I've wanted is to understand animals and their nervous systems. But I do find it compelling that the limits of modern AI are often where animals excel, like in the ability to generalize, in data-scarce regimes, and the ability to learn highly adaptable and robust control algorithms. And I do still hold with Feynman in my belief that if we cannot build a natural intelligence, no matter how small, or cannot definitively show that we know how to build one, then we cannot be very comfortable claiming that we understand it.

My goal, in the end, would be a unified theory of natural intelligence, which I assume would come with working models. I don't know how far off we are from such a unified theory, or how confident I am that one exists.<sup>§</sup> This dissertation documents my efforts to wade through the uncertainty, toward finding some coherent theory—a false start and an auspicious second path—and it represents just the beginning of my own search for this scientific holy grail.

---

<sup>‡</sup>Yes, one can argue that evolution has provided eons of data that are hardwired in the insects' brains. If this is true, if evolution is the source of the boundless data we cannot seem to scrape enough of, then it's a good thing lots of researchers are still working on the big data approach. But see Appendix A.2.3 for my related thoughts on the role of evolution.

<sup>§</sup>I am *pretty* confident, based on the dramatic plasticity of animal brains and the diversity in their forms and functions. But I still wouldn't say I'm certain, because there's always the possibility that evolution the tinkerer has taken advantage of pieces I don't account for at all in this thesis, like glia (neural support cells) or neural vasculature. Take for example the amount of bloatware in every genome.<sup>55</sup> But these are only slim possibilities, in my opinion.









## Speculation

Here is part of an email that my English professor, James Wood, a renowned literary critic, sent to the class near the end of the semester.

The deadline for the final paper will be Saturday May 4th. [...] This final project should be written in enthusiasm and pleasure; it shouldn't feel like a chore, but more like an exploration.

It may be risky putting nascent thoughts in a public place, but then again, why not? In the spirit

of James Wood, science should be an exploration. Academic writing does not provide that many avenues of publishing real, unfettered speculation, especially for beginning scientists, so I will take the chance to expound on my thoughts here. If nothing else, these topics have been fun to think about (and discuss with my student/colleague Jonah Brenner, who had no choice but to listen to my rambles).

## A.1 WHAT IS STILL MISSING FROM PAN?

### A.1.1 MEMORIES

I want to be very clear about what is still missing from PaN. Here is what it can do: it can seek out reward and it has some semblance of autonomy. It can switch tasks flexibly and it can be guided by biologically relevant mechanisms (Figure 5.4). But here are some crucial things it cannot do: it cannot form or retrieve associative memories, which is an essential feature of a working nervous system. It also cannot learn sequences of actions or have a record of its location in tasks like the open-field search tasks in Figure 5.2 or 5.4. Sequence learning and location representation will almost certainly require recurrent connections in the model. We will have to work on how to best add that in. It is an engineering problem, I think. The memory problem is the more conceptually complicated one.

Fortunately, though, I think that other people have already solved it. In future projects we will have to figure out how to properly incorporate their solutions, but I am optimistic that they have done much of the hard work already.

In Khona and Fiete (2022),<sup>105</sup> the authors describe various kinds of attractors that could perform useful computations. Discrete attractors can serve as the basis for memory storage, a possibility explored by other research with recurrent neural networks. For instance, in Chapter 4, I mentioned that Clark and Abbott (2024)<sup>37</sup> had shown that coupled activity and weight updates allowed for a broader set of computational possibilities than either alone. What I did not mention was that they,

along with earlier work<sup>10,138,137</sup>, were actually using Hebbian and anti-Hebbian plasticity rather than predictive or self-supervised update rules.

Hebbian plasticity can be a way to enforce variance in neural representations, which is a trait we need to add to PaN to allow for memory storage. Otherwise, the predictive energy of PaN pulls all representations to become the same over time.<sup>74</sup> And the purpose of the Clark and Abbott paper was really to show that the activity-weight coupling, along with Hebbian updates, allowed for a “new mechanism of working memory.” Thus, my next research goal is to use Hebbian learning to combine the decision-making ability of PaN with the memory storage of these other models.<sup>74,57,37</sup>

### A.1.2 SLEEP

Sleep is a bit of a mystery to me. All I can say for now is that I would be very surprised if I found that future iterations of PaN did not require a sleep-like phase to maintain its performance. Even in *C. elegans*<sup>148</sup> and Hydra,<sup>98</sup> we have observed sleep-like states. It is such an evolutionarily costly feature and seems to have so many computational benefits,<sup>109</sup> that I think it is probably necessary to maintain the capacity of a continuously running nervous system.

I am not nearly confident enough to present real data on this yet, but I will mention: in very preliminary simulations of PaN with a Hebbian learning component, I found that neuron weights were reaching some stuck stable state, often saturated at the value I'd clip weights at (the maximum magnitude possible). When I was trying to get these modified PaN networks to learn context-dependent actions, which would require some implementation of memory, they appeared to learn in the early stages of training, but once the saturation state was reached, performance would crash.

To fix this issue, I wonder whether one could leverage the weight saturation, perhaps using the right recurrent connections, to turn the stuck state into a sleep state with globally propagating waves. The initiation of these waves would still have to be autonomous for the model to continue posing as a possible model of natural intelligence. So would their termination. But I think the waves

themselves could be a mechanism to redistribute neuron weights. If we take the role of sleep in *C. elegans* and larger animals as an example, the redistribution ought to both store memories and refresh the capacity to learn new ones.

### A.1.3 PAIN

I have been hitting my head against this problem for almost a year now, basically since the first conception of PaN while dawdling around the summer after my second year. The problem of pain in PaN boils down to this: if PaN operates by maximizing signal-to-noise ratios (SNRs), and anything that maximizes SNR is seen as a rewarding action that is then reinforced to produce reward-seeking behavior, then how do I represent anything aversive? How do I build in a way to communicate a very strong signal that PaN definitely *doesn't* want, if that is the opposite of how it fundamentally works?

I've had a few guesses over the last couple months, before the release of the PaN preprint.<sup>125</sup> Originally I'd wanted to include a solution to the aversive signaling problem in the paper, because it's such an important missing piece, but unfortunately, none of the solutions I thought were obvious at the start have worked out. The thing is, the problem of pain could kill the model—if PaN or a PaN-like agent can't represent and avoid aversive signals, then that is too large a mismatch between the model and small nervous systems; PaN will go belly-up. But I don't think I'm quite there yet.

My initial guess was that if we simply increased the weight noise of “nociceptive” input neurons, we could get aversive behavior. Then any signal would lead to high unpredictability. I thought this would work because of Figure 4.8b and e, where high weight noise does lead to avoidance of the input signal.

But this only worked when the entire network had very high weight noise, and any signal at all sent a lot of noise through the network. If I tried to make only one input neuron noisy, then the network appeared to only consider the expected value of the input, possibly because the activity

settling was slow enough for fast fluctuations not to matter. One solution could be to increase the activity learning rate for the nociceptive neurons, and I haven't fully tried that yet. But I also suspect that more might be required for truly aversive behavior.

If you were watching an animal take actions, how would you know that it's found one of the actions to be painful? You might infer it if the animal always avoids taking one action even during exploratory phases. That could suggest that even during exploration, a painful action must be actively inhibited. I did search the literature for pain circuits in *C. elegans*, but I didn't find it that helpful, because as usual, the literature describes the circuits that exist rather than how they might be learned. One thing I did notice, though, was that models of pain and pain circuitry almost always includes some kind of reciprocal inhibition.<sup>71</sup> So in future work, I'll try to increase the activity learning rate for nociceptive neurons, and if that doesn't get me as robust avoidance behavior as I would like, I'll look at how one might design learning rules that include inhibitory circuitry (that are still local and based on energy functions). In the spirit of the rest of the PaN work, I'd like to avoid hardwiring any circuits, to adhere to the search for a set of general principles for natural intelligence.

## A.2 WHAT COULD THE EMERGENT BRAIN ACCOUNT FOR?

PaN is an algorithm that operates on emergence. It is not by any means the first algorithm to do so, <sup>88,156</sup> but it is the first (as far as I am aware) that uses solely emergent principles to learn how to take actions in an environment. Here I discuss a few topics that I think such an algorithm could shed new light on, albeit in a highly speculative manner.

The topics of play and aesthetics, in particular, should not be seen as tangential to the function of living nervous systems. I think that they're essential to behavior in all but the tiniest animals, but that they are very hard to study and so haven't been in mainstream computational neuroscience research, to say the least. This is reasonable; why not focus on the things we know we can tackle? At the same time, it's a shame that these areas have gotten less attention just because they are less reliable or less quantifiable than other properties of animalian nervous systems.

### A.2.1 THE SPECIALIZATION OF BRAIN REGIONS

I can't tell if this claim is anything special. To some it might be obvious, but I want to say it anyway, because to others it might not be. If you squint at Chapters 3 to 5 in this thesis, you might end up with the same suspicion I have at the moment, which is that perhaps the allocation of different tasks to different brain regions is almost completely attributable to the inductive biases of the regions given their architectures and connectivities. That is, maybe the reason my hippocampus acts as a hippocampus, or my cerebellum acts as a cerebellum, is mostly because they're a slightly better shape for whatever they do than the other brain regions are.

The prediction here would be that if I just make a rough approximation of the shape of a vertebrate brain, giving it basal ganglia, cortex, etc., put the sensory inputs and motor outputs in the right places, and restrict a few connections between modules, then everything else will fall into place. The basal ganglia-shaped region will start to regulate motor control, the cortex-shaped region will

be responsible for higher-order reasoning, the hippocampus-shaped region will be responsible for memory consolidation.

There is a long way to go before I think we can test this prediction, but there are some medical reports and experiments suggesting it could pan out as I've written. Because if there is little that specializes a brain region other than its architecture (or perhaps broad axes of variation between regions; see the next section A.2.2), then when people are born missing weirdly big chunks of their neural architectures, you might expect the brain to be able to reallocate the necessary stuff to other parts—parts that might not be *as* good at the task as the missing chunk, but enough to perform the many complicated, coordinated behaviors to survive.

There are reports, for instance, of multiple people now who have been born without cerebella.<sup>122</sup> The human cerebellum is reported to have about 80% of the total neurons in the brain, and that people can survive without it is insane if this estimate is anywhere close to accurate.<sup>165</sup> People with cerebellar injury suffer much worse loss of function than people who are born without one, suggesting an important role in the initial allocation of tasks during growth and development. Patients without cerebella can have marked impairment of motor function, but only “mild to moderate neuropsychological impairments in IQ, planning behavior, visual, verbal and spatial memory, visuospatial perception and attention.”<sup>122</sup> And in one woman with cerebellar agenesis, she can lead “a useful though simple life, and is able to work in an electronics workshop.”<sup>122</sup>

A similar case arises in a more specialized structure: the human olfactory bulb. The absence of an olfactory bulb can come with anosmia, as one would expect, but it doesn't always. In 2020, Weiss et al. published a report of two women with normal odor processing and seemingly normal activity in the brain where olfactory bulbs usually project, but who were missing the olfactory bulbs themselves.<sup>200</sup>

Compare these results to what one might expect using a classical reinforcement learning perspective. In an actor-critic RL architecture,<sup>112</sup> one network is assigned to learn value estimates and

another is assigned to learn probability distributions of actions. You *cannot* separate them and maintain function. This is not a claim; this is how the agents are designed. Some neuroscientists argue that different regions of human or animal brains act as actors and other regions act as critics.<sup>5,103,129</sup> Under the RL paradigm, one would take these regions as fixed, incapable of computing anything but their evolutionarily assigned role. Not only would this present a problem in terms of evolutionary continuity (the structures would have undergone a discrete change in function at some point, which is unlikely but admittedly possible), but it is also not clear how brain regions might reallocate functions in the case of complete agenesis. The RL model also ignores the bigger problems of how rewards are represented, communicated, and change to begin with, except to assign these hard questions to the magic of evolution (see Section A.2.3).

The way I've argued it, the RL take is a bit of a straw man. In reality, the RL perspective has a different role to the emergence perspective: RL describes roughly what the nervous system *becomes*, while emergence explains how it might get there. If my hypotheses are correct, then the mistake only occurs when one attributes RL with both roles.

#### A.2.2 BROAD AXES OF VARIATION

In a way, the noise parameter sweeps in Figure 4.4i-j and 4.8b are pretty suggestive. They suggest that one can tune activity and weight noise and change big-picture behavior: the amount of exploration or the amount of reward fixation, only with couple dials. A related hypothesis/concept is stated in Chandra and Khona's 2024 preprint on development in the primate visual system (emphasis mine):

We identify a theoretical principle — local greedy wiring minimization via spontaneous drive (GWM-S) — implemented by the mechanism, and use this insight to propose biologically distinct growth rules that predict similar endpoints but testably



distinguishable developmental trajectories. The same rules predict how input geometry and cortical geometry together drive emergence of hierarchical, convolution-like, spatially and topographically organized sensory processing pathways for different modalities and species, providing a possible explanation for the observed pluripotency of cortical structure formation. **We find that the few parameters governing structure emergence in the growth rule constitute simple knobs for rich control.**

Similarly, in PaN, there are only a few parameters that have large, but not always direct effects on behavior. I don't think PaN is complete (Section A.1), but it already makes some interesting predictions about how nervous systems might vary between each other.

The Big Five Personality inventory in psychology<sup>43</sup> is one of the few psychological metrics that is claimed to be stable over time. However, this claim has been disputed, and some people may have more consistent characteristics than others.<sup>72,205</sup> At the same time, there are disorders that can be felt as part of a personality—Tourette syndrome, for instance,<sup>168\*</sup> or obsessive compulsive disorder.

What an emergent model like PaN may predict is that there are relatively few “knobs” in neural computation, and these knobs can tune a wide variety of traits. Krama et al.<sup>113</sup> showed that variability of behavior in fruit flies was tunable by serotonin signaling, which also represents a sparse control for a general behavior. In a similar vein, we show in Figures 4.4 and Section 4.6 that moving along the activity/weight noise plane changes the amount of reward-seeking, exploration, and exploitation in a PaN network. In the case of activity and weight noise, these parameters could be tuned by any number of genes—any gene related to the reliability of an ion channel or protein relevant to synaptic function, or the fidelity of action potential propagation could be part of the reason one organism might reside in a given place in the activity/weight noise plane.

---

\*In the essay *A Surgeon's Life* in the collection *An Anthropologist on Mars*, Oliver Sacks writes: “It is difficult for Bennett, and is often difficult for Touretters, to see their Tourette's as something external to themselves, because many of its tics and urges may be felt as intentional, as an integral part of the self, the personality, the will.”

Such a model could explain why some axes of variation in nervous systems are only identifiable through behavior; that is, they cannot be traced back to a small set of genes responsible. Instead, there tend to be puzzlingly large numbers of implicated genes for many disorders, even though the disorders are consistent enough to be diagnosed.<sup>†</sup>

In a more complete model (Section A.1), we would have a larger number of broad axes to tune than in PaN, which is mostly just activity and weight noise, and activity and weight learning rates. The axes in future models could include the relative strength of a Hebbian learning component, the excitatory/inhibitory neuron ratio, the ratio of recurrent to feedforward connections, etc. Any of these, or combinations of them, could have associated and consistent behavioral axes of variation.

### A.2.3 AESTHETICS

Sometime during my senior year of high school, the school newspaper did a student survey and asked everybody about their favorite classes. The winners were overwhelmingly in the fine arts department: theater, band, studio art, orchestra. The results were curious to me, as we were a public high school in a science-heavy town, but they were also fairly intuitive. We had some great and caring STEM teachers, but the way high school works in the US, you usually have to proactively search in order to find any semblance of fun or creativity in math or physics before you find it.

In this field it can be easy for me to forget what was quite obvious in high school: aesthetics are important. Usually in neuroscience this is explained away by the evolutionary advantage of caring about one's appearance or image, from the angle of attracting a partner; but how could that explain all of art, and how we respond to it? There are legions of artists out there who live for what they do, and some may do it in part to attract partners, but not all. Never always. In fact, becoming a

---

<sup>†</sup>Perhaps the most notable of this category is autism spectrum disorder, where the strongest genetic links can be traced back to just 10-20% of cases.<sup>68</sup> Autism spectrum disorder specifically might be related to noise itself—see the reference by Bhaskaran et al., titled *Endogenous noise of neocortical neurons correlates with atypical sensory response variability in the Fmr1-/- mouse model of autism*.<sup>20</sup> But I want to emphasize again that this section is *wildly* speculative.

professional artist or musician frequently comes with the *loss* of food security and ability to attract a partner. The experience of art can be one of the most rewarding parts of anybody's life; many of my friends work during the day but really live for their baking, or readings, or music, or pottery. Others have devoted their lives to artistic careers. How, then, can any theory or model of neuroscience be complete without accounting for the power of aesthetics?

In Darwin's day, an explanation for the diversity of species was pawned off to higher intelligence. Nowadays, I wonder whether we're following a similar line of reasoning in neuroscience for the existence of the diversity of natural behavior, except that we've replaced the role of God with evolution. What I mean is that, independent of how real God is and how real evolution is, an effect of saying "God created the shape of the species" or "evolution shaped the connections of the neural circuits" is to obstruct understanding more than explain how anything works, because you've swept a mysterious phenomenon into an also pretty mysterious black box. This was much more apparent in the case of God (but was it, in Darwin's time?), but I believe it is also true for the case of evolution. It's how people use and interpret the explanation of evolution that frustrates me, because I think these sorts of answers discourage the continued study of something for which we think we already know the solution.

I'm not saying evolution doesn't play a role in behavioral shaping. Of course it does. But it doesn't *really* explain the mechanics behind aesthetic preference. It relates to the cause but not the implementation. Evolution doesn't explain *how*, in my brain, today I want to write some self-indulgent prose that nobody but I will read, and tomorrow I want to mess around on my violin instead. It doesn't explain how, within a set of genetically identical twins, one might be really into written poetry and the other can't stand rhymes unless they come with a backing track. All I'm saying here is that I don't think art can be fully or satisfactorily explained by offloading it to evolution.

That is another reason why I believe that the emergent, autonomous brain<sup>159</sup> is the best theory/model we have right now in neuroscience: the appreciation and creation of art is a natural con-

sequence. Aesthetics is not something you program or train in if you decide you want it, like in other machine learning models. It is not a side effect of other, more important preferences that support survival/reproduction. In the emergent brain, the aesthetic sense is inherent—*it's a feature, not a bug*.

#### “REWARD” IS A CONSTRUCT IN THE EMERGENT BRAIN

Consider Figure 5.3, which shows that small PaN networks can form preferences for different tasks. The way PaN works is by doing anything it can to maximize a signal-to-noise ratio inside itself. It goes after stronger sensory inputs because those give it more signal. When it has the option between two actions that give the same strength signal to two different sensory inputs, it chooses between them evenly.

But now imagine a slightly different scenario (there are details that may have to be tweaked in PaN to get this to work, but here's the general concept). Instead of two discrete actions that provide two completely separate inputs, say the network can choose to pick one or both of the actions at every timestep. The sensory neurons are connected to different subsets of the network, so PaN gets the most signal when it chooses both actions rather than one at a time.

Now imagine that this sensory layer is not a sensory layer at all—it is in fact a hidden layer preceded by many other hidden layers, and all of them much larger than two neurons. First, what does each neuron's activation correspond to? Each one corresponds to some combination of inputs from the previous layer, which depends on the network's connectivity. Eventually that means that the neuron activates with a combination of sensory inputs, after the combination has gone through some sequential processing, and maybe you've even found a neuron that is a hierarchical representation of some concept. Let's pretend that you have found such a neuron, and, in the spirit of classical neuroscience, it strongly activates when you think about Jennifer Aniston.

In PaN, the activation itself is part of the “reward”—that's one way to think about it. If you had

a PaN model with a very large number of Jennifer Aniston cells in its hidden layers, you would expect that it would really like to take actions that allow it to keep seeing, hearing, or thinking about Jennifer Aniston. Maybe these Jennifer Aniston cells formed because you watched too much Friends, and in that case, you might also have a very large number of Matt LeBlanc cells. And if you do, then the most rewarding action you can take would be to find a way to see and hear both Jennifer Aniston and Matt LeBlanc at once, which is probably to watch more Friends.

Now you have a new effective reward function: television that has both Jennifer Aniston and Matt LeBlanc. In PaN, this new reward has emerged from experience. You weren't born with it. And as it emerges, it lives in a sea of simultaneously developing and decaying reward functions in other parts of your brain—tomorrow, perhaps, you're done with Friends and find yourself really intrigued by New Girl instead.

So far I've only discussed why combinations of rewards can be stronger than the individual components. How does this relate to an aesthetic sense?

#### A THEORY OF EASY AND DIFFICULT BEAUTY

My English professor Louisa Thomas would frequently tell us that good writing “should make you feel.” She gave us an essay by the writer Chloé Cooper Jones, which suggested that there were different *ways* that art or beauty could make you feel. In Cooper Jones' memoir *Easy Beauty*, she marks an axis for the experience of beauty; on one side is easily accessible pleasure and on the other complex, layered, and “difficult beauty.” Easy beauty is available whenever one wants it—beach reads, pop music, Beyoncé concerts, she suggests. And because of the availability of easy beauty, and its lack of depth, critics sometimes look down on it as a shallow kind of art.

Difficult beauty, on the other hand, comes with time and with study. It's the appreciation of James Joyce or Emmy Noether, neither of which I imagine comes quickly to most people, or of a Shostakovich symphony: these things can be rewarding in isolation, but they only reveal their full

depth once one has developed an understanding of the pieces that built them—the particular keys<sup>‡</sup>, intervals, harmonies—plus their history and context altogether. Listening to Shostakovich No. 7 as a child was one thing, but listening to it after learning what Shostakovich had intended for the piece, the global politics involved, and that the audience had sat in the concert hall weeping during the premiere because they knew what he had meant it for—these were not the same experiences.

In the emergent model of the brain, reward can be immediate, strong, and sparse. This could be like the pleasure of sugar. Since it depends on only a few features, neurons adapt and this fast pleasure, this easy beauty, can quickly fade. But reward can also come from the activation of many components at once, in some combination that drives strong signal through the network by virtue of numbers and density rather than the sheer magnitude of any one part. In a more layered beauty, with a larger number of components that comprise it, these components can continually shift and be replaced while maintaining the idea of the whole. Under this model, it's more work to reach a sense of difficult beauty, because you have to learn a lot of the components that allow you to appreciate it, but this also makes it hard to acclimatize, and there is a sense of depth and robustness to it.

I hold with Cooper Jones and those that push back against the idea that one form of art is better than the other; both easy and difficult beauty have their place. But regardless of the value of each kind of art or aesthetic experience, I'm writing about this topic to point out that people can generally agree that some experiences are faster to please, and these are usually the ones most connected to strong sensory stimuli. Other experiences, in contrast, take contemplation and learning to appreciate. What fascinates me about the emergent nervous system model, where reward is really the continued propagation of signal over noise, is that these understandings we have developed of aesthetics, whether through daily human experience or through lengthy study and professional

---

<sup>‡</sup>I think if you ask any musician, they will have different emotional associations with different key signatures. For me, classically trained in violin and piano, a piece in E $\flat$  is subtler, cooler than a piece in D, which open and friendly but a little plain. A jazz musician might give you a different answer.

criticism, almost fall out for free.

#### A.2.4 PLAY

After I became disillusioned with the reinforcement learning project, I spent over a year doing almost nothing. I greatly appreciate the patience my advisors had with me during this time. I tried and failed to learn the trumpet. I travelled across Europe using trains and hostels like the other students trying to find themselves on their gap years. I tried and failed to run a marathon, too, and the training I underwent took me along a lot of rivers where I could watch ducks, geese, and swans for hours at a time. Mostly, I spent a lot of time thinking it might be nice to be a goose.

Geese lived in herds, which may not be the technically correct word for them, but it feels right. These herds roamed the riversides grazing away entire days. The geese lived in their food and yet tore at it maniacally; when they got tired they would stand on one foot, or bury their heads in the feathers of their backs, which doubled as pillows upon which they could tuck down their soft white eyelids.

A herd might step into the water for a swim. They'd float along aimlessly in a loose string of birds. Sometimes I'd catch a few geese standing by a riverfront, just outside the water, their beaks pointed in the same direction at somewhere slightly above the horizon. I would crouch and try and get in their heads, to find out what they were looking at, or maybe waiting for.

While watching the geese I could always contrive some evolutionary purpose for their actions. They hissed at dogs and children to protect themselves. They trawled through rivers to diversify their diets from grasses and dandelion flowers. Every story I invented could explain why the geese had evolved these behaviors, but I felt, with time, that nothing was as deeply true as saying that they were just doing what felt right to them in the moment. I wished I could spend my days like the geese did, too, except that the graduate students I knew who abided by that principle tended to spend a lot of their time smoking weed. While I wasn't opposed to that, entirely, I must admit I wanted a

more imaginative solution. So for those long months, I thought vaguely about what I wanted to study next.

Of all classifications of animal behavior, the one I find most interesting is play. Play seemed to embody everything I saw in the birds: goals that were transient, but as intense a drive as any other; play was behavior for behaviors' sake. But I found it impossible to communicate play as a goal for my algorithm, to put it in the shape of an equation to my advisors. "No behavioral concept has proven more ill-defined, elusive, controversial, and even unfashionable," wrote the biologist E.O. Wilson on the subject.<sup>202</sup>

And yet play reverberates throughout the animal kingdom. An overwhelming amount of science is done simply because scientists like to play with their ideas. The existence of vertebrate play is almost inarguable, but cases are being built—have been built, for centuries—across the invertebrates, too.<sup>48</sup> Charles Darwin himself contributed to the invertebrate case: "Even insects play together," he wrote, "as has been described by that excellent observer, P. Huber, who saw ants chasing and pretending to bite each other, like so many puppies."<sup>42</sup> Notes from another ant observer, Aguste Forel: "It is a well-established fact, therefore, that on fine, calm days when they are feeling no hunger or any other cause for anxiety, certain ants entertain themselves with sham fights, without doing each other any harm... This is one of their most amusing habits."<sup>30</sup>

Wrestling is a relatively controversial form of play. Some argue that it is obviously useful for future survival, so cannot be considered "true play." Here, then, is a less controversial case recorded by the naturalist Henry Walter Bates, while he was studying army ants along the Amazon (*The Naturalist on the River Amazons*, 1863<sup>14</sup>).

The life of these [ants] is not all work, for I frequently saw them very leisurely employed in a way that looked like recreation. When this happened, the place was always a sunny nook in the forest... The main column of the army and the branch columns...were in their ordinary relative positions; but, instead of pressing forward



eagerly, and plundering right and left, they seemed to have been all smitten with a sudden fit of laziness.

Some were walking slowly about, others were brushing their antennae with their forefeet; but the drollest sight was their cleaning one another. The actions of these ants looked like simple indulgence in idle amusement. Have these little creatures, then, an excess of energy...and do they expend it in mere sportiveness, like young lambs or kittens or in idle whims like rational beings?

It is probable that these hours of relaxation and cleaning may be indispensable to the effective performance of their harder labours, but whilst looking at them, the conclusion that the ants were engaged merely in play was irresistible.

I write all this because play, like aesthetics, seems to be one of the things that make life worth living. I think the two are intricately related. Under the emergent model of the brain, just like the emergence of aesthetic reward, play might be based on an invented, combinatorial, goal-directed system of pleasure, except possibly with greater emphasis on a predicted outcome than in artistic pursuits.

Mechanistically, how would play work in a PaN-like model? In play, there is usually a predicted, imagined goal. The fulfillment of that goal through the correct sequence of motor actions in a model like PaN would send signal through the network, and thus motivate future actions to continue fulfilling that goal. These goals can be fixated upon, like we show in our model, but they can also change depending on the environment or an internal state. In the most addictive games or play environments, a positive feedback loop occurs to keep the entity hooked on the game. Prediction leads to action leads to fulfillment leads to the activation of the same neurons that gave the prediction, and the loop repeats. In such a model this loop is continually reinforced, until noise or adaptation (Figure 4.6d might be relevant) kicks it out of the loop, and the network decides to go do

something else.

In both animals and PaN, there is clear malleability of the goals that the entity tries to achieve. Your environment can give you new goals, evidenced by the fact that the scoreboard of a basketball game shouldn't mean anything in an evolutionary context. Goals in the emergent brain are malleable by nature, and most video games and sports take advantage of this malleability. The invented goals in play are often very clear, and forms of play with clear goals are the most engaging, perhaps because they best leverage the prediction-action-fulfillment loop. The goals an animal pursues during play can be invented on one's own, like a child trying to throw a paper ball into a wastebasket, or they can be suggested by an external entity, like in board games or organized sports. Yet even when goals are externally presented, people like to invent their own goals within the external ones. Take, for example, the writer David Epstein's experience watching Roger Federer at the U.S. Open.<sup>52</sup>

Back when I was at [*Sports Illustrated*], I once saw [Federer] warming up at the U.S. Open. Most of the athletes I'd seen were stretching, or rallying, or practicing powerful serves. Federer appeared to have stationed a ballboy across the court with his hand outstretched, and he seemed to be trying to hit soft bouncers right into the boy's hand without him having to move it. Basically, it looked like something a kid would enjoy.

Perhaps this lack of goal malleability in modern AI is behind our suspicion that it does not quite work the way that we do. Like I said in Section 2.6, the dominant training method of artificial neural networks, backpropagation, requires one to set a goal at the very start, and the network itself cannot change it. To say that evolution *entirely* explains the setting of such a goal—that play behavior exists because it improves learning and survival, to separate it into distinct reward functions for social, tactile, or motor play, etc.—is, I think, to forget how overwhelming the experience can be, and that in the moment, our motivations for play are really just to play.

### A.2.5 DISCUSSION

Some of us are lucky enough that the evolutionary necessities of food and physical security are not all-consuming concerns. When this is true, isn't it interesting that what feels most meaningful to people are social connections, a sense of play, and the appreciation or creation of art? I'm sure a lot of what I've written in this appendix is incorrect and/or very naive. But I want to emphasize that right now, there is very little idea of how play or aesthetics might work at all in artificial neural networks. If I were studying backprop-trained ANNs as a model for the brain and I wanted to understand anything about play or aesthetics, I'd have no idea where to start, and I'd have little confirmation that anything I studied in the ANN should correspond to how the brain works.

At the very least, PaN and the self-supervised behavior framework represent a new direction from which to tackle these questions. I think the emergent model of nervous systems is capable of explaining much more than current models can, like how goals in play might originate, or how aesthetic preferences form or change over time, and why creativity drives so much of human (if not also other animal) behavior; because in PaN, and in any other emergent SSB model, any reward-seeking behavior is driven by the same principles that play and aesthetic preference might derive from, and the consequence is that fun and pleasure and beauty might motivate an animal just as much (or even more!) than the purely utilitarian. In a lot of cases, too, contest and beauty elevate the utilitarian far beyond what is necessary for survival and reproduction, evidenced by all the wasted effort we put into athletic competition, into games as a pastime, into the experiences of good food and good sex. And I think the importance of art and play, the centrality of it in human experience, should be intuitive to anybody who is alive, anybody who has a favorite sport or team or song or smell or painting or poem, just like it was intuitive to me and my friends back in high school.





# References

- [noa] Reinforcement Learning Resources — Stable Baselines 2.10.2 documentation.
- [2] Abel, D., Barreto, A., Van Roy, B., Precup, D., van Hasselt, H. P., & Singh, S. (2024). A definition of continual reinforcement learning. *Advances in Neural Information Processing Systems*, 36.
- [3] Afraz, S.-R., Kiani, R., & Esteky, H. (2006). Microstimulation of inferotemporal cortex influences face categorization. *Nature*, 442(7103), 692–695. Number: 7103 Publisher: Nature Publishing Group.
- [4] Andersen, R. A., Aflalo, T., Bashford, L., Bjånes, D., & Kellis, S. (2022). Exploring Cognition with Brain–Machine Interfaces. *Annual Review of Psychology*, 73(1), 131–158. \_eprint: <https://doi.org/10.1146/annurev-psych-030221-030214>.
- [5] Araújo, A., Duarte, I. C., Sousa, T., Oliveira, J., Pereira, A. T., Macedo, A., & Castelo-Branco, M. (2024). Neural inhibition as implemented by an actor-critic model involves the human dorsal striatum and ventral tegmental area. *Scientific Reports*, 14(1), 6363.
- [6] Arcaro, M. J., Ponce, C., & Livingstone, M. (2020). The neurons that mistook a hat for a face. *Elife*, 9, e53798.
- [7] Arcaro, M. J., Schade, P. F., Vincent, J. L., Ponce, C. R., & Livingstone, M. S. (2017). Seeing faces is necessary for face-domain formation. *Nature neuroscience*, 20(10), 1404–1412.
- [8] Ardiel, E. L. & Rankin, C. H. (2010). An elegant mind: Learning and memory in *Caenorhabditis elegans*. *Learning & Memory*, 17(4), 191–201. Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab.
- [9] Ariyanto, M., Refat, C. M. M., Hirao, K., & Morishima, K. (2023). Movement Optimization for a Cyborg Cockroach in a Bounded Space Incorporating Machine Learning. *Cyborg and Bionic Systems*, 4, 0012. Publisher: American Association for the Advancement of Science.

- [10] Ba, J., Hinton, G. E., Mnih, V., Leibo, J. Z., & Ionescu, C. (2016). Using fast weights to attend to the recent past. *Advances in neural information processing systems*, 29.
- [11] Barabási, D. L., Schuhknecht, G. F., & Engert, F. (2024). Functional neuronal circuits emerge in the absence of developmental activity. *Nature Communications*, 15(1), 364.
- [12] Barbulescu, R., Mestre, G., Oliveira, A. L., & Silveira, L. M. (2023). Learning the dynamics of realistic models of c. elegans nervous system with recurrent neural networks. *Scientific Reports*, 13(1), 467.
- [13] Barra, A., Beccaria, M., & Fachechi, A. (2018). A new mechanical approach to handle generalized hopfield neural networks. *Neural Networks*, 106, 205–222.
- [14] Bates, H. W. (1863). *The naturalist on the River Amazons*. Murray.
- [15] Bayona, N. A., Bitensky, J., & Teasell, R. (2005). Plasticity and reorganization of the uninjured brain. *Topics in stroke rehabilitation*, 12(3), 1–10.
- [16] Bengio, Y., Lahlou, S., Deleu, T., Hu, E. J., Tiwari, M., & Bengio, E. (2023). Gflownet foundations. *Journal of Machine Learning Research*, 24(210), 1–55.
- [17] Bergmann, E., Gofman, X., Kavushansky, A., & Kahn, I. (2020). Individual variability in functional connectivity architecture of the mouse brain. *Communications Biology*, 3(1), 1–10. Number: 1 Publisher: Nature Publishing Group.
- [18] Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., et al. (2023). Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3), 8.
- [19] Bhardwaj, A., Thapliyal, S., Dahiya, Y., & Babu, K. (2018). FLP-18 Functions through the G-Protein-Coupled Receptors NPR-1 and NPR-4 to Modulate Reversal Length in *Caenorhabditis elegans*. *Journal of Neuroscience*, 38(20), 4641–4654. Publisher: Society for Neuroscience Section: Research Articles.
- [20] Bhaskaran, A. A., Gauvrit, T., Vyas, Y., Bony, G., Ginger, M., & Frick, A. (2023). Endogenous noise of neocortical neurons correlates with atypical sensory response variability in the *fmr1-y* mouse model of autism. *Nature Communications*, 14(1), 7905.
- [21] Biehl, M., Pollock, F. A., & Kanai, R. (2021). A technical critique of some parts of the free energy principle. *Entropy*, 23(3), 293.
- [22] Bogacz, R. (2017). A tutorial on the free-energy framework for modelling perception and learning. *Journal of mathematical psychology*, 76, 198–211.

- [23] Bonizzato, M. & Martinez, M. (2021). An intracortical neuroprosthesis immediately alleviates walking deficits and improves recovery of leg control after spinal cord injury. *Science Translational Medicine*, 13(586), eabb4422. Publisher: American Association for the Advancement of Science.
- [24] Bono, S., Madan, S., Grover, I., Yasueda, M., Breazeal, C., Pfister, H., & Kreiman, G. (2024). Look around! unexpected gains from training on environments in the vicinity of the target. *arXiv preprint arXiv:2401.15856*.
- [25] Bostrom, N. & Sandberg, A. (2009). Cognitive Enhancement: Methods, Ethics, Regulatory Challenges. *Science and Engineering Ethics*, 15(3), 311–341.
- [26] Brandt, R., Gergou, A., Wacker, I., Fath, T., & Hutter, H. (2009). A *Caenorhabditis elegans* model of tau hyperphosphorylation: Induction of developmental defects by transgenic overexpression of Alzheimer’s disease-like modified tau. *Neurobiology of Aging*, 30(1), 22–33.
- [27] Brenner, S. (1974). The genetics of *caenorhabditis elegans*. *Genetics*, 77(1), 71–94.
- [28] Briscoe, G. & Dini, P. (2010). Towards autopoietic computing. In *Digital Ecosystems: Third International Conference, OPAALS 2010, Aracaju, Sergipe, Brazil, March 22-23, 2010, Revised Selected Papers 3* (pp. 199–212).: Springer.
- [29] Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., & Ramesh, A. (2024). Video generation models as world simulators.
- [30] Burghardt, G. M. (2005). *The genesis of animal play: Testing the limits*. MIT press.
- [31] Buzsáki, G. (2019). *The Brain from Inside Out*. Oxford University Press.
- [32] Byrnes, S. (2023). Why i’m not into the free energy principle.
- [33] Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018). Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)* (pp. 67–74).: IEEE.
- [34] Chittka, L. (2022). *The mind of a bee*. Princeton University Press.
- [35] Christodoulou, P. (2019). Soft actor-critic for discrete action settings. *arXiv preprint arXiv:1910.07207*.
- [36] Churchland, M. M., Yu, B. M., Cunningham, J. P., Sugrue, L. P., Cohen, M. R., Corrado, G. S., Newsome, W. T., Clark, A. M., Hosseini, P., Scott, B. B., et al. (2010). Stimulus onset quenches neural variability: a widespread cortical phenomenon. *Nature neuroscience*, 13(3), 369–378.



- [37] Clark, D. G. & Abbott, L. F. (2024). Theory of coupled neuronal-synaptic dynamics. *Phys. Rev. X*, 14, 021001.
- [38] Clopath, C. (2012). Synaptic consolidation: an approach to long-term learning. *Cognitive neurodynamics*, 6(3), 251–257.
- [39] Coddington, L. T., Lindo, S. E., & Dudman, J. T. (2023). Mesolimbic dopamine adapts the rate of learning from action. *Nature*, 614(7947), 294–302.
- [40] Cook, S. J., Jarrell, T. A., Brittin, C. A., Wang, Y., Bloniarz, A. E., Yakovlev, M. A., Nguyen, K. C., Tang, L. T.-H., Bayer, E. A., Duerr, J. S., et al. (2019). Whole-animal connectomes of both *Caenorhabditis elegans* sexes. *Nature*, 571(7763), 63–71.
- [41] Dale, H. (1935). Pharmacology and nerve-endings.
- [42] Darwin, C. (1877). A biographical sketch of an infant. *Mind*, 2(7), 285–294.
- [43] De Raad, B. (2000). *The big five personality factors: the psycholexical approach to personality*. Hogrefe & Huber Publishers.
- [44] Degraeve, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de las Casas, D., Donner, C., Fritz, L., Galperti, C., Huber, A., Keeling, J., Tsimpoukelli, M., Kay, J., Merle, A., Moret, J.-M., Noury, S., Pesamosca, F., Pfau, D., Sauter, O., Sommariva, C., Coda, S., Duval, B., Fasoli, A., Kohli, P., Kavukcuoglu, K., Hassabis, D., & Riedmiller, M. (2022). Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897), 414–419.
- [45] Destexhe, A. & Rudolph-Lilith, M. (2012). *Neuronal noise*, volume 8. Springer Science & Business Media.
- [46] Deza, A., Liao, Q., Banburski, A., & Poggio, T. (2020). Hierarchically compositional tasks and deep convolutional networks. *arXiv preprint arXiv:2006.13915*.
- [47] Dobs, K., Martinez, J., Kell, A. J., & Kanwisher, N. (2021). Brain-like functional specialization emerges spontaneously in deep neural networks. *bioRxiv*.
- [48] Dona, H. S. G., Solvi, C., Kowalewska, A., Mäkelä, K., MaBouDi, H., & Chittka, L. (2022). Do bumble bees play? *Animal Behaviour*, 194, 239–251.
- [49] Dong, X., Kheiri, S., Lu, Y., Xu, Z., Zhen, M., & Liu, X. (2021). Toward a living soft micro-robot through optogenetic locomotion control of *Caenorhabditis elegans*. *Science Robotics*, 6(55).
- [50] Donnelly, J. L., Clark, C. M., Leifer, A. M., Pirri, J. K., Haburcak, M., Francis, M. M., Samuel, A. D. T., & Alkema, M. J. (2013). Monoaminergic Orchestration of Motor Programs in a Complex *C. elegans* Behavior. *PLOS Biology*, 11(4).

- [51] Enriquez-Geppert, S., Huster, R. J., & Herrmann, C. S. (2013). Boosting brain functions: Improving executive functions with behavioral training, neurostimulation, and neurofeedback. *International Journal of Psychophysiology*, 88(1), 1–16.
- [52] Epstein, D. (2022). Here’s Why Federer’s Developmental Story Should Be As Famous As He Is.
- [53] Erickson, J. C., Herrera, M., Bustamante, M., Shingiro, A., & Bowen, T. (2015). Effective Stimulus Parameters for Directed Locomotion in Madagascar Hissing Cockroach Biobot. *PLOS ONE*, 10(8), e0134348. Publisher: Public Library of Science.
- [54] Escobar, A., Kim, S., Primack, A. S., Duret, G., Juliano, C. E., & Robinson, J. T. (2023). Relationship between neural activity and neuronal cell fate in regenerating hydra revealed by cell-type specific imaging. *bioRxiv*, (pp. 2023–03).
- [55] Fagundes, N. J., Bisso-Machado, R., Figueiredo, P. I., Varal, M., & Zani, A. L. (2022). What we talk about when we talk about “junk dna”. *Genome Biology and Evolution*, 14(5), evaco55.
- [56] Faisal, A. A., Selen, L. P., & Wolpert, D. M. (2008). Noise in the nervous system. *Nature reviews neuroscience*, 9(4), 292–303.
- [57] Fang, C. & Stachenfeld, K. L. (2023). Predictive auxiliary objectives in deep rl mimic learning in the brain. *arXiv preprint arXiv:2310.06089*.
- [58] Fedorenko, E., Duncan, J., & Kanwisher, N. (2013). Broad domain generality in focal regions of frontal and parietal cortex. *Proceedings of the National Academy of Sciences*, 110(41), 16616–16621.
- [59] Forel, A. (1928). *The Social World of the Ants Compared with that of Man: By Auguste Forel...*, volume 2. GP Putnam’s sons.
- [60] Fountas, Z., Sajid, N., Mediano, P., & Friston, K. (2020). Deep active inference agents using monte-carlo methods. *Advances in neural information processing systems*, 33, 11662–11675.
- [61] Friston, K. (2005). A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological sciences*, 360(1456), 815–836.
- [62] Friston, K. (2019). A free energy principle for a particular physics. arXiv:1906.10184 [q-bio].
- [63] Friston, K., Da Costa, L., Sajid, N., Heins, C., Ueltzhöffer, K., Pavliotis, G. A., & Parr, T. (2023). The free energy principle made simpler but not too simple. *Physics Reports*, 1024, 1–29.
- [64] Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., Pezzulo, G., et al. (2016). Active inference and learning. *Neuroscience & Biobehavioral Reviews*, 68, 862–879.

- [65] Garner, K. & Dux, P. E. (2015). Training conquers multitasking costs by dividing task representations in the frontoparietal-subcortical system. *Proceedings of the National Academy of Sciences*, 112(46), 14372–14377.
- [66] Gauthier, I., Tarr, M. J., Anderson, A. W., Skudlarski, P., & Gore, J. C. (1999). Activation of the middle fusiform 'face area' increases with expertise in recognizing novel objects. *Nature neuroscience*, 2(6), 568–573.
- [67] Gazzaniga, M. S. (2015). *Tales from both sides of the brain: A life in neuroscience*. Ecco/HarperCollins Publishers.
- [68] Geschwind, D. H. (2011). Genetics of autism spectrum disorders. *Trends in cognitive sciences*, 15(9), 409–416.
- [69] Giurfa, M., Marcout, C., Hilpert, P., Thevenot, C., & Rugani, R. (2022). An insect brain organizes numbers on a left-to-right mental number line. *Proceedings of the National Academy of Sciences*, 119(44), e2203584119.
- [70] Groth, O., Wulfmeier, M., Vezzani, G., Dasagi, V., Hertweck, T., Hafner, R., Heess, N., & Riedmiller, M. (2021). Is curiosity all you need? on the utility of emergent behaviours from curious exploration. *arXiv preprint arXiv:2109.08603*.
- [71] Guo, D., Perc, M., Liu, T., & Yao, D. (2018). Functional importance of noise in neuronal information processing. *Europhysics Letters*, 124(5), 50001. Publisher: EDP Sciences, IOP Publishing and Società Italiana di Fisica.
- [72] Gurven, M., Von Rueden, C., Massenkoff, M., Kaplan, H., & Lero Vie, M. (2013). How universal is the big five? testing the five-factor model of personality variation among forager-farmers in the bolivian amazon. *Journal of personality and social psychology*, 104(2), 354.
- [73] Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., & Abbeel, P. (2018). Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.
- [74] Halvagal, M. S. & Zenke, F. (2023). The combination of hebbian and predictive plasticity learns invariant object representations in deep sensory networks. *Nature Neuroscience*, 26(11), 1906–1915.
- [75] Hamblin, C. & Alvarez, G. (2021). Viscnn: A tool for visualizing interpretable subgraphs in cnns. *Journal of Vision*, 21(9), 2674–2674.
- [76] Hanson, A. (2023). On being a hydra with, and without, a nervous system: what do neurons add? *Animal cognition*, 26(6), 1799–1816.

- [77] Haspel, G., Boyden, E. S., Brown, J., Church, G., Cohen, N., Fang-Yen, C., Flavell, S., Goodman, M. B., Hart, A. C., Hobert, O., et al. (2023). To reverse engineer an entire nervous system. *arXiv preprint arXiv:2308.06578*.
- [78] Haydari, A. & Yilmaz, Y. (2022). Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(1), 11–32. Conference Name: IEEE Transactions on Intelligent Transportation Systems.
- [79] Hebart, M. N., Dickter, A. H., Kidder, A., Kwok, W. Y., Corriveau, A., Van Wicklin, C., & Baker, C. I. (2019). Things: A database of 1,854 object concepts and more than 26,000 naturalistic object images. *PloS one*, 14(10), e0223792.
- [80] Helmholtz, H. v. (1948). Concerning the perceptions in general, 1867.
- [81] Hernandez-Nunez, L., Belina, J., Klein, M., Si, G., Claus, L., Carlson, J. R., & Samuel, A. D. (2015). Reverse-correlation analysis of navigation dynamics in *Drosophila* larva using optogenetics. *eLife*, 4, e06225. Publisher: eLife Sciences Publications, Ltd.
- [82] Hesse, J. K. & Tsao, D. Y. (2020). The macaque face patch system: a turtle’s underbelly for the brain. *Nature Reviews Neuroscience*, 21(12), 695–716.
- [83] Hinterwirth, A. J., Medina, B., Lockey, J., Otten, D., Voldman, J., Lang, J. H., Hildebrand, J. G., & Daniel, T. L. (2012). Wireless Stimulation of Antennal Muscles in Freely Flying Hawkmoths Leads to Flight Path Changes. *PLOS ONE*, 7(12), e52725. Publisher: Public Library of Science.
- [84] Hinton, G. E. & Plaut, D. C. (1987). Using fast weights to deblur old memories. In *Proceedings of the ninth annual conference of the Cognitive Science Society* (pp. 177–186).
- [85] Hollenstein, J., Auddy, S., Saveriano, M., Renaudo, E., & Piater, J. (2022). Action Noise in Off-Policy Deep Reinforcement Learning: Impact on Exploration and Performance. *Transactions on Machine Learning Research*.
- [86] Holzer, R. & Shimoyama, I. (1997). Locomotion control of a bio-robotic system via electric stimulation. In *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97*, volume 3 (pp. 1514–1519 vol.3).
- [87] Hommel, B., Chapman, C. S., Cisek, P., Neyedli, H. F., Song, J.-H., & Welsh, T. N. (2019). No one knows what attention is. *Attention, Perception, & Psychophysics*, 81, 2288–2303.
- [88] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8), 2554–2558.
- [89] Huang, Y. & Rao, R. P. (2011). Predictive coding. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2(5), 580–593.

- [90] Husson, S. J., Gottschalk, A., & Leifer, A. M. (2013). Optogenetic manipulation of neural activity in *C. elegans*: from synapse to circuits and behaviour. *Biology of the Cell*, 105(6), 235–250.
- [91] Ibarz, J., Tan, J., Finn, C., Kalakrishnan, M., Pastor, P., & Levine, S. (2021). How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40(4-5), 698–721. Publisher: SAGE Publications Ltd STM.
- [92] Isomura, T., Shimazaki, H., & Friston, K. J. (2022). Canonical neural networks perform active inference. *Communications Biology*, 5(1), 55.
- [93] Iturrate, I., Pereira, M., & Millán, J. d. R. (2018). Closed-loop electrical neurostimulation: Challenges and opportunities. *Current Opinion in Biomedical Engineering*, 8, 28–37.
- [94] Jaeger, J. (2023). Artificial intelligence is algorithmic mimicry: why artificial” agents” are not (and won’t be) proper agents. *arXiv preprint arXiv:2307.07515*.
- [95] James, W. (2007). *The principles of psychology*, volume 1. Cosimo, Inc.
- [96] Jospin, M., Qi, Y., Stawicki, T., Boulin, T., Schuske, K., Horvitz, H., Bessereau, J.-L., Jorgensen, E., & Jin, Y. (2009). A Neuronal Acetylcholine Receptor Regulates the Balance of Muscle Excitation and Inhibition in *Caenorhabditis elegans*. *PLoS biology*, 7, e1000265.
- [97] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873), 583–589.
- [98] Kanaya, H. J., Park, S., Kim, J.-h., Kusumi, J., Krenenou, S., Sawatari, E., Sato, A., Lee, J., Bang, H., Kobayakawa, Y., et al. (2020). A sleep-like state in hydra unravels conserved sleep mechanisms during the evolutionary development of the central nervous system. *Science advances*, 6(41), eabb9415.
- [99] Kanwisher, N. (2010). Functional specificity in the human brain: a window into the functional architecture of the mind. *Proceedings of the national academy of sciences*, 107(25), 11163–11170.
- [100] Kanwisher, N., McDermott, J., & Chun, M. M. (1997). The fusiform face area: a module in human extrastriate cortex specialized for face perception. *Journal of neuroscience*, 17(11), 4302–4311.
- [101] Kanwisher, N. & Yovel, G. (2006). The fusiform face area: a cortical region specialized for the perception of faces. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 361(1476), 2109–2128.
- [102] Kashin, S. M., Feldman, A. G., & Orlovsky, G. N. (1974). Locomotion of fish evoked by electrical stimulation of the brain. *Brain Research*, 82(1), 41–47.

- [103] Kearney, M. G. (2020). *An actor-critic circuit in the songbird enables vocal learning*. PhD thesis, Duke University.
- [104] Kell, A. J., Yamins, D. L., Shook, E. N., Norman-Haignere, S. V., & McDermott, J. H. (2018). A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3), 630–644.
- [105] Khona, M. & Fiete, I. R. (2022). Attractor and integrator networks in the brain. *Nature Reviews Neuroscience*, 23(12), 744–766.
- [106] Kim, K. & Li, C. (2004). Expression and regulation of an FMRFamide-related neuropeptide gene family in *Caenorhabditis elegans*. *Journal of Comparative Neurology*, 475(4), 540–550.   
\_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cne.20189>.
- [107] Kim, S. & Robinson, J. T. (2023). Phototaxis is a state-dependent behavioral sequence in *hydra vulgaris*. *bioRxiv*, (pp. 2023–05).
- [108] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13), 3521–3526.
- [109] Klinzing, J. G., Niethard, N., & Born, J. (2019). Mechanisms of systems memory consolidation during sleep. *Nature neuroscience*, 22(10), 1598–1610.
- [110] Kocabas, A., Shen, C.-H., Guo, Z. V., & Ramanathan, S. (2012). Controlling interneuron activity in *Caenorhabditis elegans* to evoke chemotactic behaviour. *Nature*, 490(7419), 273–277. ISBN: 1476-4687 Publisher: Nature Publishing Group.
- [111] Koch, C. (2004). *Biophysics of computation: information processing in single neurons*. Oxford university press.
- [112] Konda, V. & Tsitsiklis, J. (1999). Actor-critic algorithms. *Advances in neural information processing systems*, 12.
- [113] Krama, T., Munkevics, M., Krams, R., Grigorjeva, T., Trakimas, G., Jöers, P., Popovs, S., Zants, K., Elferts, D., Rantala, M. J., et al. (2023). Development under predation risk increases serotonin-signaling, variability of turning behavior and survival in adult fruit flies *drosophila melanogaster*. *Frontiers in behavioral neuroscience*, 17, 1189301.
- [114] Krauzlis, R. J., Wang, L., Yu, G., & Katz, L. N. (2023). What is attention? *Wiley Interdisciplinary Reviews: Cognitive Science*, 14(1), e1570.
- [115] Kriegeskorte, N. (2015). Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annual review of vision science*, 1, 417–446.

- [116] Krotov, D. (2023). A new frontier for hopfield networks. *Nature Reviews Physics*, 5(7), 366–367.
- [117] Lafer-Sousa, R., Wang, K., Azadi, R., Lopez, E., Bohn, S., & Afraz, A. (2023). Behavioral detectability of optogenetic stimulation of inferior temporal cortex varies with the size of concurrently viewed objects. *Current Research in Neurobiology*, 4, 100063.
- [118] LeCun, Y., Touresky, D., Hinton, G., & Sejnowski, T. (1988). A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1 (pp. 21–28).
- [119] LeDoux, J. (2010). *The cognitive neuroscience of mind: a tribute to Michael S. Gazzaniga*. MIT Press.
- [120] Lee, J. B., Yonar, A., Hallacy, T., Shen, C.-H., Milloz, J., Srinivasan, J., Kocabas, A., & Ramanathan, S. (2019). A Compressed Sensing Framework for Efficient Dissection of Neural Circuits. *Nature methods*, 16(1), 126–133.
- [121] Leifer, A. M., Fang-Yen, C., Gershow, M., Alkema, M. J., & Samuel, A. D. T. (2011). Optogenetic manipulation of neural activity in freely moving *Caenorhabditis elegans*. *Nature Methods*, 8(2), 147–152. Number: 2 Publisher: Nature Publishing Group.
- [122] Lemon, R. & Edgley, S. (2010). Life without a cerebellum. *Brain*, 133(3), 652–654.
- [123] Lennie, P. (2003). The cost of cortical computation. *Current biology*, 13(6), 493–497.
- [124] Li, C. (2024). RLWorms. *GitHub*.
- [125] Li, C., Brenner, J. W., Boesky, A., Ramanathan, S., & Kreiman, G. (2024). Neuron-level prediction and noise can implement flexible reward-seeking behavior. *bioRxiv*, (pp. 2024–05).
- [126] Li, C. & Deza, A. (2021). What matters in branch specialization? using a toy task to make predictions. In *SVRHM 2021 Workshop@ NeurIPS*.
- [127] Li, C. & Gullledge, A. T. (2021). Nmda receptors enhance the fidelity of synaptic integration. *Eneuro*, 8(2).
- [128] Li, C., Kreiman, G., & Ramanathan, S. (2022). Integrating artificial and biological neural networks to improve animal task performance using deep reinforcement learning. *bioRxiv*, (pp. 2022–09).
- [129] Liakoni, V., Lehmann, M. P., Modirshanechi, A., Brea, J., Lutti, A., Gerstner, W., & Preuschoff, K. (2022). Brain signals of a surprise-actor-critic model: Evidence for multiple learning modules in human decision making. *NeuroImage*, 246, 118780.

- [130] Lillicrap, T. (2023). Model-based reinforcement learning and the future of language models.
- [131] Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., & Hinton, G. (2020). Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6), 335–346. Number: 6 Publisher: Nature Publishing Group.
- [132] Lovas, J. R. & Yuste, R. (2021). Ensemble synchronization in the reassembly of hydra’s nervous system. *Current Biology*, 31(17), 3784–3796.
- [133] Lu, Y., Truccolo, W., Wagner, F. B., Vargas-Irwin, C. E., Ozden, I., Zimmermann, J. B., May, T., Agha, N. S., Wang, J., & Nurmikko, A. V. (2015). Optogenetically induced spatiotemporal gamma oscillations and neuronal spiking activity in primate motor cortex. *Journal of Neurophysiology*, 113(10), 3574–3587. Publisher: American Physiological Society.
- [134] Luczak, A., McNaughton, B. L., & Kubo, Y. (2022). Neurons learn by predicting future activity. *Nature Machine Intelligence*, 4(1), 62–72.
- [135] Markram, H. (2006). The blue brain project. *Nature Reviews Neuroscience*, 7(2), 153–160.
- [136] McMullin, B. (2004). Thirty years of computational autopoiesis: A review. *Artificial life*, 10(3), 277–295.
- [137] Miconi, T., Rawal, A., Clune, J., & Stanley, K. O. (2020). Backpropamine: training self-modifying neural networks with differentiable neuromodulated plasticity. *arXiv preprint arXiv:2002.10585*.
- [138] Miconi, T., Stanley, K., & Clune, J. (2018). Differentiable plasticity: training plastic neural networks with backpropagation. In *International Conference on Machine Learning* (pp. 3559–3568).: PMLR.
- [139] Millidge, B. (2019). Combining active inference and hierarchical predictive coding: A tutorial introduction and case study.
- [140] Millidge, B., Salvatori, T., Song, Y., Bogacz, R., & Lukasiewicz, T. (2022a). Predictive coding: Towards a future of deep learning beyond backpropagation? *arXiv preprint arXiv:2202.09467*.
- [141] Millidge, B., Seth, A., & Buckley, C. L. (2022b). Predictive Coding: a Theoretical and Experimental Review. *arXiv:2107.12979 [cs, q-bio]*.
- [142] Millidge, B., Tschantz, A., & Buckley, C. L. (2022c). Predictive coding approximates backprop along arbitrary computation graphs. *Neural Computation*, 34(6), 1329–1368.
- [143] Mirescu, C. & Gould, E. (2006). Stress and adult neurogenesis. *Hippocampus*, 16(3), 233–238.



- [144] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533. Number: 7540 Publisher: Nature Publishing Group.
- [145] Mueller, S., Wang, D., Fox, M. D., Yeo, B. T. T., Sepulcre, J., Sabuncu, M. R., Shafee, R., Lu, J., & Liu, H. (2013). Individual Variability in Functional Connectivity Architecture of the Human Brain. *Neuron*, 77(3), 586–595.
- [146] Mumford, D. (1992). On the computational architecture of the neocortex: Ii the role of cortico-cortical loops. *Biological cybernetics*, 66(3), 241–251.
- [147] Nagel, G., Szellas, T., Huhn, W., Kateriya, S., Adeishvili, N., Berthold, P., Ollig, D., Hege-  
mann, P., & Bamberg, E. (2003). Channelrhodopsin-2, a directly light-gated cation-selective  
membrane channel. *PNAS*, 100(24), 13940–13945.
- [148] Nichols, A. L., Eichler, T., Latham, R., & Zimmer, M. (2017). A global brain state underlies  
c. elegans sleep behavior. *Science*, 356(6344), eaam6851.
- [149] Nikishin, E., Izmailov, P., Athiwaratkun, B., Podoprikin, D., Garipov, T., Shvechikov, P.,  
Vetrov, D., & Wilson, A. G. (2018). Improving Stability in Deep Reinforcement Learning  
with Weight Averaging.
- [150] Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., & Carter, S. (2020). Zoom in:  
An introduction to circuits. *Distill*. <https://distill.pub/2020/circuits/zoom-in>.
- [151] Oliviers, G., Bogacz, R., & Meulemans, A. (2024). Learning probability distributions of  
sensory inputs with monte carlo predictive coding. *bioRxiv*, (pp. 2024–02).
- [152] Oord, A. v. d., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predic-  
tive coding. *arXiv preprint arXiv:1807.03748*.
- [153] OpenAI, Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi,  
D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J.,  
Petrov, M., Pinto, H. P. d. O., Raiman, J., Salimans, T., Schlatter, J., Schneider, J., Sidor, S.,  
Sutskever, I., Tang, J., Wolski, F., & Zhang, S. (2019). *Dota 2 with Large Scale Deep Rein-  
forcement Learning*. Technical Report arXiv:1912.06680, arXiv. arXiv:1912.06680 [cs, stat]  
type: article.
- [154] Palyanov, A., Khayrulin, S., Larson, S. D., & Dibert, A. (2012). Towards a virtual c. elegans:  
A framework for simulation and visualization of the neuromuscular system in a 3d physical  
environment. *In silico biology*, 11(3-4), 137–147.

- [155] Park, S.-G., Jeong, Y.-C., Kim, D.-G., Lee, M.-H., Shin, A., Park, G., Ryoo, J., Hong, J., Bae, S., Kim, C.-H., Lee, P.-S., & Kim, D. (2018). Medial preoptic circuit induces hunting-like actions to target objects and prey. *Nature Neuroscience*, 21(3), 364–372. Number: 3 Publisher: Nature Publishing Group.
- [156] Parr, T., Pezzulo, G., & Friston, K. J. (2022). *Active inference: the free energy principle in mind, brain, and behavior*. MIT Press.
- [157] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32* (pp. 8024–8035). Curran Associates, Inc.
- [158] Peckham, P. H. & Knutson, J. S. (2005). Functional electrical stimulation for neuromuscular applications. *Annual Review of Biomedical Engineering*, 7, 327–360.
- [159] Pessoa, L. (2023). The entangled brain. *Journal of cognitive neuroscience*, 35(3), 349–360.
- [160] Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., et al. (2020). Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*.
- [161] Rao, R. P., Gklezacos, D. C., & Sathish, V. (2022). Active predictive coding: A unified neural framework for learning hierarchical world models for perception and planning. *arXiv preprint arXiv:2210.13461*.
- [162] Razeto-Barry, P. (2012). Autopoiesis 40 years later. a review and a reformulation. *Origins of Life and Evolution of Biospheres*, 42(6), 543–567.
- [163] Riddle, D. L., Blumenthal, T., Meyer, B. J., & Priess, J. R. (1997). *Mechanosensory Control of Locomotion*. Cold Spring Harbor Laboratory Press. Publication Title: *C. elegans II*. 2nd edition.
- [164] Romano, D., Donati, E., Benelli, G., & Stefanini, C. (2019). A review on animal–robot interaction: from bio-hybrid organisms to mixed societies. *Biological Cybernetics*, 113(3), 201–225.
- [165] Roostaei, T., Nazeri, A., Sahraian, M. A., & Minagar, A. (2014). The human cerebellum. *Neurologic clinics*, 32(4), 859–869.
- [166] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088), 533–536.

- [167] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3), 211–252.
- [168] Sacks, O. (2012). *An anthropologist on Mars: Seven paradoxical tales*. Vintage.
- [169] Sakai, A., Sunagawa, T., Madan, S., Suzuki, K., Katoh, T., Kobashi, H., Pfister, H., Sinha, P., Boix, X., & Sasaki, T. (2022). Three approaches to facilitate invariant neurons and generalization to out-of-distribution orientations and illuminations. *Neural Networks*, 155, 119–143.
- [170] Salvatori, T., Song, Y., Millidge, B., Xu, Z., Sha, L., Emde, C., Bogacz, R., & Lukasiewicz, T. (2022). Incremental predictive coding: A parallel and fully automatic learning algorithm. *arXiv preprint arXiv:2212.00720*.
- [171] Salzman, C., D., Britten, K. H., & Newsome, W. T. (1990). Cortical microstimulation influences perceptual judgements of motion direction. *Nature*, 346, 174–177.
- [172] Sanchez, C. J., Chiu, C.-W., Zhou, Y., González, J. M., Vinson, S. B., & Liang, H. (2015). Locomotion control of hybrid cockroach robots. *Journal of The Royal Society Interface*, 12(105), 20141363. Publisher: Royal Society.
- [173] Sarma, G. P., Lee, C. W., Portegys, T., Ghayoomie, V., Jacobs, T., Alicea, B., Cantarelli, M., Currie, M., Gerkin, R. C., Gingell, S., Gleeson, P., Gordon, R., Hasani, R. M., Idili, G., Khayrulin, S., Lung, D., Palyanov, A., Watts, M., & Larson, S. D. (2018). OpenWorm: overview and recent advances in integrative biological simulation of *Caenorhabditis elegans*. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1758), 20170382. Publisher: Royal Society.
- [174] Sato, H., Berry, C. W., Casey, B. E., Lavella, G., Yao, Y., VandenBrooks, J. M., & Maharbiz, M. M. (2008). A cyborg beetle: Insect flight control through an implantable, tetherless microsystem. In *2008 IEEE 21st International Conference on Micro Electro Mechanical Systems* (pp. 164–167). ISSN: 1084-6999.
- [175] Schild, L. C. & Glauser, D. A. (2015). Dual Color Neural Activation and Behavior Control with Chrimson and CoChR in *Caenorhabditis elegans*. *Genetics*, 200(4), 1029–1034.
- [176] Schmidt, S., Gull, S., Herrmann, K.-H., Boehme, M., Irintchev, A., Urbach, A., Reichenbach, J. R., Klingner, C. M., Gaser, C., & Witte, O. W. (2021). Experience-dependent structural plasticity in the adult brain: How the learning brain grows. *NeuroImage*, 225, 117502.
- [177] Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T., & Silver, D. (2020). Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839), 604–609. Number: 7839 Publisher: Nature Publishing Group.

- [178] Shorten, C. & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60.
- [179] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484–489.
- [180] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., & Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*, 550(7676), 354–359.
- [181] Smith, M. A.-Y., Honegger, K. S., Turner, G., & de Bivort, B. (2022). Idiosyncratic learning performance in flies. *Biology Letters*, 18(2), 20210424.
- [182] Song, Y., Millidge, B., Salvatori, T., Lukasiewicz, T., Xu, Z., & Bogacz, R. (2022). Inferring Neural Activity Before Plasticity: A Foundation for Learning Beyond Backpropagation. Pages: 2022.05.17.492325 Section: New Results.
- [183] Song, Y., Millidge, B., Salvatori, T., Lukasiewicz, T., Xu, Z., & Bogacz, R. (2024). Inferring neural activity before plasticity as a foundation for learning beyond backpropagation. *Nature Neuroscience*, (pp. 1–11).
- [184] Standage, T. (2016). From not working to neural networking. *The Economist. Special Report*.
- [185] Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., & Christiano, P. F. (2020). Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33, 3008–3021.
- [186] Sun, Z. & Firestone, C. (2020). The dark room problem. *Trends in Cognitive Sciences*, 24(5), 346–348.
- [187] Sussillo, D., Stavisky, S. D., Kao, J. C., Ryu, S. I., & Shenoy, K. V. (2016). Making brain–machine interfaces robust to future neural variability. *Nature communications*, 7(1), 1–13. Publisher: Nature Publishing Group.
- [188] Sutton, R. (2019). The bitter lesson. *Incomplete Ideas (blog)*, 13(1), 38.
- [189] Sutton, R. S. & Barto, A. G. (2018a). *Reinforcement learning: An introduction*. MIT press.
- [190] Sutton, R. S. & Barto, A. G. (2018b). *Reinforcement Learning, second edition: An Introduction*. MIT Press. Google-Books-ID: uWV0DwAAQBAJ.

- [191] Talwar, S. K., Xu, S., Hawley, E. S., Weiss, S. A., Moxon, K. A., & Chapin, J. K. (2002). Rat navigation guided by remote control. *Nature*, 417(6884), 37–38. Number: 6884 Publisher: Nature Publishing Group.
- [192] Tankus, A., Fried, I., & Shoham, S. (2014). Cognitive-motor brain–machine interfaces. *Journal of physiology, Paris*, 108(1), 38–44.
- [193] Thiebaut de Schotten, M. & Forkel, S. J. (2022). The emergent properties of the connected brain. *Science*, 378(6619), 505–510.
- [194] Troemel, E. R., Sagasti, A., & Bargmann, C. I. (1999). Lateral signaling mediated by axon contact and calcium entry regulates asymmetric odorant receptor expression in *C. elegans*. *Cell*, 99(4), 387–398.
- [195] Tschantz, A., Baltieri, M., Seth, A. K., & Buckley, C. L. (2020a). Scaling active inference. In *2020 international joint conference on neural networks (ijcnn)* (pp. 1–8).: IEEE.
- [196] Tschantz, A., Seth, A. K., & Buckley, C. L. (2020b). Learning action-oriented models through active inference. *PLoS computational biology*, 16(4), e1007805.
- [197] Ushiku, Y., Muraoka, H., Inaba, S., Fujisawa, T., Yasumoto, K., Gunji, N., Higuchi, T., Hara, Y., Harada, T., & Kuniyoshi, Y. (2012). Isi at imageclef 2012: Scalable system for image annotation. In *CLEF (Online Working Notes/Labs/Workshop)*: Citeseer.
- [198] Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C., & Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782), 350–354.
- [199] Voss, C., Goh, G., Cammarata, N., Petrov, M., Schubert, L., & Olah, C. (2021). Branch specialization. *Distill*, 6(4), e00024–008.
- [200] Weiss, T., Soroka, T., Gorodisky, L., Shushan, S., Snitz, K., Weissgross, R., Furman-Haran, E., Dhollander, T., & Sobel, N. (2020). Human olfaction without apparent olfactory bulbs. *Neuron*, 105(1), 35–45.
- [201] Wen, Q., Po, M. D., Hulme, E., Chen, S., Liu, X., Kwok, S. W., Gershow, M., Leifer, A. M., Butler, V., Fang-Yen, C., Kawano, T., Schafer, W. R., Whitesides, G., Wyart, M., Chklovskii, D. B., Zhen, M., & Samuel, A. D. T. (2012). Proprioceptive Coupling within Motor Neurons Drives *C. elegans* Forward Locomotion. *Neuron*, 76(4), 750–761.
- [202] Wilson, E. O. (1975). *Sociobiology: The new synthesis*. Sociobiology: The new synthesis. Oxford, England: Belknap Press of Harvard U Press. Pages: ix, 697.

- [203] Witvliet, D., Mulcahy, B., Mitchell, J. K., Meirovitch, Y., Berger, D. R., Wu, Y., Liu, Y., Koh, W. X., Parvathala, R., Holmyard, D., et al. (2021). Connectomes across development reveal principles of brain maturation. *Nature*, 596(7871), 257–261.
- [204] Wong, C.-C., Chien, S.-Y., Feng, H.-M., & Aoyama, H. (2021). Motion Planning for Dual-Arm Robot Based on Soft Actor-Critic. *IEEE Access*, 9, 26871–26885. Conference Name: IEEE Access.
- [205] Wright, A. J. & Jackson, J. J. (2023). Are some people more consistent? examining the stability and underlying processes of personality profile consistency. *Journal of Personality and Social Psychology*, 124(6), 1314.
- [206] Wurman, P. R., Barrett, S., Kawamoto, K., MacGlashan, J., Subramanian, K., Walsh, T. J., Capobianco, R., Devlic, A., Eckert, F., Fuchs, F., Gilpin, L., Khandelwal, P., Kompella, V., Lin, H., MacAlpine, P., Oller, D., Seno, T., Sherstan, C., Thomure, M. D., Aghabozorgi, H., Barrett, L., Douglas, R., Whitehead, D., Dürr, P., Stone, P., Spranger, M., & Kitano, H. (2022). Outracing champion Gran Turismo drivers with deep reinforcement learning. *Nature*, 602(7896), 223–228.
- [207] Xu, J., Galardi, M. M., Pok, B., Patel, K. K., Zhao, C. W., Andrews, J. P., Singla, S., McCafferty, C. P., Feng, L., Musonza, E. T., Kundishora, A. J., Gummadavelli, A., Gerrard, J. L., Laubach, M., Schiff, N. D., & Blumenfeld, H. (2020). Thalamic Stimulation Improves Postictal Cortical Arousal and Behavior. *Journal of Neuroscience*, 40(38), 7343–7354. Publisher: Society for Neuroscience Section: Research Articles.
- [208] Yan, G., Vértes, P. E., Towlson, E. K., Chew, Y. L., Walker, D. S., Schafer, W. R., & Barabási, A.-L. (2017). Network control principles predict neuron function in the caenorhabditis elegans connectome. *Nature*, 550(7677), 519–523.
- [209] Yang, J., Huai, R., Wang, H., Lv, C., & Su, X. (2015). A robo-pigeon based on an innovative multi-mode telestimulation system. *Bio-Medical Materials and Engineering*, 26 Suppl 1, S357–363.
- [210] Zeiler, M. D. & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818–833).: Springer.
- [211] Zheng, N., Jin, M., Hong, H., Huang, L., Gu, Z., & Li, H. (2017). Real-time and precise insect flight control system based on virtual reality. *Electronics Letters*, 53(6), 387–389. \_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1049/el.2016.3048>.
- [212] Zheng, N., Ma, Q., Jin, M., Zhang, S., Guan, N., Yang, Q., & Dai, J. (2019). Abdominal-Waving Control of Tethered Bumblebees Based on Sarsa With Transformed Reward. *IEEE Transactions on Cybernetics*, 49(8), 3064–3073. Conference Name: IEEE Transactions on Cybernetics.

- [213] Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., & Torralba, A. (2017). Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6), 1452–1464.







**T**HIS THESIS WAS TYPESET using  $\LaTeX$ , originally developed by Leslie Lamport and based on Donald Knuth's  $\TeX$ . The body text is set in 11 point Egenolff-Berner Garamond, a revival of Claude Garamont's humanist typeface. The above illustration, "Science Experiment 02", was created by Ben Schlitter and released under [CC BY-NC-ND 3.0](#). A template that can be used to format a PhD thesis with this look and feel has been released under the permissive MIT (X11) license, and can be found online at [github.com/suchow/Dissertate](https://github.com/suchow/Dissertate) or from its author, Jordan Suchow, at [suchow@post.harvard.edu](mailto:suchow@post.harvard.edu).

