Winter 1-15-2021

Constructing and Analyzing Neural Network Dynamics for Information Objectives and Working Memory

Elham Ghazizadeh Ahsaei
Washington University in St. Louis

WASHINGTON UNIVERSITY IN ST. LOUIS

School of Engineering and Applied Science
Department of Electrical and Systems Engineering

Dissertation Examination Committee:
ShiNung Ching, Chair
Jr-Shin Li
Joseph O'Sullivan
Barani Raman
Lawrence Snyder

Constructing and Analyzing Neural Network Dynamics for Information Objectives and Working Memory

by

Elham Ghazizadeh Ahsaei

A dissertation presented to
The Graduate School
of Washington University in
partial fulfillment of the
requirements for the degree
of Doctor of Philosophy

January 2021
St. Louis, Missouri

Table of Contents

List of Figures

Acknowledgments

I would like to express my deep gratitude to Dr. ShiNung Ching for giving me the opportunity to conduct my Ph.D. research in the topics that interest me the most. I am grateful for his insightful guidance and helpful feedbacks to complete this dissertation.

I would like to thank my dissertation defense committee members, Dr. Jr-Shin Li, Dr. Joseph O'Sullivan, Dr. Barani Raman and Dr. Lawrence Snyder for taking the time out of their busy schedule and providing me with constructive comments to enrich my dissertation. I also would like to thank Dr. Arthur and Dr. O'Sullivan for all their kind and sincere supports throughout my Ph.D. studies.

To my current and previous lab-mates, thanks for the fun, support and cheerful memories. Special thanks to Dr. Peng Yi for helpful mentoring discussions about my research and enthusiastically encouraging me to keep on grinding out anytime I hit a wall with my project.

Also, I would like to acknowledge all my supportive and caring friends. Specially, I would like to thank Maryam and Ladan for being sisters as well as my best friends. Their presence has been the best gift that I could have ever wished for during my stay in St. Louis.

My parents and my brother have provided me with endless support, unconditional love and immense joy despite being far away from me during my Ph.D. studies. Even the most beautiful words fall short to portray the depth of my gratitude for having them in my life.

Elham Ghazizadeh Ahsaei

Washington University in Saint Louis

January 2021

Dedicated to my parents and my brother.

ABSTRACT OF THE DISSERTATION

Constructing and Analyzing Neural Network Dynamics for Information Objectives and Working Memory

by

Elham Ghazizadeh Ahsaei

Doctor of Philosophy in Electrical Engineering

Washington University in St. Louis, 2021

Professor ShiNung Ching, Chair

Creation of quantitative models of neural functions and discovery of underlying principles of how neural circuits learn and compute are long-standing challenges in the field of neuroscience. In this work, we blend ideas from computational neuroscience, information and control theories with machine learning to shed light on how certain key functions are encoded through the dynamics of neural circuits. In this regard, we pursue the 'top-down' modeling approach of engineering neuroscience to relate brain functions to basic generative dynamical mechanisms. Our approach encapsulates two distinct paradigms in which 'function' is understood. In the first part of this research, we explore the synthesis of neural dynamics for task-independent, well-defined objective function: the *information processing capacity* of neural circuits/networks. We contribute our efforts to devise a strategy to optimize the dynamics of the network at hand using information maximization as an objective function. In this vein, our principle contributions are in terms of mathematical formulation of the optimization problem and proposing a simplification method to reduce the computational burden associated with mutual information optimization. Then, we illustrate the novelty of our ideas for well-understood dynamical systems. Our methodology results in dynamics that generically perform as encoder of afferent inputs distribution and facilitate information propagation. However, determining

a well-defined mathematical objective function may not be straightforward in all cases, e.g. complex cognitive functions. To address this issue, in the second part of this research we consider top-down synthesis on the basis of a surrogate task. In particular, we optimize 'artificial' recurrent networks in order to perform a computational task that embodies the function we are interested in studying, i.e. *working memory*. We contribute our efforts to propose a realistic training paradigm for recurrent neural networks and elucidate how dynamics of the optimized artificial networks can support computations implemented in memory functions. We will discuss the theoretical and technical steps involved in our interpretations, as well as remaining open questions and future directions.

Chapter 1

Introduction

1.1 Overview and Context

Neuroscience is a wide-ranging field of study that involves understanding the biological and computational basis of behavior, learning, memory, perception and action [1]. Historically, a driving motivator in neuroscience has been a desire to link observations about cognitive behavior with actual physical and physiological processes that support such behavior [2]. Early studies of the brain began with empirical investigations of the mind, the essence of feelings and the causes of psychiatric disorders [1, 3]. With the development of scientific and experimental methods, scientists argued about the relation between mental processes and different brain regions [4]. For instance, during the 19th century, scientists postulated as to the existence of a specific area for each mental processes in the brain (i.e., the 'homonculus' [5]) and thus people's skills and abilities could be recognized by the size and shape of their skulls [5]. In the 1860s, Pierre Broca performed seminal research studying the relationship between language and brain through his clinical studies of patients with brain damage [2].

More recently, the advent of electroencephalography, modern imaging techniques and advances in neural recording and stimulation devices has led to tremendous progress in understanding the structure of the brain at cellular and molecular levels [2, 6] and the discovery of a wide range of brain electrical phenomena, from the spiking activity of neurons to the slower oscillations of small populations [7]. However, despite these decades of progress, there are still countless unanswered questions about the brain function and many research challenges yet remain to be addressed.

A paramount challenge in contemporary systems neuroscience research involves the elucidation of formal links between activity in large-scale neural circuits and complex behaviors [3, 5, 8]. While experimental capability has elevated to prospects of observing even-increasing swaths of the brain, there remains a gap in technical capability as conceptual and theoretical formalisms for making sense of these high-dimensional observations [9]. In this regard, it has been widely recognized that to confront the immense complexity of the brain across multiple scales of temporal and spatial organization and extract theoretical/conceptual models of the brain functions and neural data, neuroscience can benefit from tools and theories from other disciplines such as mathematics, physics and engineering, within the overall framework of 'computational neuroscience' [10]. In particular, neural systems are not simply an entangled web of complexities that exist for their own sake, but rather they have evolved to solve a myriad of computational problems. Thus, engineering science, including design and analysis frameworks in signal processing, dynamical systems and control theory, can serve as sources of inspiration to understand how neural systems may be solving certain problems and as tools for analyzing models of neural circuits [10, 11].

1.1.1 Engineering Neuroscience and Dynamics in the Brain

A long-standing challenge in computational neuroscience involves studying the brain computations underlying complex functions such as sensory coding, muscle control, perception and decision making [11, 12, 13]. In this regard, there is one aspect of neural activity where engineering theory can play an especially powerful role: the study of brain *dynamics*, or the generative mechanisms within neural circuits that give rise to time-varying neural activity that eventually forms the substrate for complex behavior [14].

Dynamics in the nervous system can be understood at many levels of spatial scale ranging from molecules, synapses, neurons, networks, maps and systems (see Fig. 1.1). In this dissertation, focus is directed at the neuronal and circuit/network level of description. In particular, we recognize that neural activity is not simply mapped statically to high level-functions such as perception, movement and learning but rather evolves through time as a function of internal and external processes. As such, the brain is a coevolving dynamic system and characterization of its dynamics is fundamental to our understanding of its function. In this regard, engineering theory is ideally situated to lend novel hypothesis and interpretations regarding the mechanisms and role of brain dynamics. This paradigm of 'engineering neuroscience' can constitute a new and potentially powerful paradigm in computational neuroscience, and its exploration is a central arc of this dissertation.

1.1.2 Dichotomy of Bottom-up and Top-down Modeling

A key approach in computational and engineering neuroscience is the use of mathematical models to guide scientific inquiry regarding functional mechanisms within the brain.

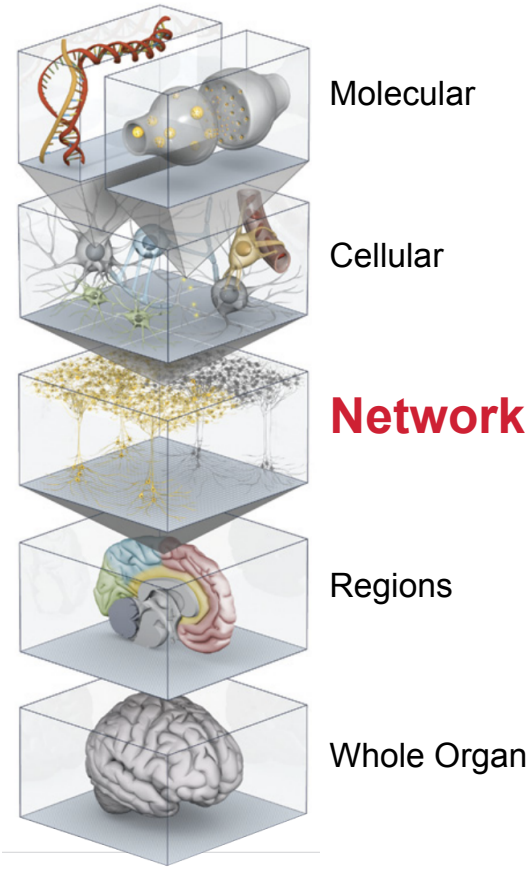

Figure 1.1: Different levels of the brain functions [15].

Development of mechanistic circuit models that link neural activity with the high-level brain function can be organized into two broad categories: 1) Bottom-up modeling and 2) Top-down modeling. In the former, one starts by studying and modeling the dynamics of low-level circuit parts (i.e., neurons), which are then composed into larger networks that are then studied to infer high-level functionality. Such methods offer potentially high levels of biophysical interpretability and compatibility, and have a long history in computational neuroscience. Bottom-up approaches have been particularly successful at creating explanatory circuit models for broad electrophysiological phenomena such as oscillations and synchrony [16].

Conversely, in the top-down approach, one starts by formulating a high-level hypothesis about a function, which is then used as a means to construct or synthesize the low level parts. In neuroscience, such a paradigm has sometimes been referred to as a 'normative' approach [8, 16] and relies heavily on methods from optimization. The resultant synthesized dynamics can then be evaluated for biological plausibility by comparing the predictive behavior of the model with observed/recorded neural data. Both bottom-up and top-down approaches can lead to quite complex dynamical models that are difficult to analyze analytically [8]. Numerical methods, including dimensionality reduction and other manifold learning strategies, are thus important tools in elucidating the fundamental dynamical mechanisms embedded within these models [17].

This dissertation will focus on the top-down engineering neuroscience approach. Our goal is to build network dynamics that we can analyze in order to understand basic generative mechanisms, while also explicitly relating to high-level functional requirements. Our goal is to understand how *and* why neural activity is generated within brain networks. With most computational approaches in neuroscience, we make particular choices in the tradeoff between numerical and analytical tractability and the level of biophysical detail embodied in our models. Our approach encapsulates two distinct paradigms in which 'function' is understood. A schematic of the dissertation roadmap is shown in Fig. 1.2.

1.1.3 Normative Synthesis with and without Mathematical Objectives

Specifically, the first part of the dissertation explores the construction of neuronal dynamics for a well-defined mathematical objective that is not explicitly linked to a particular task. This objective encapsulates, in a general sense, the *information processing capacity* of neural circuits.

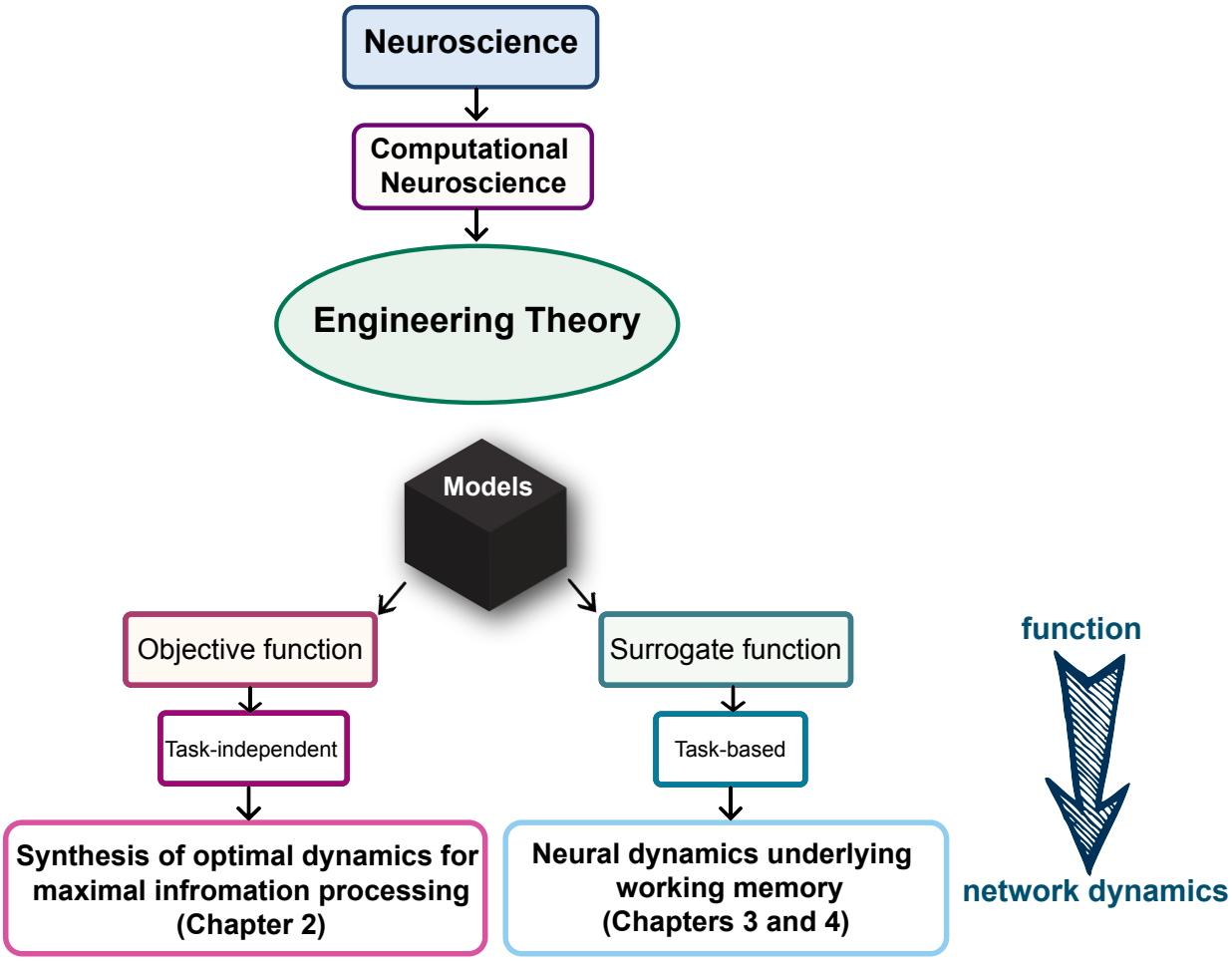

Figure 1.2: Dissertation structure

We elucidate the neuronal dynamics that are best conducive of information propagation in neural circuits via extremizing this objective function.

However, determining a well-defined mathematical objective function may not be straightforward in all cases. For example, complex cognitive processes may not be neatly quantified in a closed-form expression. To tackle this issue, in the second part of the dissertation we consider top-down synthesis on the basis of a surrogate task. That is, we optimize networks in order to perform a computational task that embodies the function we are interested in studying. Such an approach draws heavily from recent efforts in distributed optimization and

neural-network-based machine learning [18]. The function we focus on is working memory. Specifically, to generate a top-down model of working memory function, we optimize 'artificial' neural networks to perform cognitive tasks that exhibit working memory requirements. Next, we elucidate how dynamics of the optimized artificial networks can support computations implemented in memory function. In the following, we elaborate on some of the details in a brief introduction for each chapter of the dissertation.

1.2 Synthesis of Dynamics for Information Objectives (Chapter 2)

Fundamental biological functions such as perception, cognition, action and adaptation rely on neural information processing and thus neural circuits naturally lend themselves to be studied with information theory [19]. Shannon's theory of information was originally used to analyze and optimize man-made communication systems, for which the functioning principles are known. Nevertheless, it was soon realized that information theory can be applied to broader settings (e.g. theoretical neuroscience) to obtain insight about the functioning of systems for which the underlying principles are far from fully understood, e.g. neural systems [20].

In this context, the concept of empowerment was first introduced in [21] as a hypothetical, information-based utility function which might be considered as a formal definition of 'intrinsic motivation' for learning and adaptation. Subsequently, empowerment has been widely used in reinforcement learning as a general framework for obtaining agent policies based on the value of information rather than manually-designed, task-specific utility functions [22, 23, 24]. Further, the mathematical definition of empowerment is highly related to the 'Infomax' objective used in theoretical studies of neural coding [25, 26, 27]. At a conceptual level,

empowerment can be understood as the maximum potential effect that an input can impart on the states of a system. In other words, a high empowerment corresponds to a system with higher information processing capacity.

In this chapter, we pursue the top-down approach of computational modeling and address *what* sort of neural network dynamics are most conducive of afferent information? We formalize our hypothesis using a well-defined information theoretic objective function, here empowerment. In this regard, if empowerment is high, then the inputs are well-encoded in the states of the network in question. Thus, we optimize the network's empowerment to shed light on the neural dynamics capable of such functionality.

Particularly, the main facet of our approach involves adapting the information-theoretic definition of empowerment into the context of dynamical system, which in turn are used as abstract models of neural circuits. However, the calculations underlying empowerment are highly taxing for high dimensional state space of neural circuits. We contribute our efforts to posit a simplification on the empowerment calculation based on the system impulse response in conjunction with the adaptation of the variational lower bound for empowerment approximation. Then, we explicitly define a parametrization of the dynamical system for which we seek to maximize the empowerment. We will elaborate on the mathematical details of the optimization of variational empowerment with respect to the parameterization of system dynamics in this chapter.

Afterward, we proceed to show the efficacy of our proposed method on canonical and well-understood dynamical systems such as linear dynamical systems and inverted pendulum as well as a well known neural mass model, i.e. the Wilson-Cowan model. Our results show that the maximally conducive dynamics created via this procedure lead to 'simple' dynamics with

a single unstable fixed point and a large number of reachable states, which agrees with basic intuition and the sort of neural dynamics thought to be pervasive in actual neuronal circuits.

1.3 Optimization and Learning of Network Dynamics via Reinforced Regression for High-level Function (Chapter 3)

As part of the Central Nervous System (CNS), the brain regulates several functions including vital functions such as breathing, basic functions like sleeping or eating and high-level functions such as reasoning, verbal communication and holding traces of complex information online simultaneously. In particular, high-level cognitive functions are mental operations by which the brain derives representation of relevant information from external inputs and use them to modify or produce behavior [3]. Despite decades of neuroscience research, understanding the neural bases of cognitive functions is yet a challenge [5]. For instance, it is not yet fully understood how sensory information is perceived and how such perceptions are transformed into meaningful representations to be recruited into immediate plans or performing complex tasks, like decision making. Clearly, these are complicated processes and deriving such an understanding is not simple.

In Chapter 3, we will apply the top-down approach of theoretical neuroscience and try to answer this question: *how* neural circuit dynamics give rise to high-level functions? Answering this question requires formulation of a hypothesis that explains the high-level function in question. However, in this case it is not feasible to determine an explicit, well-defined objective function that generates such function. To tackle this challenge, we turn to task-based surrogate modeling via the artificial counterpart of neuronal circuits, i.e. artificial

neural networks. As such, by optimizing artificial recurrent neural networks (RNNs) to implement cognitive tasks we can identify the dynamical system that generates input to output mappings. Then, the emergent dynamics of the synthesized network can be studied as a generative mechanism underlying the computations of the high-level function.

Although RNNs have been widely adopted to generate hypothesis about operations of neuronal circuits, cautionary details need to be considered when reasoning about certain principles that govern neural computations [28, 29]. Because each optimized RNN merely generates a potential hypothesis that can achieve the task at hand, the ability of these optimized networks to predict actual brain activity may depend crucially on certain restrictions regarding the optimization method that is used; e.g., by regularizing or encouraging solutions that manifest more realistic connection motifs [30, 31].

Expanding on the aforementioned idea, we propose a training method for RNNs by adding realistic constraints on the learning paradigm. Specifically, we blend ideas from trial-based reinforcement learning with online error regression method to develop our training method. We proceed this chapter by elaborating on the computational details of the proposed method.

1.4 Analysis of Emergent Dynamics Associated with Working Memory (Chapter 4)

In Chapter 3, we developed the framework to generate mechanistic hypotheses of high-level cognitive functions. We will deploy this optimization paradigm in Chapter 4 by considering a key component of cognitive functions, i.e. *working memory*, that allows to carry out various tasks such as learning, problem-solving, natural language, reasoning and planning. In particular, working memory (WM) is a temporary store that allows for active manipulation

of information in the absence of external stimuli [32]. In this regard, understanding how neural circuits retain and represent memory have been the focus of substantial experimental and computational neuroscience research.

In particular, WM is associated with a delay period following stimulus presentation and prior to onset of a behavioral response or action, such that the memory of past stimuli is retained via an invariant neuronal representation [33, 34, 35]. Many experimental and theoretical works on WM center on understanding neural activity during these delay periods [33, 36, 37]. Characterization of delay activity dichotomizes into two broad categories: 1) persistent activity, wherein neurons are tuned to relevant features of a stimulus and produce a relatively constant activity throughout delay intervals [38, 39]. 2) time-varying activity, wherein neural activity transiently ramp up and down during delay periods [37, 40]. Presumably, these two descriptions may have distinct underlying mechanisms [41, 42, 43].

To disambiguate the underpinning dynamical mechanisms of neural activity patterns during WM periods, in this chapter, we first elaborate on the design of a simple cognitive task that exhibits WM requirements. Next, we address the remaining open questions regarding the circumstances under which different network dynamics may arise and the specific functional advantages of one versus the other. In this regard, we first optimize thousands of RNNs across different hyperparameters and initialization schemes. Then, we reverse engineer the trained networks to reveal the dynamical properties of the networks. Interestingly, we identify a diversity of mechanisms that achieve the desired task computations but varying in their key dynamical properties. In particular, we find that networks can display predominantly asymptotically stable fixed points, stable limit cycle attractors, or a combination thereof.

To shed light on the functional advantages of each of these dynamical mechanisms, we design post hoc experiments to interrogate the trained networks. In particular, we examine the

durability of memory representations to increase in memory demands and the robustness of networks performance to perturbations. Our key finding is that memory representations need not to be encoded directly in network attractors, but can be mediated by transient dynamics formed in the overall network vector field. It turns out that such a mechanism is both efficient and robust as compared to the other emergent network dynamics.

1.5 Summary of Contributions (Chapter 5)

This dissertation, in the domain of engineering neuroscience, will show how merging methods and ideas from engineering theory with neuroscience can lend the conceptual hypothesis regarding how the brain works. Accordingly, the contributions of this dissertation coincide with both basic engineering theory and in theoretical neuroscience. The major specific contributions are:

1.5.1 Engineering theory

- A method to simplify the computational burden of calculating empowerment, i.e., the information-theoretic channel capacity associated with a dynamical system. (Chapter 2)
- A method to find optimal neuronal dynamics for empowerment maximization via considering a parametrization along a continuum of possible dynamics. (Chapter 2)
- A method to optimize recurrent neural networks in the context of task-based top-down modeling. The method blends trial-based reinforcement learning with continuous regression methods. (Chapter 3)

- Development of numerical sensitivity analyses to elucidate the different dynamical methods embedded within neural network models, in order to link these dynamical mechanisms to high-level functionality. (Chapter 4)

1.5.2 Theoretical neuroscience

- A top-down approach to understand the relation between a general, task-independent function and neuronal dynamics, where the function can be achieved by maximizing a well-defined mathematical objective function, i.e. empowerment. (Chapter 2)
- Understanding the functional advantages of oscillatory neuronal dynamics in the context of information encoding. (Chapter 2)
- A detailed accounting of different network-level mechanisms that can sustain a specific cognitive function: working memory. (Chapter 2)
- A analysis of the functional tradeoffs associated with the above competing mechanisms and an assessment of their biological significance and plausibility. (Chapter 4)

The dissertation will conclude in Chapter 5 with concluding remarks and a brief discussion on perspectives gleaned, remaining challenges and prospects for future inquiry along these lines of investigation.

Chapter 2

Synthesis of Dynamics for Information Objectives

2.1 Introduction

Neural circuits process information and examining such processings is of prime importance to the goal of obtaining insight on the functionality of neural systems from a theoretical standpoint. This relates to the fundamental problem of neural coding, or how circuits in the brain represent afferent information. In particular, we define the notion of functionality in terms of information theoretic quantities, since such quantities allow us to address the concept of optimality in a general way that is well-defined, task-independent and applicable universally to any input-system interaction. As such, employing the top-down modeling approach, we use the notion of empowerment as our objective function. Conceptually, empowerment can be understood as the maximum potential effect that an input can impart on the states of a system. Thus, empowerment is a property of the input-to-output (or, input-to-state)

transfer function of a dynamical system and is fundamentally related to the notions of channel capacity and reachability, from communication and control theories, respectively [44]. The greater the empowerment of a system, the greater the diversity of states/outputs an input can induce. This problem setup allows the desirable dynamics of the network to emerge only via the empowerment maximization criterion without having any a priori knowledge about the optimal dynamics.

In the remaining of this chapter, we first adopt the definition of empowerment to the dynamical system in question. Next, we elaborate on the mathematical details of empowerment approximation based on our proposed method. Lastly, we reveal the optimal system dynamics obtained from empowerment maximization.

2.2 Problem Formulation and the System Model

In this section, we present the mathematical definition of empowerment as our objective function. Then, we provide the dynamical system model under consideration and specifically note the parameterization over which empowerment will be maximized. Throughout this chapter the terms dynamical system and network are used interchangeably.

2.2.1 Empowerment

Mutual Information is a core information theoretic quantity that measures the dependency between two random variables. In our formulation, the dynamical system can be perceived as a communication channel from afferent inputs to states. Given the current state s, the mutual information between the input $\mathbf{u}$ and the final state $\mathbf{s}'$ is:

$$\mathcal{I}(\mathbf{u};\mathbf{s}'|\mathbf{s}=\mathrm{s}) = \int\int p(\mathrm{s}'|\mathrm{u},\mathrm{s})\,\omega(\mathrm{u}|\mathrm{s})\log\frac{p(\mathrm{s}',\mathrm{u}|\mathrm{s})}{p(\mathrm{s}'|\mathrm{s})\omega(\mathrm{a}|\mathrm{s})}d\mathrm{u}\,d\mathrm{s}' \tag{2.1}$$

where we denote $\omega(\mathrm{u}|\mathrm{s})$ as the *input distribution* and $p(\mathrm{s}',\mathrm{u}|\mathrm{s})$ as the *transition distribution*. With this formulation, empowerment is simply the channel capacity, i.e., the maximum information that an input can potentially emit to the system by manipulating the states [21, 45] via actions/inputs. Particularly, for the given current state, the empowerment is:

$$\mathcal{E}(\mathrm{s}) = \max_{\omega(\mathrm{u}|\mathrm{s})} \mathcal{I}(\mathbf{u};\mathbf{s}'|\mathrm{s}) \tag{2.2}$$

Here, u can be seen as an exogenous input (that is uncorrelated with the dynamics of the system), referred to in the communication theory literature as a 'source'. Consequently, empowerment quantifies the extent to which the source is encoded in the system states. Fig. 2.1 shows empowerment landscape for a maze.

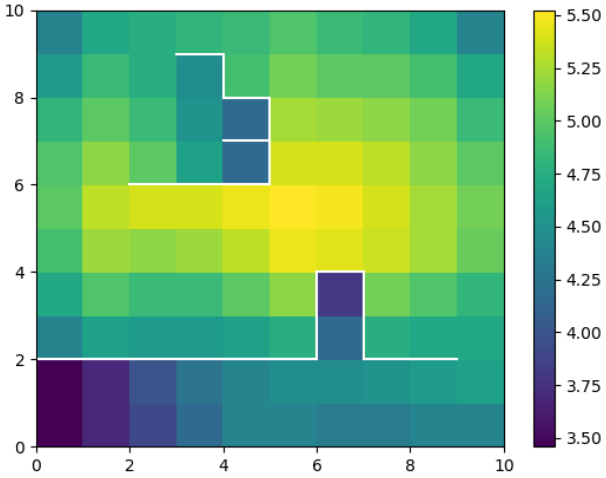

Figure 2.1: Empowerment landscape for a maze, where the value of empowerment (over 5 time steps) is represented for each state. The white lines display the walls in the maze. In the center of the maze, where there is more number of possible states (as compared to the corners or near the walls) the value of empowerment is higher and thus the input is well-encoded in the states of the maze [21] .

2.2.2 Dynamical System Model

We consider networks that can be described as dynamical systems of the affine form:

$$\dot{\mathrm{s}}(t) = f_{\mathbf{K}}(\mathrm{s}(t)) + \mathrm{u}(t) \tag{2.3}$$

where $f(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ denotes the vector field, which is in this case parameterized by $\mathbf{K}$. Here, $\mathrm{s}(t)$ and $\mathrm{u}(t)$ are the states and inputs at time t, respectively. We discretize the system

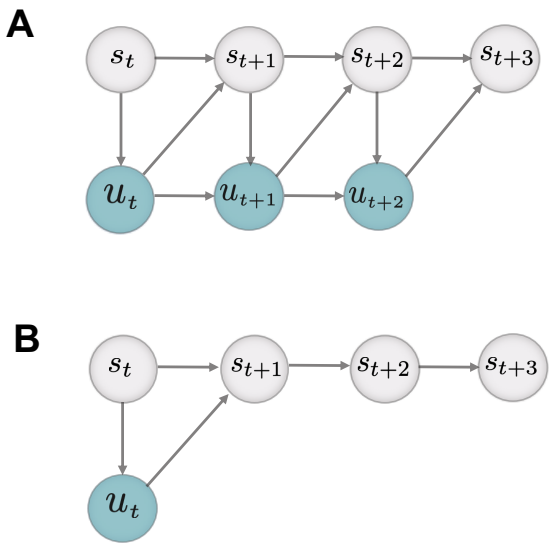

Figure 2.2: Panel A shows the evolution of states over 3 sequence of inputs. Panel B presents our proposed method of increasing the horizon of empowerment calculation.

using a fixed time-step dt, leading to:

$$\mathrm{s}_{t+1} = \mathrm{s}_t + (f_{\mathbf{K}}(\mathrm{s}_t) + \mathrm{u}_t)\, dt. \tag{2.4}$$

2.2.3 Simplified Empowerment Calculation via Impulse Response

The discrete model in equation (2.4) is a one-step autoregressive equation that describes the effect of an input at the current time on the state of the system at the subsequent time. When characterizing empowerment it is desirable to capture the effect of input actions over a prolonged temporal horizon [46, 47, 48]. In such a procedure, one obtains an optimal input sequence leading to a trajectory within the state space (see, e.g., Fig 2.2.A where a sequence of 3 inputs evolves the state according to (2.4)).

Because of the dynamical nature of the system, we posit a simplification of this idea wherein we instead calculate empowerment based on the system impulse response. We consider only a single step input $\mathrm{u}_t$at time t, then allow the states to evolve over a horizon of n time steps i.e. s_{t+n} in (2.5).This constrains the search space (to a single input step), while nonetheless allowing the system dynamics to evolve the state over larger swaths of the state space. That is, we provide the dynamical system with enough time to nontrivially react to the input. Comparing the graphs in Fig. 2.2 A with B, we can see that using the scheme in panel A, the computational complexity of calculating empowerment is linearly proportional to the number of inputs. However, using the impulse response simplification, we can reduce the computational complexity to a constant with respect to a single input.

For n time steps, we can obtain the final state as follows:

$$\mathrm{s}_{t+n} = \mathrm{s}_t + f_{\mathbf{K}}(...f_{\mathbf{K}}((f_{\mathbf{K}}(\mathrm{s}_t) + \mathrm{u}_t)))dt \tag{2.5}$$

where the number of recursions of $f_{\mathbf{K}}$ is n. To make the notion of empowerment meaningful, it is necessary to introduce stochasticity to the above dynamics (in order to create nontrivial probability distributions). We do this by assuming additive noise in the readout of the system states, i.e.,

$$\mathbf{s}_t = \mathrm{s}_t + \mathbf{w}_t, \tag{2.6}$$

where $\mathbf{w}_t$ is an uncorrelated noise process.

2.2.4 Empowerment Maximization for Synthesizing Optimal Network Dynamics

System Parameterization

Thus far, we have set up the problem of calculating the empowerment of a dynamical system, as per equation (2.2). Our goal at this point is to treat the question of *maximizing* the empowerment of the system at hand with respect to its parameterization, i.e. $\mathbf{K}$. In particular, suppose that $\mathbf{K}$ in equation (2.5) represents a degree of freedom to alter the dynamics (e.g., in the case of a pendulum, altering the mass or length of the rod). We denote the number of discretized states of our considered system as M. For a given current state of the system, i.e. $\mathrm{s}_t^{(m)} \in \{\mathrm{s}_t^{(1)}, ..., \mathrm{s}_t^{(M)}\}$, we posit the problem of empowerment maximization with respect to the parameterization $\mathbf{K}$ as follows:

$$\max_{\mathbf{K}} \mathcal{E}(\mathrm{s}_t) = \max_{\mathbf{K}} \max_{\omega} \frac{1}{M} \sum_{m} \mathcal{I}(\mathbf{u}_t; \mathbf{s}_{t+n} | \mathrm{s}_t^{(m)}) \tag{2.7}$$

Variational Empowerment Maximization

The fundamental algorithm to obtain the channel capacity (here, empowerment) is the Blahut-Arimoto (BA) algorithm [49, 50], which is an enumeration-based approach with exponential computational complexity. Thus, it cannot be applied for evaluating mutual information over the continuous domain of variables. However, the intensive computational complexity of empowerment calculation necessitates devising an effective method for its approximation. The authors in [46] proposed variational inference method as a scalable approach for empowerment approximation with the aim of using empowerment as a proxy

for interinsically-motivated reinforcement learning. Similarly, the authors in [22] addressed the empowerment approximation over continuous systems for learning optimal policies via empowerment maximization.

The problem (2.7) is challenging since it involves nested maximization over a continuous state space and nontrivial probability distributions. *Prima facie*, this is near intractable. Thus, in order to proceed, an approximation is required and for this we turn to the popular approach of using a variational lower bound for the key quantities [22, 46, 51, 52]. We will specifically adopt the variational lower bound for empowerment approximation in [22], which enables evaluation of mutual information for continuous variables. However, we will use this computational approach in a different context (i.e., to synthesize dynamics) and with an additional level of approximation to aid tractability. We proceed to discuss these contributions.

In particular, the major computational challenge alluded to above arises from the intractability of the probability terms in equation (2.1) and the integration over the continuous domain of all inputs and states. To circumvent the mentioned issues, we can rewrite equation (2.1) as:

$$\mathcal{I}(\mathbf{u}; \mathbf{s}_{t+n}|\mathrm{s}_t) = \int \int p(\mathrm{s}_{t+n}|\mathrm{u}_t, \mathrm{s}_t)\, \omega(\mathrm{u}_t|\mathrm{s}_t) \log \frac{p(\mathrm{u}_t|\mathrm{s}_{t+n}, \mathrm{s}_t)}{\omega(\mathrm{u}_t|\mathrm{s}_t)}\, d\mathrm{u}_t\, d\mathrm{s}_{t+n} \tag{2.8}$$

where $p(\mathrm{u}_t|\mathrm{s}_{t+n})$ is the *posterior distribution* of inputss. In a Bayesian sense, the input distribution, $\omega(\mathrm{u}_t|\mathrm{s}_t)$, is the prior.

Due to the intractability of the true posterior distribution, we use the mutual information variational bound, introduced in [25], to approximate the above equation as follows:

$$\hat{\mathcal{I}}(\mathbf{s}_{t+n}; \mathbf{u}_t|\mathrm{s}_t) = \int \int p(\mathrm{s}_{t+n}, \mathrm{u}_t|\mathrm{s}_t) \log \frac{q(\mathrm{u}_t|\mathrm{s}_{t+n}, \mathrm{s}_t)}{\omega(\mathrm{u}_t|\mathrm{s}_t)}\, d\mathrm{u}_t\, d\mathrm{s}_{t+n} \tag{2.9}$$

where $q(\mathrm{u}_t|\mathrm{s}_{t+n}, \mathrm{s}_t)$ is the *variational distribution* that approximates the true posterior. Using the variational approximation method, obtaining the variational distribution can be considered as an optimization problem where $q_\xi(\mathrm{u}_t|\mathrm{s}_{t+n}, \mathrm{s}_t)$ is a variational family of distributions with parameter ξ [53]. Given that the variational distribution expressively represents the true posterior distribution, a tight variational lower bound can be achieved [22]. That is

$$\begin{aligned}
\mathcal{I} - \hat{\mathcal{I}} &= \int\int p(\mathrm{s}_{t+n}, \mathrm{u}_t|\mathrm{s}_t) \log \frac{p(\mathrm{u}_t|\mathrm{s}_{t+n}, \mathrm{s}_t)}{q_\xi(\mathrm{u}_t|\mathrm{s}_{t+n}, \mathrm{s}_t)} \, d\mathrm{u}_t \, d\mathrm{s}_{t+n} \\
&= \int p(\mathrm{s}_{t+n}|\mathrm{s}_t) \left[\mathrm{KL}(\, p(\mathrm{u}_t|\mathrm{s}_{t+n}, \mathrm{s}_t) \,||\, q_\xi(\mathrm{u}_t|\mathrm{s}_{t+n}, \mathrm{s}_t))\right] d\mathrm{s}_{t+n} \\
&= \mathbb{E}_{\mathrm{s}_{t+n} \sim p(\mathrm{s}_{t+n}|\mathrm{s}_t)}[\mathrm{KL}(\, p(\mathrm{u}_t|\mathrm{s}_{t+n}, \mathrm{s}_t) \,||\, q_\xi(\mathrm{u}_t|\mathrm{s}_{t+n}, \mathrm{s}_t))]
\end{aligned} \tag{2.10}$$

where KL denotes the KL-divergence. If the variational distribution is similar to the true posterior distribution, the KL-divergence is close to zero. Fig. 2.3 shows the schematic of variational approximation.

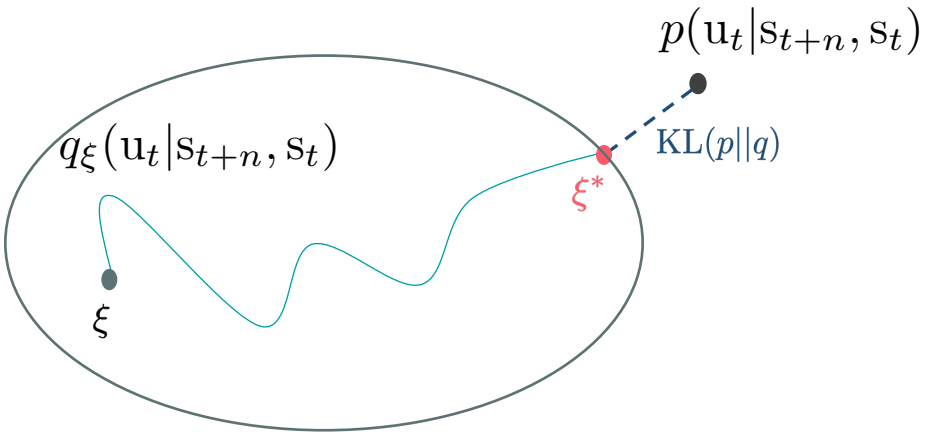

Figure 2.3: Schematic of Variational Approximation

Hence, we can obtain a variational lower bound on the empowerment as follow:

$$\hat{\mathcal{E}}(\mathrm{s}_t) = \max_{\omega,q} \hat{\mathcal{I}}(\mathbf{s}_{t+n}; \mathbf{u}_t|\mathrm{s}_t) \tag{2.11}$$

To perform the optimization of the variational bound, we can obtain the input distribution and the variational distribution via deep neural networks parameterized by ϕ and ξ, respectively. Sharing the same ideas with amortized variational inference [54, 55], using neural networks(NNs) as a mapping from states to the distribution parameters enables us to only deal with finite number of neural network parameters (i.e. weights and biases) instead of learning a separate input distribution and variational distribution for each state. In this work, we choose the input and variational distributions from the Gaussian family as follows:

$$\begin{aligned} \omega(\mathrm{u}_t|\mathrm{s}_t^{(m)}) &= \mathcal{N}(\mu_\phi(\mathrm{s}_t^{(m)}), \sigma_\phi^2(\mathrm{s}_t^{(m)})I) \\ q(\mathrm{u}_t|\mathrm{s}_t^{(m)}, \mathrm{s}_{t+n}) &= \mathcal{N}(\mu_\xi(\mathrm{s}_t^{(m)}, \mathrm{s}_{t+n}), \sigma_\xi^2(\mathrm{s}_t^{(m)}, \mathrm{s}_{t+n})I) \end{aligned} \tag{2.12}$$

where mean μ and variance σ are also parameterized by NNs. $\theta = \{\phi, \xi\}$ are the joint parameter set of equation (2.11). Via joint optimization of variational bound w.r.t. parameters of ω and q, we can achieve the action distribution that maximizes the mutual information and the variational distribution that is responsible for the tightness of the variational lower bound. Exploiting the variational lower bound on empowerment, we pose our optimization problem as follows:

$$\max_{\mathbf{K}} \max_{\theta} J(\mathbf{K}, \theta), \tag{2.13}$$

where

$$J(\mathrm{K}, \theta) = \frac{1}{M} \sum_{m} \hat{\mathcal{I}}(\mathbf{s}_{t+n}; \mathbf{u}_t | \mathrm{s}_t^{(m)}) \tag{2.14}$$

and

$$\hat{\mathcal{I}}(\mathbf{s}_{t+n}; \mathbf{u}_t | \mathrm{s}_t^{(m)}) = \tag{2.15}$$

$$\mathbb{E}_{\mathrm{s}_{t+n} \sim p(\mathrm{s}_{t+n}, \mathrm{u}_t | \mathrm{s}_t^{(m)})} [\log q(\mathrm{u}_t | \mathrm{s}_{t+n}, \mathrm{s}_t^{(m)}) - \log \omega(\mathrm{u}_t | \mathrm{s}_t^{(m)})]$$

We can perform the joint Monte-Carlo sampling method to obtain samples from the joint distribution, $p(\mathrm{s}_{t+1}, \mathrm{u}_t | \mathrm{s}_t^{(m)})$, and use the reparameterization trick [56, 57] to evaluate the stochastic gradients of the objective function with respect to $\mathbf{K}$ and θ as:

$$\begin{gathered} \frac{\partial}{\partial \theta} \hat{\mathcal{I}}(\mathbf{s}_{t+n}; \mathbf{u}_t | \mathrm{s}_t^{(m)}) \approx \\ \frac{1}{L} \sum_l \frac{\partial}{\partial \theta} [\log q_\xi(\mathrm{u}_t^{(l)} | \mathrm{s}_{t+n}^{(l)}, \mathrm{s}_t^{(m)}) - \log \omega_\phi(\mathrm{u}_t^{(l)} | \mathrm{s}_t^{(m)})] \end{gathered} \tag{2.16}$$

and

$$\begin{gathered} \frac{\partial}{\partial \mathbf{K}} \hat{\mathcal{I}}(\mathbf{s}_{t+n}; \mathbf{u}_t | \mathrm{s}_t^{(m)}) \approx \\ \frac{1}{L} \sum_l \frac{\partial}{\partial \mathbf{K}} [\log q(\mathrm{u}_t^{(l)} | \mathrm{s}_{t+n}^{(l)}, \mathrm{s}_t^{(m)}) - \log \omega(\mathrm{u}_t^{(l)} | \mathrm{s}_t^{(m)})] \end{gathered} \tag{2.17}$$

where L is the number of samples. Algorithm 1 summarizes the optimization procedure. It is worth mentioning that if f_K is differentiable, a gradient-based approach can be used to update the decision variables. When examining convergence, we look at the magnitude of the gradients with respect to the decision variables and the magnitude of the cost function, J. However, we do not set any formal thresholds on these quantities, since they depend on the dynamics and parameters of the system under consideration.

Algorithm 1 Maximization of Empowerment Variational Lower Bound, w.r.t. $\theta = \{\phi, \xi\}$ and $\mathbf{K}$

Initialize uniform samples from state space, $\{\mathrm{s}_t^{(m)}\}_{m=1,\dots,M}$
repeat
for each $\mathrm{s}_t^{(m)}$ **do**
    draw one sample from $\omega(\mathrm{u}_t|\mathrm{s}_t^{(m)})$:
    $\mathrm{u}_t \sim \omega(\mathrm{u}_t|\mathrm{s}_t^{(m)})$
    transit to the final state s_{t+n}
end for
$\mathrm{J} = \frac{1}{M}\sum_m \log q(\mathrm{u}_t|\mathrm{s}_t^{(m)}, \mathrm{s}_{t+n}) - \log \omega(\mathrm{u}_t|\mathrm{s}_t^{(m)})$
$\theta \leftarrow \theta + \eta_\theta \nabla_\theta J$ for r epoches
$\mathbf{K} \leftarrow \mathbf{K} + \eta_{\boldsymbol{K}} \nabla_{\mathbf{K}} J$
until convergence

2.3 Results

We proceed to show the efficacy of the proposed method on three canonical examples: linear dynamical systems, the inverted pendulum on fixed pivot and the Wilson-Cowan neural mass model. We then discuss the emergent dynamics of these systems following empowerment maximization.

2.3.1 Efficacy of the Proposed Impulse Response Procedure

First, we demonstrate the efficacy of the proposed empowerment calculation procedure.

Linear system

For the 2-dimensional linear systems, equation (2.5) is:

$$\mathbf{s}_{t+n} = \mathrm{s}_{t+n-1} + (\mathbf{A}\mathrm{s}_{t+n-1} + \mathrm{u}_t)dt + \mathbf{w}_{t+n}, \tag{2.18}$$

where $\mathbf{A} \in \mathbb{R}^{2\times 2}$ determines the vector field of the system. In this example, $\mathbf{A}$ is chosen so that the origin is unstable. Fig. 2.4 (b) and (c), depict the computed empowerment over the state space for horizons of 2 and 5 steps, respectively. As it is seen, the states around the origin have largest empowerment, which is intuitive given the dynamics of the system (because the origin is unstable, inputs applied for these initial conditions have the potential to access the largest swaths of state space).

As means of comparison, we have also performed calculation of the empowerment using a full iteration of actions (i.e., as in Fig. 2.2A). Fig. 2.4 (d) shows the obtained empowerment landscape for this case. As seen, the proposed impulse response-based approach compares favorably to the full solution. Full details regarding simulations are found in Appendix A.

Pendulum

To show the performance of our method for nonlinear systems, we studied the simple pendulum with the following dynamics

$$\ddot{s} = -\tfrac{3g}{2l}\sin(s+\pi) + \tfrac{3}{ml^2}(u - 0.05\dot{s}) \tag{2.19}$$

For the sake of comparison with previous works [22, 47], we considered the optimization over $n = 50$ steps. As seen in Fig. 2.5, the impulse response method produced an interpretable empowerment landscape, wherein the locus of high empowerment corresponds with the unstable manifold associated with the equilibrium at the origin (i.e., the pendulum in the inverted position). Like the unstable linear system, this is intuitive because on this manifold the input is able to drive the system to a larger swath of the state space. This is exactly equivalent to the idea of reachability in control systems analysis.

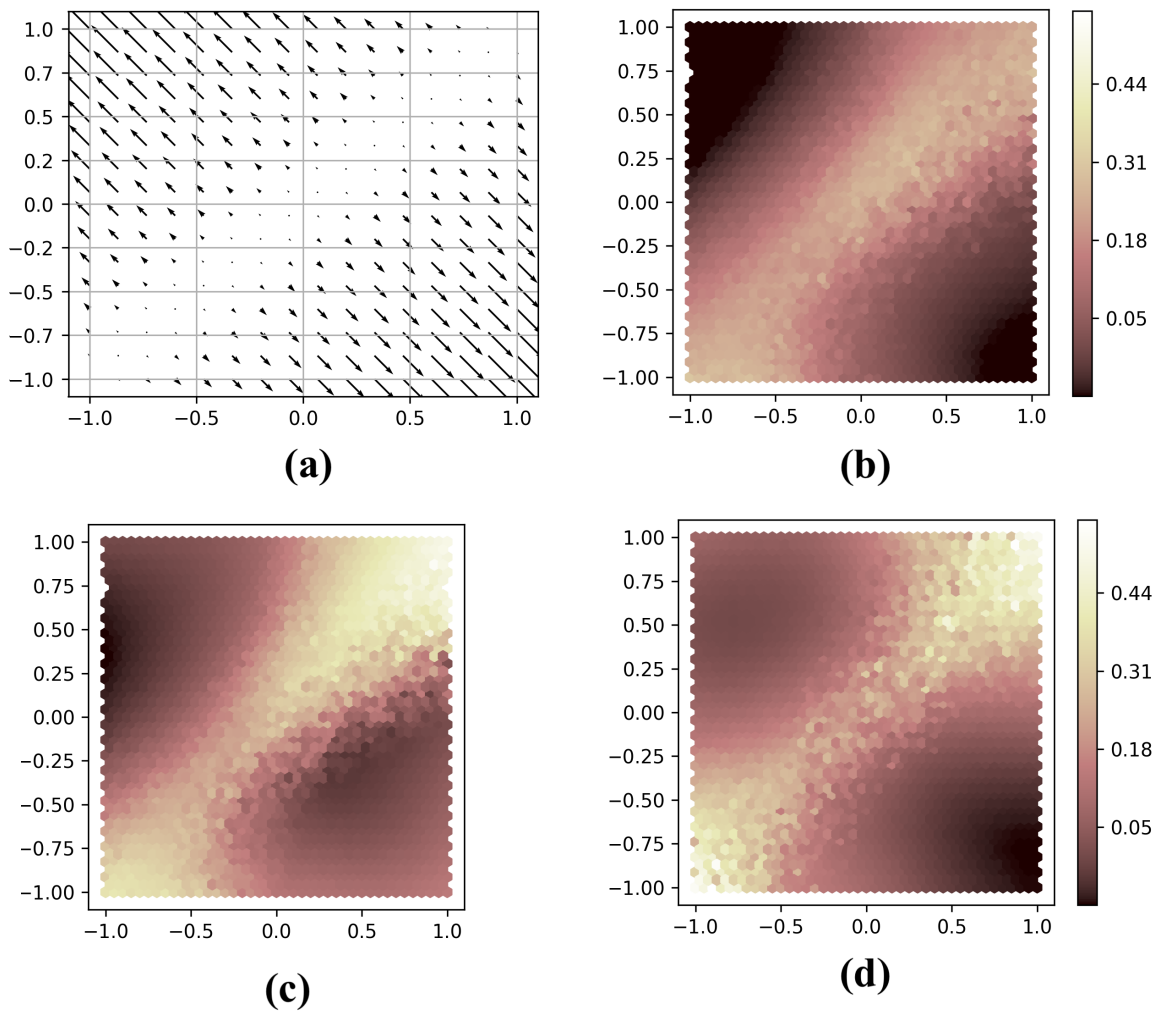

Figure 2.4: The vector field and empowerment landscape result from n step empowerment calculation. Plot (a) shows the state space and its corresponding vector field of a 2 dimensional linear system and (b) depicts the corresponding empowerment landscape (in *nats*) optimized over two time steps of states. Plot (c) and (d) depict the comparison of empowerment (in *nats*) landscape over 5 time steps of states (i.e., the proposed impulse response method) versus 5 time steps of actions, respectively.

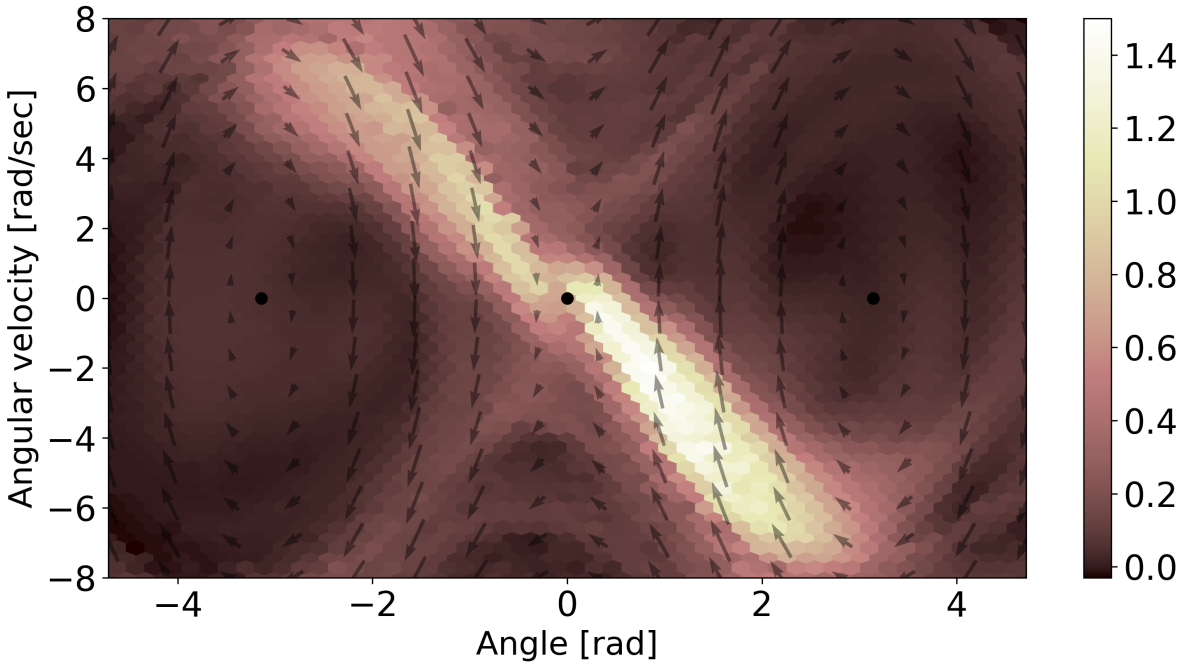

Figure 2.5: The pendulum vector field and its corresponding empowerment landscape (in *nats*) obtained via empowerment calculation over 50 step states. Fixed points are shown with black dots and arrows represent the direction of the vector field.

The Wilson-Cowan model

We also demonstrate the performance of our method for the canonical neural mass model, the Wilson-Cowan model, which provides a course-grained representation of the overall activity of populations of neurons [58, 59, 60]. The 2-dimensional system describing the excitatory and inhibitory activity of these populations is:

$$\begin{aligned} \dot{s}_e &= -s_e + (1 - r_e s_e)\mathcal{F}_e(c_1 s_e - c_2 s_i + I_e + P) \\ \dot{s}_i &= -s_i + (1 - r_i s_i)\mathcal{F}_i(c_3 s_e - c_4 s_i + I_i + Q) \end{aligned} \tag{2.20}$$

where s_e and s_i are the overall activity in the excitatory and inhibitory populations. Here, r_j, $j \in \{i, e\}$ represents a constant describing the refractory period. c_1, c_2, c_3 and c_4 present the strength of excitatory and inhibitory interactions. P and Q are the excitation level in the system and I_e and I_i are the control input currents that affect the respective populations. Also, $\mathcal{F}_j$ is a sigmoid function:

$$\mathcal{F}_j(s) = \frac{1}{1 + \exp[-a_j(s - \theta_j)]} - \frac{1}{1 + \exp(a_j\theta_j)} \tag{2.21}$$

where a and θ are free parameters representing the slope and the threshold, respectively. We applied our method to the Wilson-Cowan model over $n = 100$ steps. Particularly, we study the case wherein the system exhibits 3 multiple fixed points (2 stable fixed points and an unstable fixed point, see Appendix A for details). Fig. 2.6, depicts the empowerment landscape obtained using the impulse response method. As seen, the locus of high empowerment corresponds to the unstable manifold associated with the unstable fixed point.

2.3.2 Synthesizing Optimal Dynamics

We now move to the main problem considered in this paper: optimization of the empowerment with respect to thedynamics. We performed this study for the same systems considered above.

Linear and nonlinear systems

For simplicity, we first consider a horizon of two steps and a linear parameterization as follows:

$$\mathbf{s}_{t+2} = \mathrm{s}_t + f(f(\mathrm{s}_t) + \mathbf{K}\mathrm{s}_t + \mathrm{u}_t)dt + \mathbf{w}_{t+2} \tag{2.22}$$

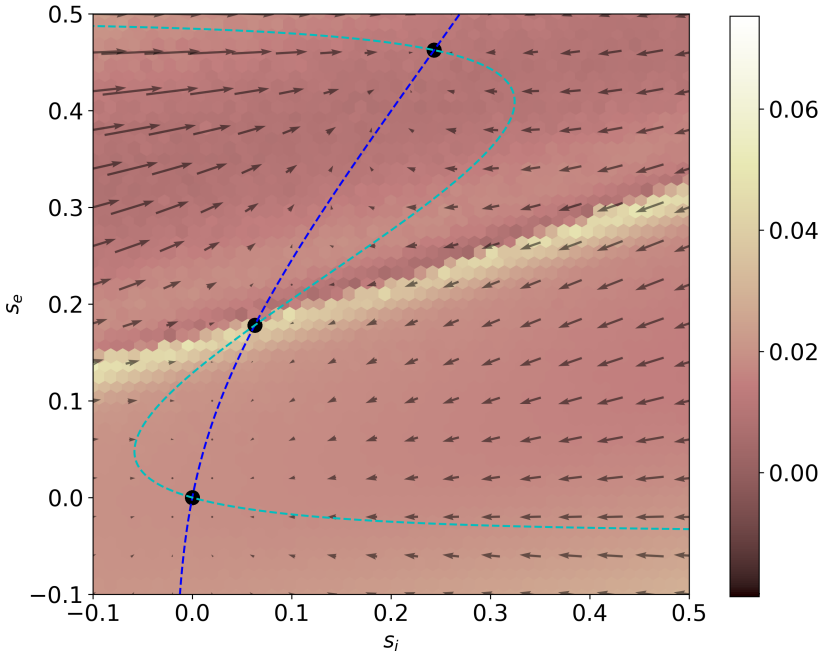

Figure 2.6: The vector field, nullclines and empowerment landscape result from n step empowerment calculation for the Wilson-Cowan model (the empowerment values are in *nats*). The fixed points of the system are located where nullclines intersect. Fixed points are shown with dots.

Fig. 2.7, shows the vector field and the landscape of a stable linear system before and after optimization of the linear parameterization (again, see Appendix A for details). Perhaps intuitively, the optimization has taken the original system (wherein the origin is asymptotically stable) and destabilized it. As a consequence, the system is more 'usable' in the sense that a greater diversity of states is accessible (asymptotically).

The optimal system in Fig. 2.7-(b) depicts higher empowerment around the center as opposed to Fig. 2.7-(a).

We also consider the case of a nonlinear system with multiple equilibria (again, see Appendix A for details). As shown, the optimization tends to accentuate the dominance of a single

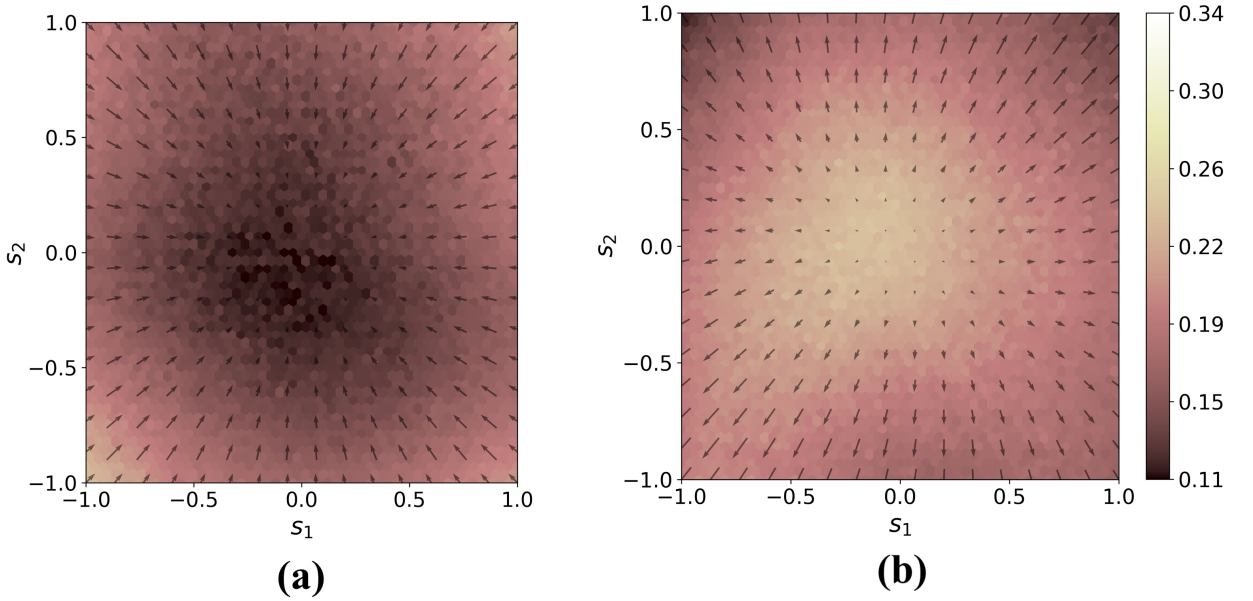

Figure 2.7: Empowerment maximization of a linear dynamical system with one stable equilibrium point (a). After optimization, the resultant environment (b) exhibits a dominant unstable equilibrium.

unstable equilibrium point. In Fig. 2.8-(a), the basal system has 4 equilibria (2 saddle, one stable and one unstable equilibrium). The optimization leaves a dominant unstable equilibrium and destroys the rest of them, Fig. 2.8-(b).

These results are intuitive insofar as the optimization appears to be simplifying the systesm dynamics and making as much of the state space to be accessible or reachable as possible. Thus, an input interacting with the optimized dynamics can 'do more' than one interacting with the basal system. Although we have considered only the simplified case of a linear parameterization, there are clear ways to generalize this concept including parameterizing the dynamics along a basis set of nonlinear functions.

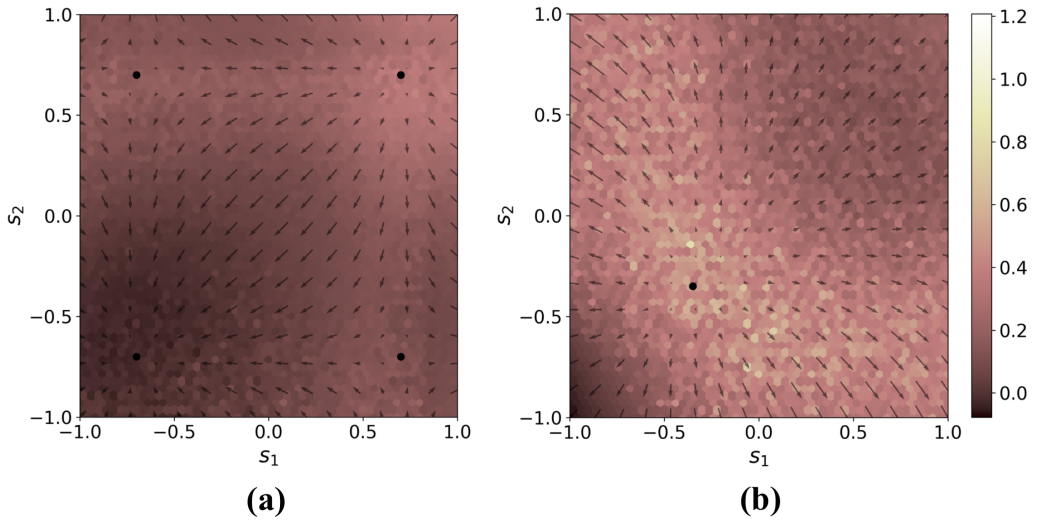

Figure 2.8: Empowerment maximization of a nonlinear dynamical system with four equilibrium points (a). After optimization, the number of equilibria is reduced and a single dominant unstable node emerges (b).

The Wilson-Cowan model

We carry out further simulations for the Wilson-Cowan model to look into the sorts of dynamics that emerge after learning the optimal parametrization of the system. In Fig. 2.9, we have considered P in equation (2.20) as the system parametrization and performed the empowerment optimization. Fig. 2.9 (a) and (b) depict the corresponding empowerment landscape of the initial and the optimal network. As seen, the emergent dynamics from empowerment maximization for $n = 300$ create a limit cycle. This observation again is intuitive since the limit cycle permits different initial conditions to remain 'separated' asymptotically (versus converging to a single stable fixed point). This observation is also intriguing from a scientific perspective, since such limit cycle oscillations are thought to be pervasive in actual neuronal dynamics.

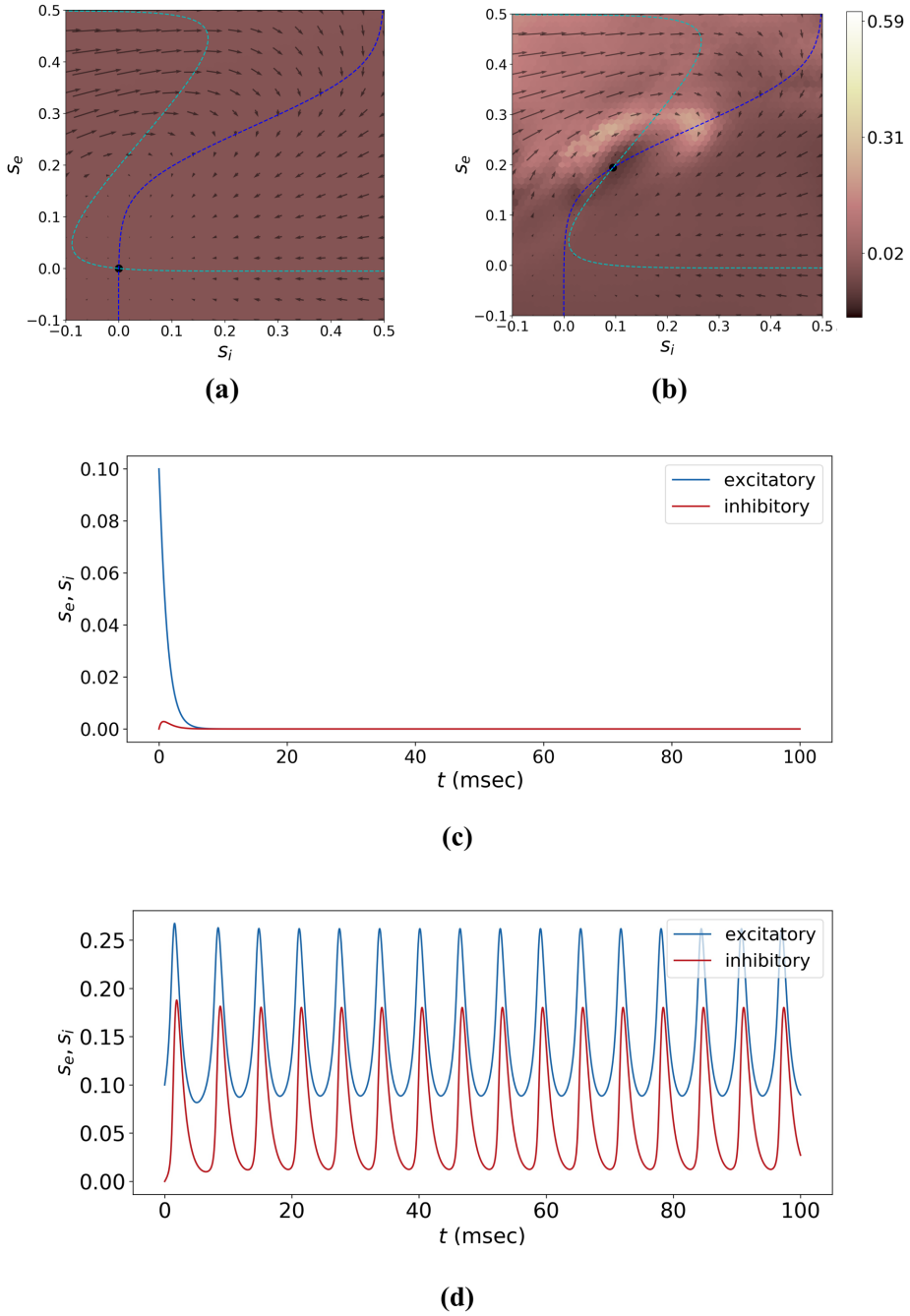

Figure 2.9: Comparison of the empowerment landscapes and the system dynamics for empowerment maximization in the Wilson-Cowan model. In (a), the system initially exhibits a stable fixed point and the empowerment landscape is flat ($P = 0$). In (b), after learning the optimal system parametrization, i.e. P, the system exhibits a limit cycle ($P = 1.177$). (c) and (d) present the comparison between the time response of the system before and after empowerment maximization.

2.4 Discussion and Conclusion

In our work, we have explored the use of empowerment as a way to shape a dynamical system so as to render it more functional for information processing. From the perspective of dynamical systems and control theory, this problem amounts to altering the dynamics of the system at hand so that as much of the state space is reachable as possible. To do so, we suggest a computational simplification to aid in the calculation of empowerment (noting that the calculation of empowerment is itself an optimization problem over the space of inputs). We then use a variational approach to facilitate the optimization of the system as intended.

Returning to the idea alluded to in the Introduction, one possible avenue for these results is to enable a study of biophysical neuronal dynamics. That is, we can fashion neurons as environments and study how their dynamics mediate information capacity. This may help us to derive general insights into the functionality of neural circuits and also reveal principles for network design that do not require task specificity. Finally, one can connect these results back to the general idea of agent policy design, by viewing the parameterization as a degree of freedom that can be chosen by the agent, so that it can both shape the environment and then exploit it according to a secondary objective. It is also worth mentioning that in our simulations, we constrained the degree of freedom as a linear parametrization of the environment; a potential future direction is augmenting the environment according to a more general parameterization, for example through the use of basis functions that allow for nonlinear manipulation of the vector field.

Chapter 3

Optimization and Learning of Network Dynamics via Reinforced Regression for High-level Function

3.1 Introduction

Neural-network-based machine learning provides an interesting and potentially insightful paradigm for us to build intuition for the nature of computations and dynamical mechanisms underlying various patterns of neural activities. Initially, such networks were simply conceived of as abstract constructs that reflected certain aspects of neurobiology, but that were mostly focused on achieving computational and engineering endpoints. However, recently scientists have begun to derive intellectual 'feedback' from artificial neural networks by using them as an analog to probe and study actual brain dynamics in top-down paradigms [18].

Perhaps most notably, artificial neural networks have emerged as a promising tool for modeling sensory systems, attention, circuit connectivity and patterns of neural activity [18, 30, 61, 62]. As such, various architectures (e.g. feedforward versus recurrent neural networks), optimization techniques and regularization methods can be explored to generate mechanistic hypothesis about the computations carried out in neural systems.

In this chapter, we deploy the top-down modeling approach by optimizing RNNs as a proxy of dynamical network systems that carry out computations that are ostensibly associated with high-level cognitive functions. In the following, we first lay the groundwork for the use of RNN modeling. Next, we provide the details of a new reinforced regression method for optimization of RNNs.

3.2 Artificial Recurrent Neural Networks

3.2.1 Motivations for Using a Recurrent Architecture

To derive an understanding of the generative processes that give rise to task-evoked neural activity patterns, we turn to a dynamical systems framework. At a high-level, these processes are captured through an equation of the form $\dot{x}(t) = F(x(t), u(t))$, where $x(t)$ is a high-dimensional vector that captures the state (e.g., activity) of neurons and $u(t)$ are exogenous inputs (e.g., stimuli). In question is the vector field $F(.)$, which models the dynamics, i.e., time-evolution, of the neural states during both stimulus-driven and autonomous periods of a given task. Unfortunately, ascertaining $F(.)$ in a bottom-up fashion is not straightforward, especially at a level of scale commensurate with networks thought to be relevant to high-level brain functions, such as prefrontal cortex [63, 64].

One approach to tackling this issue is to produce top-down hypothetical versions of $F(.)$ by optimizing artificial RNNs to engage high-level cognitive tasks [28, 31, 65, 66]. Here, rather than modeling the dynamics of the brain directly (e.g., by fitting to neural recordings), one is creating and optimizing a model that achieves the function in question, such that $F(\cdot)$ is *emergent* [29, 67]. Specifically, the *recurrent* nature of RNNs architecture develops putative neural dynamics that embed the computations associated with the function in question.

3.2.2 Overview of Training Methods of RNNs

The use of artificial RNNs in this context dates to the early work of Hopfield, wherein abstract networks were among were used to describe potential mechanisms underlying associative long-term memory [68]. Methods for training recurrent networks such as Real-Time Recurrent Learning (RTRL)[69] and Back-Propagation Through Time (BPTT) [70] were developed later and provided a means to construct more complex network architectures and dynamics. However, such training methods were primarily developed as means to create algorithmic solutions, as opposed to top-down neuroscience tools.

One of the first efforts explicitly bridge the gap between machine learning and computational neuroscience was in the area of reservoir computing [71], where recurrent neural networks were shown to be able to engage with a number of tasks of potential cognitive relevance and proposed as a means by which the brain itself might be functioning. Instances of this line of research include Liquid State Machines (LSM) [72] and Echo State Networks (ESN) [73]. Even more recently, many new methods have been proposed to optimize RNNs for the specific purpose of top-down neuroscience. These include the first-order reduced and controlled error (FORCE) [74] method, Hessian-free (HF) [67, 75] and variations on stochastic gradient descent (SGD) [31].

3.3 Reinforced Regression for Optimizing RNNs to Perform Cognitive Tasks

A widely-used and well-studied class of recurrent networks is characterized by the presence of fully random recurrent connectivity [76]. Specifically, these networks have rich dynamical repertoire and thus are capable of generating complex temporal patterns that are potentially commensurate with spontaneous cortical activities [76, 77]. In the following, we elaborate on constructing and optimizing random RNNs to perform high-level task-based functions.

3.3.1 Random Recurrent Network Model

We consider networks composed of N nonlinear firing-rate units :

$$\tau\,\dot{\mathbf{x}}(t) = -\mathbf{x}(t) + \mathbf{J}\mathbf{r}(t) \tag{3.1}$$

where $\mathbf{x} \in \mathbb{R}^N$ is the state vector and $\mathbf{r}\ (t) = \tanh(\mathbf{x}(t))$ denotes the activities obtained via applying hyperbolic nonlinearity to network states. We set the network time constant, τ, to 1 for simplicity throughout the rest of equations. We denote $\mathbf{J}$ as the initial synaptic connectivity matrix with elements drawn randomly from a Gaussian distribution, i.e. $\mathbf{J}_{ij} \sim \mathcal{N}(0, \sigma_J^2)$. Specifically, we parameterize $\sigma_J^2 = \dfrac{g^2}{N}$, so that g controls the strength of initial synaptic interactions. A schematic of these networks is presented in Fig. 3.1.

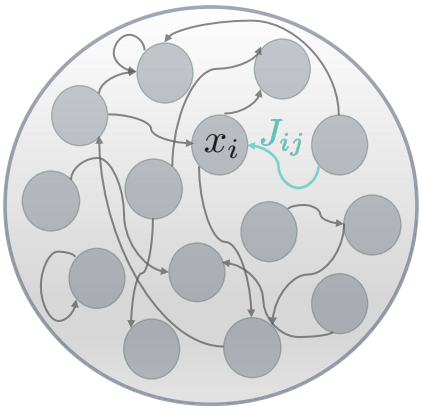

Figure 3.1: Schematic of random recurrent network

3.3.2 Basic Dynamical Characterizations of the Untrained Network

In the context of our investigation, it is important to keep in mind that the above network is in fact a dynamical system. Our eventual goal is to understand the dynamics embedded in the network vector field, including the type and stability of attractors. Certain details are useful to point out even prior to any optimization being carried out. For instance, the origin is a fixed-point of (3.1). To study the stability properties of the origin, we can consider the eigenvalues of the Jacobian matrix:

$$\mathbf{M} = -\mathbf{I} + \mathbf{J} \tag{3.2}$$

Regarding the well-known circle law of random matrix [78], the eigenvalues of random connectivity matrix $\mathbf{J}$ are distributed over a disk with radius g for $N \to \infty$ [76, 79]. Thus, depending on the strength of $\mathbf{J}$, the stability of origin varies. By shifting the stability matrix $\mathbf{M}$ by $\mathbf{I}$, we obtain the connectivity matrix as the shifted Jacobian at the origin. To perform

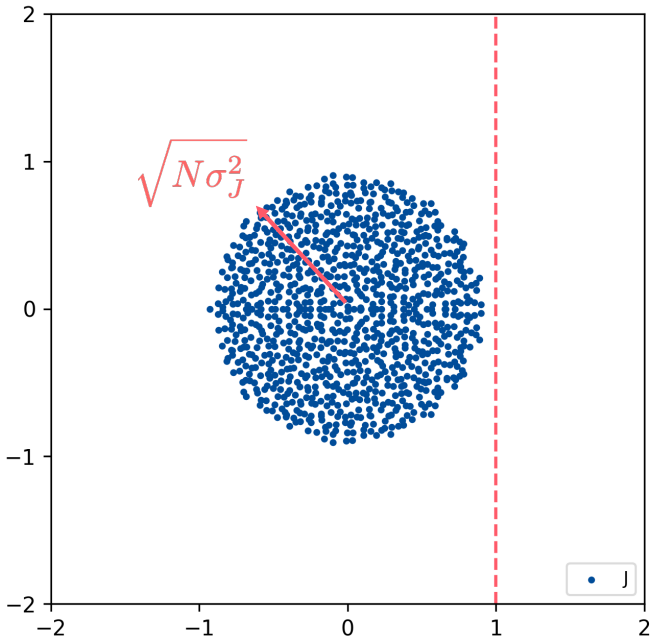

Figure 3.2: Eigenvalue spectrum of the connectivity matrix

the stability analysis, we consider the *shifted* Jacobian matrix throughout this chapter. Fig. 3.2 shows the eigenvalue distribution of the connectivity matrix, **J**:

Therefore, depending on the strength of **J** , the stability of origin varies. For $g < 1$ the origin is asymptotically stable, while for $g > 1$ the origin is unstable, suggestive of potentially chaotic dynamics in the overall network. Fig. 3.3 presents these two dynamical regimes.

3.3.3 Optimizing Networks through Structure Low-rank Augmentation

Despite the dynamic range of these networks, they cannot implement complex computations without specific optimization in terms of a functional objective [80]. Motivated by the

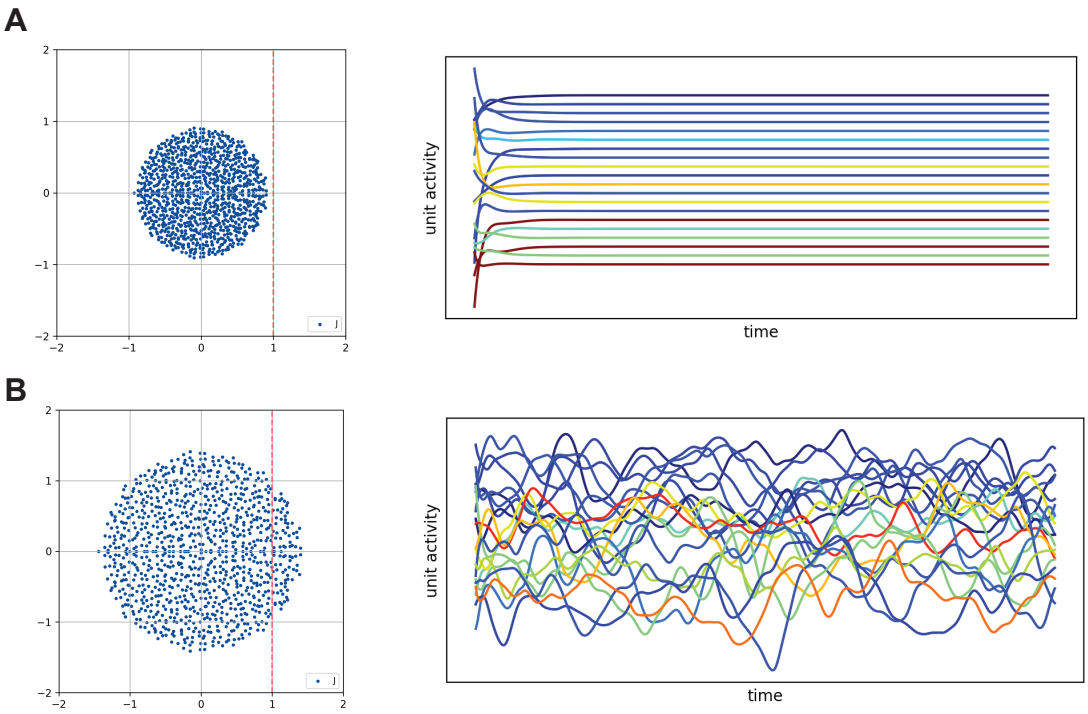

Figure 3.3: Eigenvalue spectrum and neural activities. Panel A, for $g = 0.9$, shows the eigenvalue distribution and unit activities (for a subset of neurons/units in the RNN). Panel B, for $g = 1.5$, shows the eigenvalue distribution and unit activities.

evidence that connectivity of cortical circuits is a mixture of random and structured synaptic connections and low-dimensional dynamics observed in neural populations [81], one can add low-rank structure to the random connectivity and manipulate the initial dynamics for implementing the cognitive task computations. For example, Fig. 3.4 shows the emergence of outlier eigenvalues after applying a rank 1 structure to the connectivity matrix $\mathbf{J}$. Many of the aforementioned methods for training RNNs, including the FORCE method [74] involve the use of such low-rank augmentations to a basic random connectivity structure.

We will follow a similar paradigm at the FORCE method, but make a few specific modifications with an eye towards increased interpretability relative to the original method. We proceed by

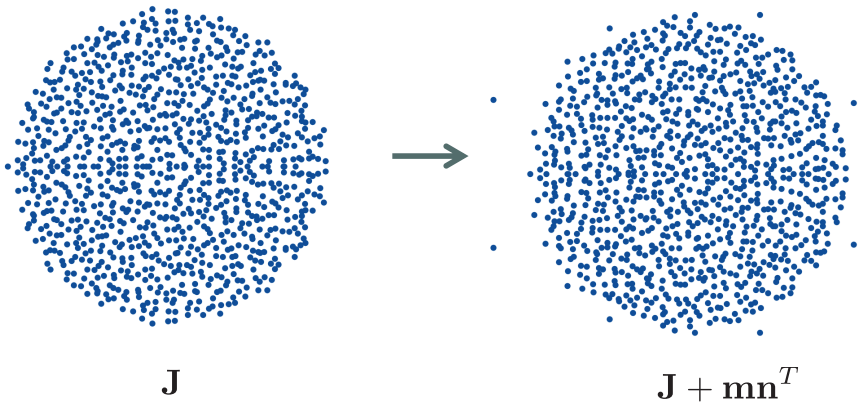

Figure 3.4: Comparison of eigenvalue distributions before and after adding the low-rank structure. ($\mathbf{m}, \mathbf{n} \in \mathbb{R}^N$)

explaining the details of our proposed method for optimizing RNNs to perform a high-level function.

3.3.4 Optimization of RNNs via Temporall Restricted Error Kernel

To parameterize the network (3.1) to learn the computation of interest, we adjust the connectivity via applying additional connections to the network:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= -\mathbf{x}(t) + (\mathbf{J} + \mathbf{W}_f \mathbf{W}_o^T)\, \mathbf{r}(t) + \mathbf{W}_i \mathbf{u}(t) \\ z_o(t) &= \mathbf{W}_o^T\, \mathbf{r}(t) \end{aligned} \tag{3.3}$$

where $z_o(t)$ is a network output. Optimization/learning proceeds by modifying the projection vectors $\mathbf{W}_o \in \mathbb{R}^{N \times 1}$. The network output $z_o(t)$ is fed back to the network via feedback weights

i.e. $\mathbf{W}_f \in \mathbb{R}^{N\times 1}$. The network receives the exogenous input (i.e., stimulus) $\mathbf{u}(t) \in \mathbb{R}^{d\times 1}$ via input weights $\mathbf{W}_i \in \mathbb{R}^{N\times d}$, where d is the input dimension.

This strategy effectively modifies the initial connectivity by addition of a low-rank component, allowing for more interpretable relations between the overall network connectivity and function [77, 82].

In general, the goal of learning is to minimize the total error between a target signal, $f(t)$, and the network output during training time, T.

$$E(t) = \frac{1}{2}\int_0^T e(t)^2 dt \tag{3.4}$$

with $e(t) = z(t) - f(t)$. In our setup, we modify readout weights $\mathbf{W}_o$ to keep the error, $e_o(t)$, between $z_o(t)$ and output targets, $f_o(t)$, small during training.

However, our goal here is not to only optimize/train RNNs to generate accurate input-output mappings. Instead, we contribute our effort to incorporating realistic assumptions and restrictions into our training framework. In particular, behaving animals do not receive a continual external supervisory error-signal throughout learning, but they obtain a reward upon successful performance at brief moments in time (i.e., the high-level notion of reinforcement learning [83]). Inspired by this, the proposed training method blends trial-based reinforcement learning with the continual error regression, such that learning is constrained to occur only during brief intervals throughout training. We refer to these epochs where regression occurs as a temporally restricted error kernel.

Therefore, in contrast to FORCE method, where the weights are updated continually during training intervals, here, we update weights only during short intervals; i.e. during the response generation. Thus, our training paradigm is a temporally regularized FORCE method that

prevents the output from overfitting to the target signal and in turn allows the network to figure out the optimal dynamics on its own. As per the FORCE method, $\mathbf{W}_o$, subject to our training paradigm, is updated using recursive least squares [74]. Hence, to reduce $e_o(t)$, we obtain

$$\begin{aligned} \mathbf{W}_o(t) &= \mathbf{W}_o(t-\Delta t) - e_o(t)\mathbf{P(t)r}(t) \\ \mathbf{P}(t) &= \mathbf{P}(t-\Delta t) - \frac{\mathbf{P}(t-\Delta t)\mathbf{r}(t)\mathbf{r}^T(t)\mathbf{P}(t-\Delta t)}{1+\mathbf{r}^T(t)\mathbf{P}(t-\Delta t)\mathbf{r}(t)} \end{aligned} \tag{3.5}$$

where $\mathbf{P}(t)$ denotes the approximate estimate for the inverse of the correlation matrix of network activities with a regularization term

$$\mathbf{P}(t) = \int_0^T \mathbf{r}(t)\mathbf{r}^T(t)dt + \alpha \boldsymbol{I}_N \tag{3.6}$$

where α is the regularization parameter and $\mathbf{I}_N$ the identity matrix. Mathematical details of obtaining the update rules (C.3) are provided in Appendix B. Fig. 3.5 shows the network architecture and the schematic of the proposed training paradigm in comparison to fully temporally unconstrained methods.

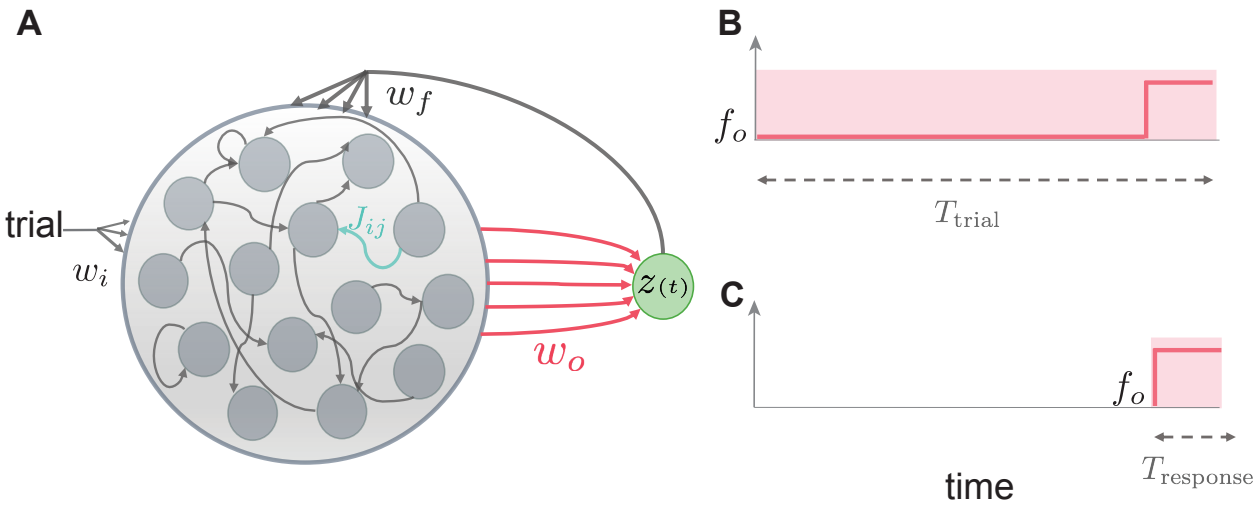

Figure 3.5: Reinforced regression. (A) shows the RNN architecture and connectivity. (B) displays the training paradigm of FORCE method, wherein $z(t)$ follows the target signal $f_o(t)$ during the total trial interval. (C) displays the training paradigm of reinforce regression, wherein $z(t)$ follows the target signal $f_o(t)$ only at brief intervals, i.e. T_{response}.

3.4 Conclusion

In this chapter, we introduced RNN modeling as a tool in top-down neuroscience. We described the basic architecture and dynamic of the class of RNNs that we are interested in using. We then provided details of a reinforced regression training method that adds interpretable structure to existing methods in the literature. Our premise is that this revised optimization framework, by leaving long temporal epochs unconstrained, may lead to a wider range of emergent dynamics within RNNs when optimized for high-level cognitive functions. In the subsequent chapter, we will evaluate the efficacy of this method on a specific paradigm *working memory* wherein we will show how rich and biologically compatible dynamics emerge in the optimized networks.

Chapter 4

Analysis of Emergent Dynamics Associated with Working Memory

4.1 Introduction

Working memory (WM) enables the storage and manipulation of information over brief periods of time. This capability is required for many cognitive tasks such as reasoning and problem-solving. Presumably, memory retention relies on an invariant latent neural representation of past stimuli [84], but the precise nature of these representations and the dynamical mechanisms by which they are created in neural circuits remain enigmatic. From a mechanistic standpoint, there are two prevailing hypotheses that address the bases of invariant neural representation during memory periods; the first hypothesis suggests that neural activity evolves toward stable self-sustained attractor states during delay periods associated with WM [85, 86, 87]. Depending on the nature of memory items and tasks, discrete attractors such as stable fixed points or continuous attractors such as line and ring

attractors have been proposed [86]. Conversely, the latter hypothesis does not necessarily require self-sustaining attractors. One premise for such a hypothesis is that high-dimensional fluctuations in neural activities may project onto a lower-dimensional latent space upon which an invariant representation is held during delay intervals [37, 42]. For example, if the activity of a neuron gradually drops during delay periods, the activity of another neuron increases to compensate for that drop. Thus, during delay, neural activity may traverse along a low-dimensional manifold corresponding to this invariant representation [88, 89].

In this spirit, recent works have tried to reconcile the aforementioned hypotheses in the context of WM [90, 91, 92, 93]. For instance, in [90], authors suggest that the existence of both transient dynamics and stable fixed points are essential for reliable task performance in the face of uncertain stimulus timing. However, there remain open questions yet to be addressed; what are the circumstances under which each of these solutions may emerge? Which mechanism is more favorable regarding the WM functionality? Are these dynamical mechanisms compatible with actual biological brain dynamics?

In Chapter 3, we set the framework for developing mechanistic hypothesis of neural dynamics based on RNN modeling. In this chapter, we first design a simple sequential, memory-dependent task: sequential pattern matching (SPM). Then to optimize RNNs, we use our proposed method, i.e. the reinforced regression method. Then, we proceed to identify the emergent network dynamics and elucidate on the functionality of the emerged solutions.

4.2 WM Task Design

In this study, we design a sequential pattern-matching task that takes into account key phases of working memory tasks: stimulus processing, memory encoding and response execution [63]. The goal is to use a task of sufficiently low dimension as to be able to perform tractable and

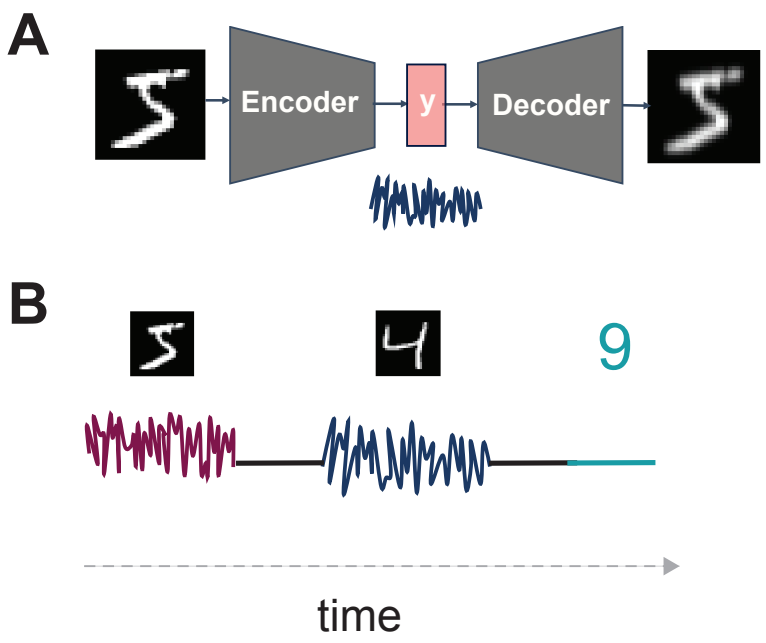

Figure 4.1: SPM Task. Panel A shows the VAE

potentially illuminating post-hoc analysis on the emergent dynamics of RNNs. In the proposed task, each trial consists of two random process stimuli alternating with delay intervals and a brief response interval (Fig. 4.1). Particularly, each stimulus is a two-dimensional Gaussian process obtained in the latent space of a Variational Auto Encoder (VAE) trained on the MNIST dataset of hand-written digits. We design the task rule to emulate summation, which differs from simple match or non-match tasks [94]. Note that in our setup, we use two different stimuli and we have 3 outcomes for summation. For example, valid trials in Fig. 4.1 are: (4,5), (4,4), (5,4) and (5,5).

4.3 Optimizing RNNs to Perform SPM Task

Recurrent networks with fully random connectivity as in equation (3.1) have a rich dynamical repertoire and thus are capable of generating complex temporal patterns that are commensurate with spontaneous cortical activities [76, 95]. To make these networks learn the function of interest and thus perform the task, we first define two variables decoded from the network activity:

$$\begin{aligned} z_o(t) &= \mathbf{W}_o^T \, \mathbf{r}(t) \\ z_d(t) &= \mathbf{W}_d^T \, \mathbf{r}(t) \end{aligned} \tag{4.1}$$

where $z_o(t)$ is a network output for generating responses, while $z_d(t)$ is a low-dimensional latent variable that is linearly decoded from neural firing rate activity, i.e. $z_d(t) = (z_{d_1}(t), z_{d_2}(t))$. In our network, invariant memory representations will be formed in this latent space. Optimization/learning proceeds by modifying the projection vectors $\mathbf{W}_o \in \mathbb{R}^{N\times 1}$ and $\mathbf{W}_d \in \mathbb{R}^{N\times 2}$. The network output $z_o(t)$ and dummy output $z_d(t)$ are fed back to the network via feedback weights i.e. $\mathbf{W}_f \in \mathbb{R}^{N\times 1}$ and $\mathbf{W}_{fd} \in \mathbb{R}^{N\times 2}$, respectively. This results in modified synaptic connectivity:

$$\dot{\mathbf{x}}(t) = -\mathbf{x}(t) + (\mathbf{J} + \mathbf{W}_f \mathbf{W}_o^T + \mathbf{W}_{fd} \mathbf{W}_d^T) \, \mathbf{r}(t) + \mathbf{W}_i \mathbf{u}(t) \tag{4.2}$$

The elements of $\mathbf{W}_f$ and $\mathbf{W}_{fd}$ are drawn independently from Gaussian distributions with zero mean and variance σ_f^2. The network receives the exogenous input (i.e., stimulus) $\mathbf{u}(t) \in \mathbb{R}^{2\times 1}$ via input weights $\mathbf{W}_i \in \mathbb{R}^{N\times 2}$ (see Fig. 4.2). This strategy effectively modifies the initial connectivity by addition of a low-rank component, allowing for more interpretable relations between the overall network connectivity and function [77, 82]. Note that a minimal rank, i.e. rank 1, perturbation could be used, but it is known to induce high correlations between emergent fixed points, thus restricting the potential range of emergent dynamics [82, 96].

Hence, to allow for a potentially wide range of solutions, we used a random connectivity plus rank 3 structure for the SPM task.

In our framework, optimization occurs only during the relevant temporal intervals in which these target signals are defined (Fig. C), which we term a temporally restricted error kernel. When applying this kernel, the total error derived for a given trial is:

$$E(t) = \frac{1}{2}\int_{\mathcal{T}_d} e_d(t)^2 dt + \frac{1}{2}\int_{\mathcal{T}_r} e_o(t)^2 dt \tag{4.3}$$

where $\mathcal{T}_d$ and $\mathcal{T}_r$ are the temporal epochs associated with the two delay periods and response period, respectively (Fig. C). Here,

$$e_d(t) = \|z_d(t) - f_d\|\,, \tag{4.4}$$

where $z_d = (z_{d_1}, z_{d_2})$ and $f_d = (f_{d_1}, f_{d_2})$ and

$$e_o(t) = \|z_o(t) - f_o\|\,. \tag{4.5}$$

Here, f_{d_1}, f_{d_2} and f_o are scalar real numbers chosen prior to optimization to represent the 2-dimensional stimulus and the trial outcome. During the delay intervals in particular, a low error thus implies that the neural activity linearly maps to a constant, invariant representation (i.e., $z_d \in \mathbb{R}^{2\times 1}$). Activity during temporal epochs outside of the these periods do not impact the error. Optimization proceeds by modifying readout weights $\mathbf{W}_o$ and $\mathbf{W}_d$ to minimize these errors.

Within the temporal error kernel, we deploy the FORCE method for parametric regression in RNNs. Here, $\mathbf{W}_o$ and $\mathbf{W}_d$ are updated using recursive least squares [74]. Briefly, to reduce

$e_o(t)$, we obtain

$$\begin{aligned}\mathbf{W}_o(t) &= \mathbf{W}_o(t-\Delta t) - e_o(t)\mathbf{P}(t)\mathbf{r}(t) \\ \mathbf{P}(t) &= \mathbf{P}(t-\Delta t) - \frac{\mathbf{P}(t-\Delta t)\mathbf{r}(t)\mathbf{r}^T(t)\mathbf{P}(t-\Delta t)}{1+\mathbf{r}^T(t)\mathbf{P}(t-\Delta t)\mathbf{r}(t)}\end{aligned} \tag{4.6}$$

where $\mathbf{P}(t)$ denotes the approximate estimate for the inverse of the correlation matrix of network activities with a regularization term

$$\mathbf{P}(t) = \int_{\mathcal{T}_r} \mathbf{r}(t)\mathbf{r}^T(t)dt + \alpha\mathbf{I}_N \tag{4.7}$$

where α is the regularization parameter and $\mathbf{I}_N$ the identity matrix. In the same manner, to reduce $e_d(t)$ we have

$$\mathbf{W}_d(t) = \mathbf{W}_d(t-\Delta t) - e_d(t)\mathbf{P}(t)\mathbf{r}(t). \tag{4.8}$$

Note that we update the associated inverse correlation matrices during training intervals $\mathcal{T}_d$ and $\mathcal{T}_r$ (shown in Fig. 4.2). In total, our training paradigm is a temporally regularized FORCE method that mitigates overfitting and in turn provides a potentially broader range of dynamical solutions to manifest. Indeed, it is known that optimizing RNNs using FORCE for a *sequential* trial-based task (here, a pattern association task with memory requirement) prevents the emergence of multiple fixed points in optimized networks, and thus can overly constrain the range of possible solution dynamics [96].

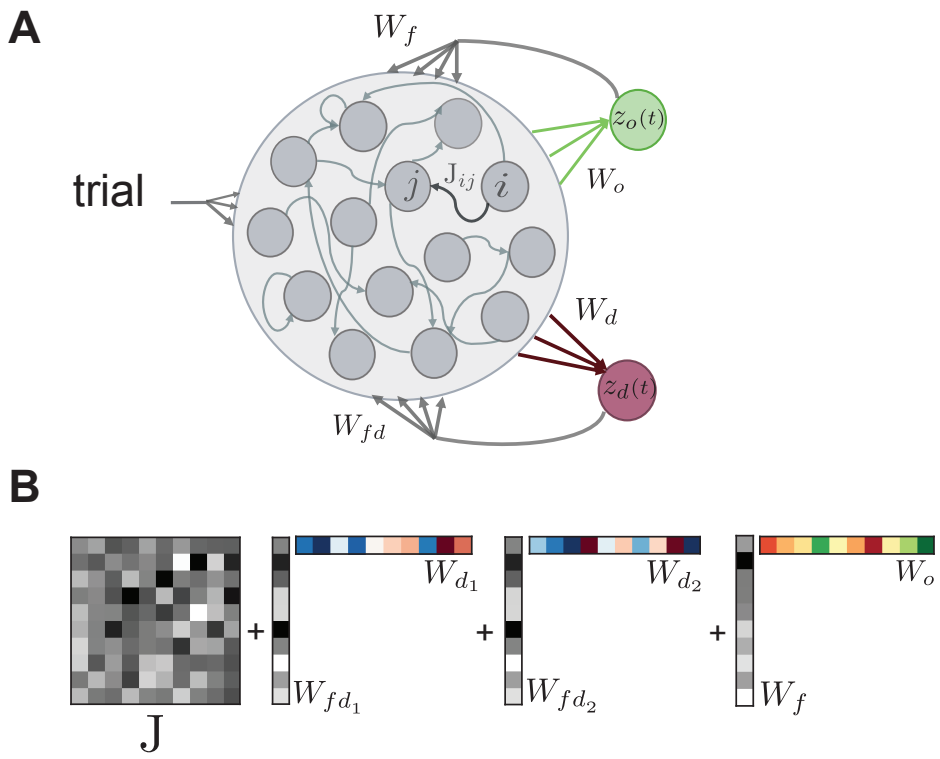

Figure 4.2: Panel A shows the schematic diagram of RNN. The network receives input trials sequentially via input weights and generates the task outputs $z_o(t)$ and memory encodings $z_d(t)$. (B) shows the initial synaptic connectivity matrix $\mathbf{J}$ and the low-rank structure added to it. We use a rank 2 structure for encoding memory and a rank 1 structure for generating response.

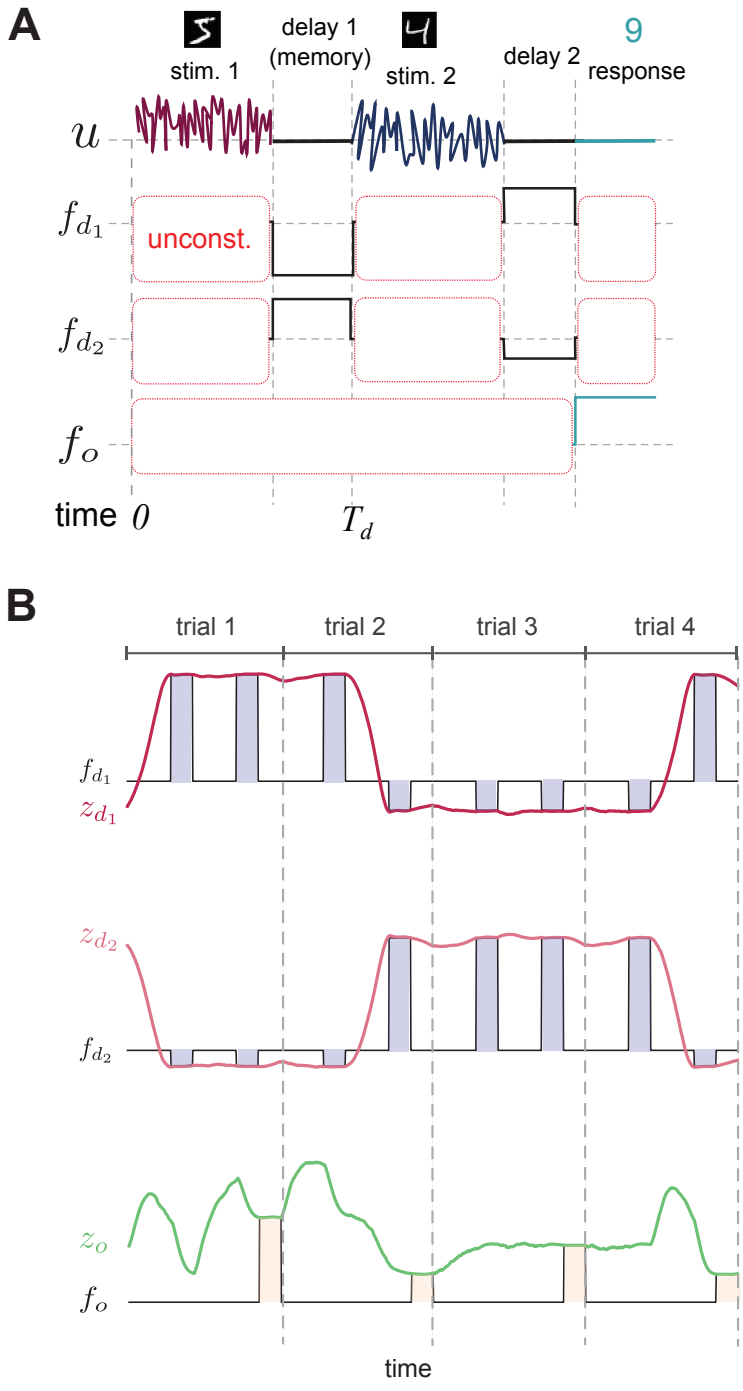

Figure 4.3: Training RNN using reinforced regression to perform SPM task. (A) shows a single trial, wherein each stimulus is a low dimensional representation of handwritten digits followed by short delay intervals. The network is optimized to generate the correct summation output during response interval. Digit representations are two dimensional Gaussian process and thus the dummy target f_d is two dimensional. (B) shows target signals $f_o(t)$ and $f_d(t)$ and their corresponding network outputs $z_o(t)$ and $z_d(t)$ for the trained network (here during 4 sequence of trials). Note that in reinforced regression paradigm, the weights are updated only during brief intervals (the shaded areas) and thus the network follows the target signals only during training intervals.

4.4 Dynamical Systems Analysis

The central theoretical question in our study pertains to analyzing the dynamics of our optimized networks. A first order question in this regard is to elucidate the landscape of attractors manifest in the network's vector field. In the same manner as discussed in Chapter 3, to study the stability of the origin, we can consider the eigenvalues of connectivity matrix (or shifted Jacobian matrix). For the optimized networks with the rank 3 structure, the Jacobian matrix at the origin is $\mathbf{J}_T = \mathbf{J} + \mathbf{W}_f \mathbf{W}_o^T + \mathbf{W}_{fd} \mathbf{W}_d^T$.

Understanding the location and stability of fixed points away from the origin is harder to ascertain analytically. Hence, we rely on a number of numerical procedures to identify these points. To locate stable fixed points used for task computations, we arrest trials at relevant time moments, then forward simulate to ascertain the asymptotic behavior of the network. In one set of simulations, this forward simulation is carried out for trials arrested at the end of the first delay period. In a second set of simulations, it is carried out after trial conclusion. The forward simulation is carried out for ten times the nominal trial length, at which time we assume the network state is in a stationary regime, i.e., within an ϵ distance of either a stable fixed point or limit cycle.

We can perform additional linearization about stable fixed points that are discovered numerically in this way. Here, the eigenvalue spectrum of Jacobian matrix, $\mathbf{Q}$, at these non-zero fixed points, denoted $\mathbf{x}^*$, is as follows

$$\mathbf{Q} = \left(\mathbf{J} + \mathbf{W}_f \mathbf{W}_o^T + \mathbf{W}_{fd} \mathbf{W}_d^T\right) \mathbf{R}' \tag{4.9}$$

where $\mathbf{R}'$ is a diagonal matrix with elements $\mathbf{R}'_{ij} = \delta_{ij}\mathbf{r}'_i$ with

$$\mathbf{r}' = 1 - \tanh^2(\mathbf{x}^*). \tag{4.10}$$

Note that if the states are largely saturated at a fixed point (as in the case of DFP encoding, Fig. 4.9C), then the entires of $\mathbf{r}'$ are very small, which contracts the spectrum of $\mathbf{Q}$.

In the following, we reveal the details of the emergent dynamics after training the networks.

4.5 Results

Optimization was terminated when the error during these epochs was below a specified threshold. As opposed to FORCE method, reinforced regression method obviates the issue of generating an error signal continuously throughout trials, since doing so may overly constrain the dynamics that the RNNs can manifest [18]. Our networks successfully solve the SPM task via several distinct dynamical mechanisms. Underlying these mechanisms are key invariant structures that are manifest in the network vector field, namely stable fixed points and attractive limit cycles. Interestingly, the mechanisms by which fixed points are used to encode latent memory representations and task outputs can vary. (Simulation parameters are provided in Appendix D.)

4.5.1 WM Can Be Encoded via Distinct Dynamical Mechanisms Associated with Tonic (or Persistent) and Phasic (or Transient) Neural Activation.

We enacted a trial-based WM task involving sequential pattern matching (SPM) that exhibits working memory requirements (Fig. 4.2). In our design, high-dimensional stimuli are encoded

as bivariate random processes, such that the network is required to temporally integrate each stimulus and then store a latent representation of said stimulus for later processing. We optimized RNNs to perform this task by using a modified FORCE method [74] that included a temporally restricted error kernel. Here, regression occurs at two phases during each trial: (i) during memory/delay periods, wherein we promote the formation of an invariant latent linear projection from neural units nominally associated with maintenance of a memory representation; and (ii) at the conclusion of each trial, wherein we promote a linearly decoded output response signal (Fig. 4.2). All other temporal epochs are unconstrained, thus obviating the need to generate an error signal continuously throughout trials, which may overly constrain the dynamics [18].

We found that optimized networks could produce both tonic and phasic activity patterns during delay periods, as exemplified for two different networks of 1000 neurons in Fig. 4.4. In order to study the dynamical mechanisms underlying these overt patterns we first used a numerical criteria on neuronal activity at the end of the delay period, T_d. Specifically, we arrested trials at T_d and forward simulated the networks autonomously to ascertain whether the activity was sustained at a fixed point. We identified four distinct dynamical mechanisms that could mediate working memory. In the case of tonic activation, network activity would indeed remain persistent, i.e., $\mathbf{x}(t)$, the state vector of neuronal activity, would remain near $\mathbf{x}(T_d)$ with $\|\dot{\mathbf{x}}\| \simeq 0$, indicative of a fixed point attractor (Fig. 4.5A). We refer to this mechanism as direct fixed point encoding (DFP). In the case of phasic patterns, $\mathbf{x}(t)$ in the forward simulation would deviate from $\mathbf{x}(T_d)$. In some cases, the network would always settle at a *different* fixed point from the memory representation (Fig. 4.5B, termed indirect fixed point encoding, IFP), independent of the stimulus or network initial condition. In other cases the network would always asymptotically approach a stable limit cycle attractor (Fig. 4.5C, limit cycle encoding, LC). In a fourth case (not depicted), the network could asymptotically

approach either a disparate fixed point or a limit cycle, depending on the stimulus realization (termed mixed encoding, see Fig. 4.6). In total, we optimized 1524 network models, of which 703 were identified of the direct fixed point (DFP) mechanism, 534 were of the indirect fixed point (IFP) mechanism, 182 were of the limit cycle (LC) mechanism, and 105 were of the mixed (Mix) mechanism. Given their dominance in the emergent solutions, our subsequent attention will be on understanding the workings of the DFP and IFP mechanisms, though we will later also untangle the factors that cause each mechanism to arise over the others.

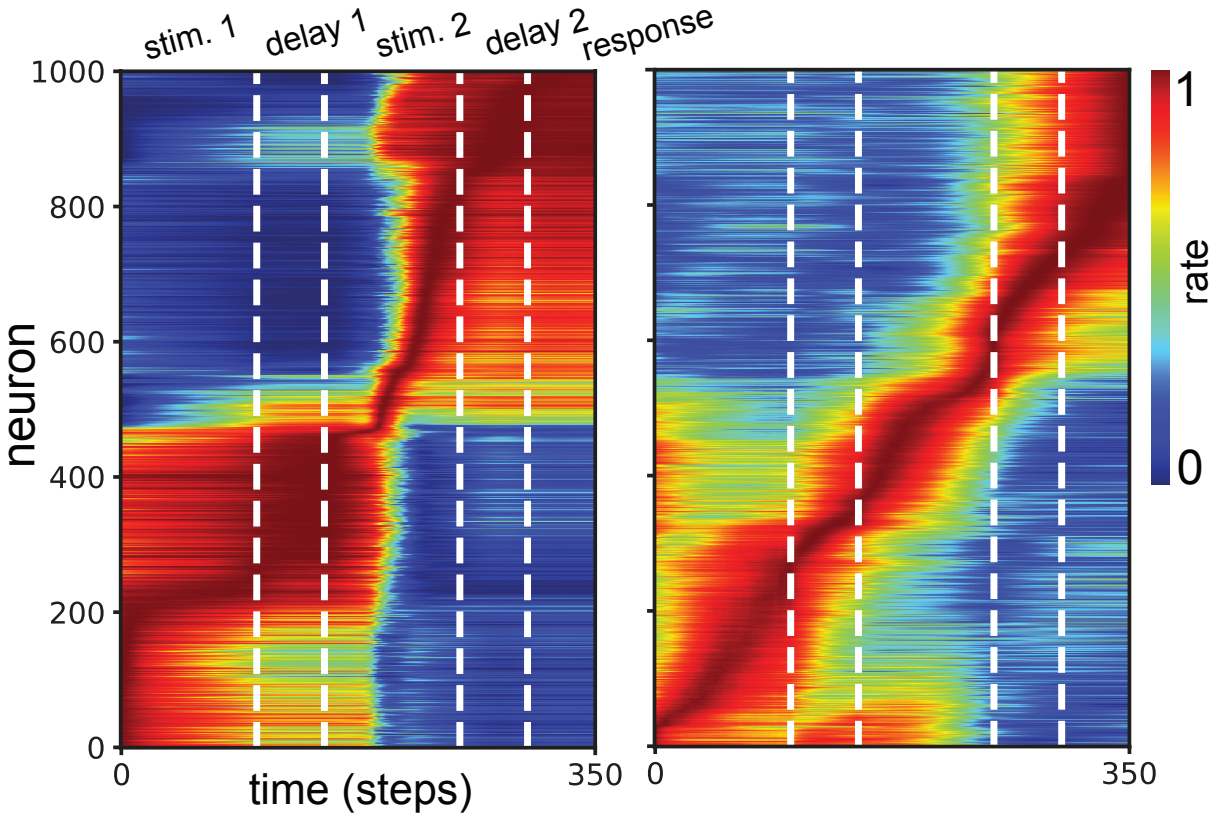

Figure 4.4: **Tonic vs. phasic pattern of neural activity.** Tonic and phasic activity for two different networks. Activity patterns (normalized) of neurons are sorted by the time of their peak value.

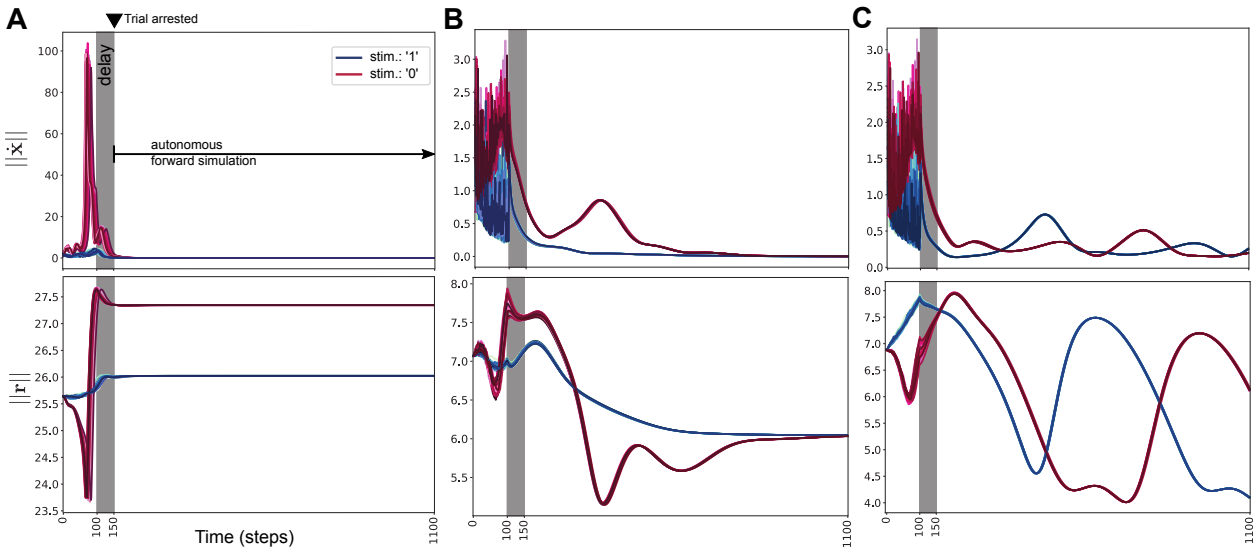

Figure 4.5: **Forward simulation of network after delay to identify distinct dynamical mechanisms underlying WM** . A, Direct Fixed Point encoding (DFP), where the network uses fixed points to encode memory representations of each stimulus. B, Indirect Fixed Point encoding (IFP), where the network asymptotically settles at a fixed point but this fixed point does not correspond to a memory representation. C, Limit Cycle (LC), where the network asymptotically approached a stable limit cycle attractor.

4.5.2 Indirect Encoding Efficiently Uses the Network Attractor Landscape.

The above findings suggests that key invariant structures in the network attractor landscape – stable fixed points and attractive limit cycles – determine whether and how delay activity takes on a tonic or phasic characteristic. To delve further into these mechanisms, we attempted to analyze how networks in each of the four categories leverage their respective attractor landscapes during the task.

We began by linearizing the dynamics at the origin and using mean-field results [82] to establish lower bounds on the number of fixed point attractors manifest in the network

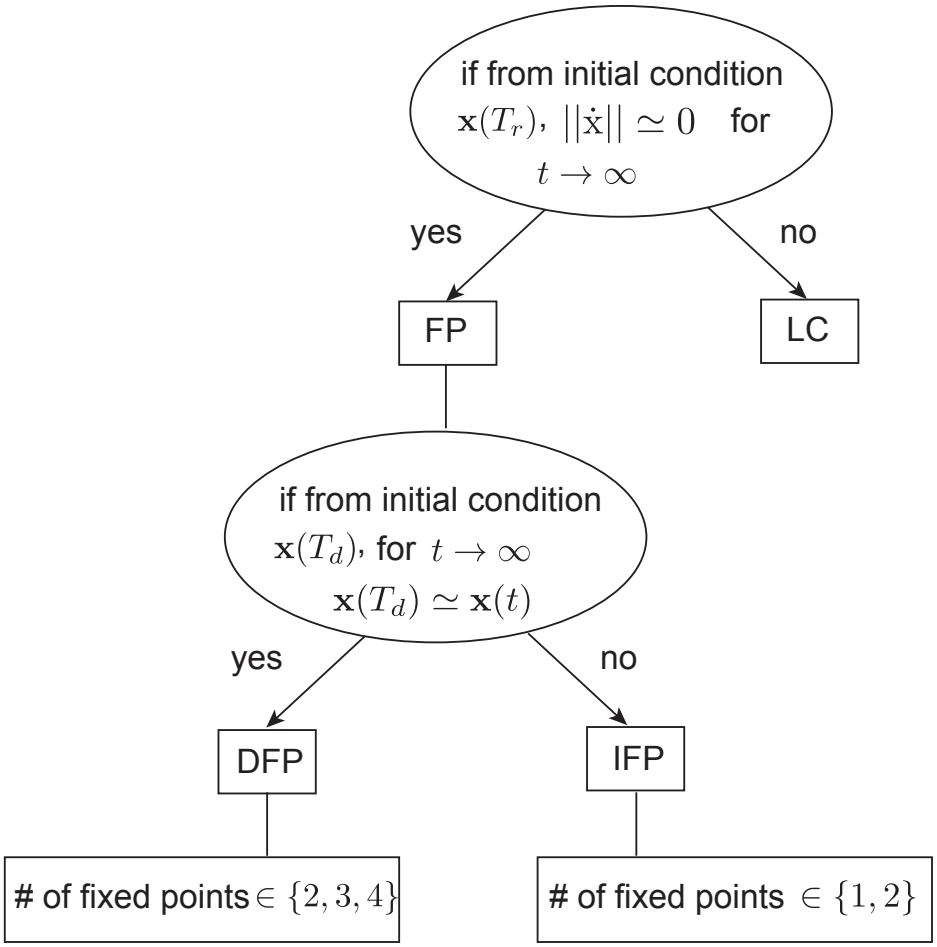

Figure 4.6: **Flow chart** Categorization of dynamical mechanisms based on (i) the type of asymptotic attractor (either fixed point or limit cycle) and (ii) whether delay periods correspond to a fixed point.

attractor landscape. Fig. 4.7A,B show how our four mechanistic categories break down along three key properties of spectra of the ensuing Jacobian matrix, where distinctions are readily observed. Most notably, the landscapes associated with direct fixed point encoding involve a greater number of fixed point attractors relative to indirect encoding. In support of this point, Fig. 4.7C illustrates representative low-dimensional projections of network activity in each of the four mechanisms with stable fixed points overlaid (here, we restrict attention to the positive quadrant, see also Discussion). In DFP encoding (Fig. 4.7C), the

sequential stimuli move the trajectory between different fixed points (associated with memory representations), culminating in an output that is itself associated with a different fixed point (i.e., here a total of four fixed points are used in the service of the task). In contrast, the landscape for IFP encoding (Fig. 4.7C) involves a *single* fixed point that does not encode memories, nor does it encode the nominal output (though, it is approached asymptotically if networks are forward simulated autonomously after trial cessation). Thus, IFP encoding is able to maintain invariant representations during the relevant memory periods without relying directly on the presence of multiple fixed point attractors (see also Fig. 4.8)

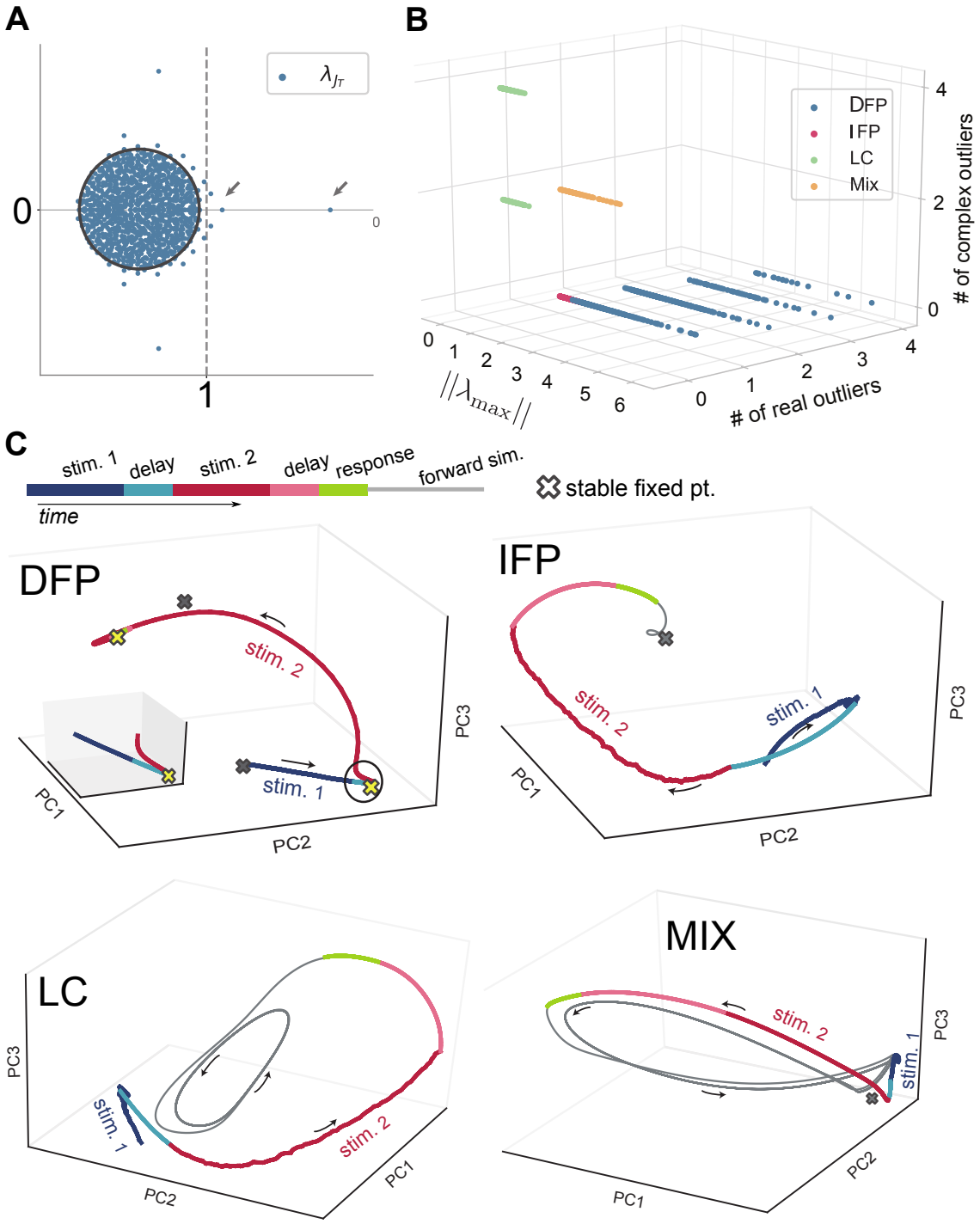

Figure 4.7: **Attractor landscape for optimized networks** A, Eigenvalue spectrum (for an exemplar DFP network). The gray circle shows the radius of the theoretical eigenvalue spectrum of the initial connectivity matrix $\boldsymbol{J}$. After optimization, a set of outliers emerges in the eigenvalue spectrum. Here, the initial connectivity matrix is the Jacobian at the origin (shifted by $-\boldsymbol{I}$). B, Categorization of four distinct mechanisms along key properties of the network Jacobian evaluated at the origin. C, Attractor landscape and trajectory of exemplar task trials. Three-dimensional neural trajectories are obtained via applying Principle Component Analysis (PCA) to 1000-dimensional neural activity from networks of each dynamical mechanism. In DFP, the network creates 4 stable fixed points to solve the SPM task. For the displayed trajectory, the network uses two fixed points (shown in yellow) to directly encode the memory and trial output (the inset shows the area inside the circle). In IFP, the memory representation and trial output are encoded along the slow manifold of the single fixed point in the state space. In LC, the trajectories approach a stable limit cycle. For the mixed mechanism, both a stable fixed point and limit cycle are observed.

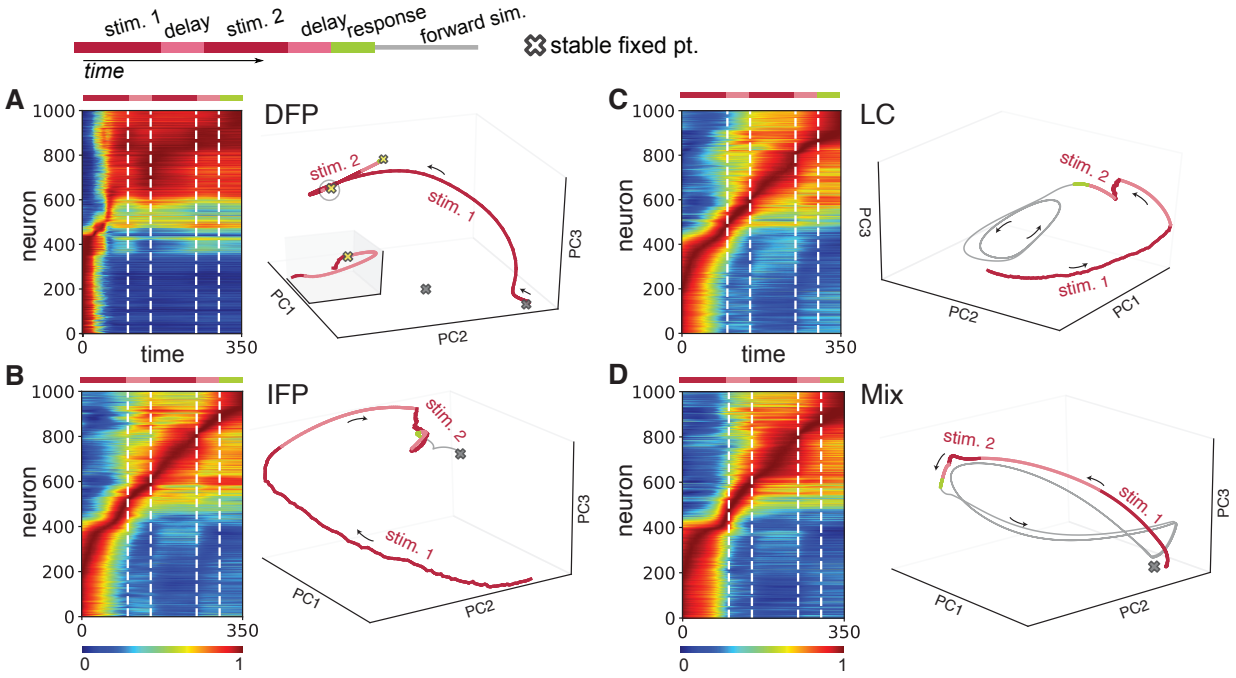

Figure 4.8: **Neural activities and associated dynamical landscape for a single trial** For the same exemplar networks as in Fig. 4.7, but a different set of stimuli (i.e., here, two realizations of the same digit are presented), neural activity and associated low-dimensional (PCA) trajectories are plotted. Note that PCA components are obtained for each exemplar network individually. The trajectories are color coded using the same scheme as the color bar on the top. In DFP, the network creates four stable fixed points to solve the SPM task (the inset shows the area inside the circle). For the displayed trajectory, the network uses two fixed points (shown in yellow) to represent memory and trial output. In IFP, memory representation and trial output are encoded along the slow manifold of the single fixed point in the state space. In LC, the trajectories approach a stable limit cycle. For the mixed mechanism, both a stable fixed point and limit cycle can be seen.

4.5.3 Indirect Encoding Uses Slow Manifolds to Sustain Memory Representations

Following from the above, IFP encoding appears to use the geometry of the stable manifolds of the single fixed point to maintain memory representations. Fig. 4.9A illustrates the spectrum of the linearized dynamics about the fixed point in the previous IFP example encoding model, where we see many eigenvalues near the imaginary axis, indicating the presence of slow, stable manifolds along which activity flows in a relatively invariant fashion. These manifolds provide

the opportunity for a low-dimensional latent representation of memory to be maintained, despite phasic activity (along the manifold). Indeed, because we encourage linearly mapped latent representations via our optimization method (see Methods), we know these manifolds have a planar geometry in the firing rate activity variables. In contrast, Fig. 4.9B illustrates the spectra resulting from linearization about two memory fixed points in a DFP model. Here we note that eigenvalues are relatively offset from the imaginary axis, indicating rapid convergence to the fixed point. This conclusion is supported in Fig. 4.9C, which shows the relative proportion of delay periods in which neurons are in the saturated (nonlinear) vs. linear range of the activation function, i.e. tanh(.) , for each model we trained. The much larger proportion of saturated neurons in DFP encoding indicates that the mass of eigenvalues for these models is relatively contracted and offset from the imaginary axis (see Methods and equation (4.9)) and thus associated with fast decay to the fixed points.

To further understand the circuit-level details mediating the DFP and IFP mechanisms, we characterized the connectivity between neurons. We first noted that DFP encoding leads to an overall much greater distribution of connectivity weights between neurons relative to IFP (Fig. 4.9D). Next, we sorted neurons according to their peak activation (as in Fig. 4.4) and examined their average pre-synaptic activity throughout the course of trials. We found that neurons in DFP encoding exhibited highly structured synaptic tuning to different stimuli and memory periods, in contrast to IFP encoding (Fig. 4.9E). Finally, we examined the bidirectional synaptic weight between 'adjacent' neurons (ones with temporally sequential maximal activation). Here, DFP exhibits no systematic connectivity structure, while IFP shows that neurons with similar peak activation times are more tightly coupled (Fig. 4.9F). This latter point suggests that traversal along the slow manifolds is mediated by an internal sequential, 'daisy chain' type of structure embedded within the trained IFP encoding network.

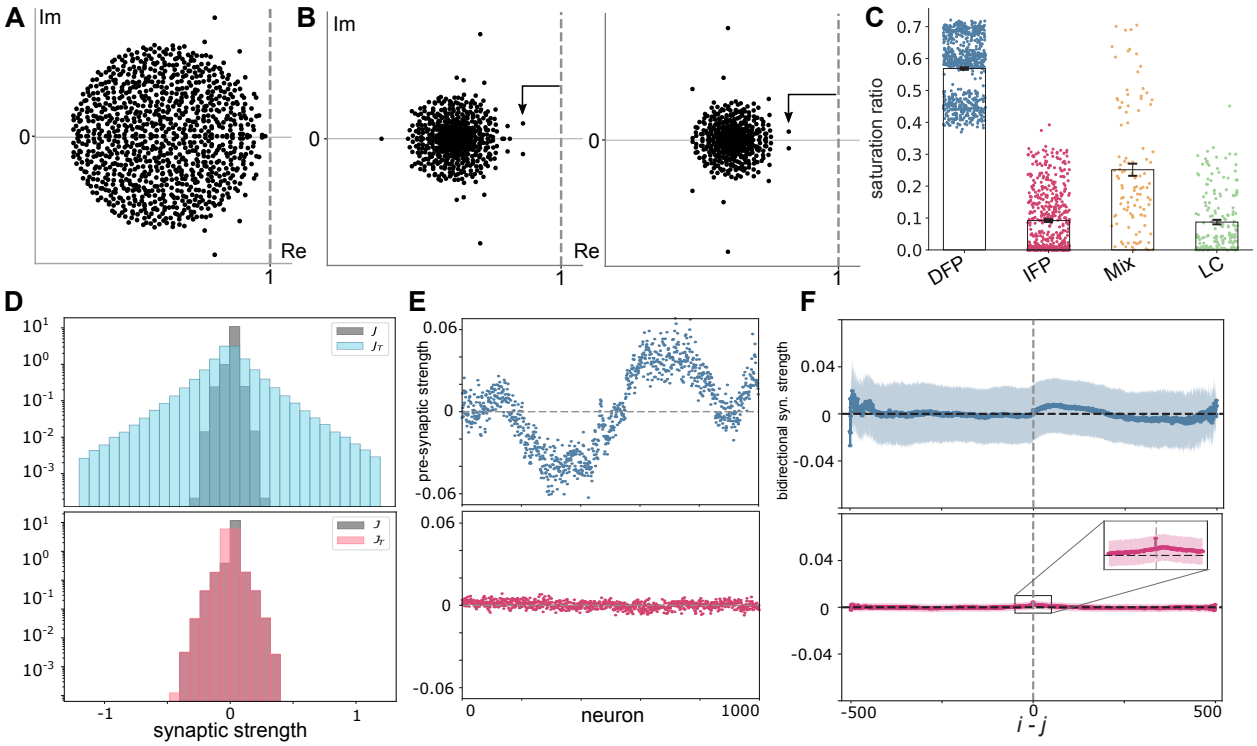

Figure 4.9: **Eigenvalue spectrum at task fixed points and connectivity characterization.** A, Eigenvalue spectrum of Jacobian matrix at the single non-zero stable fixed point of IFP (shown in Fig. 4.5b). B, Eigenvalue spectra of Jacobian matrix at memory fixed points of DFP (shown in Fig. 4.5a). C, Saturation ratio (the ratio of neurons with activity in saturated range of activation function during memory interval (averaged over all trials)) for all networks simulated (across all four mechanisms). Standard error of the mean is depicted. D, Distribution of connectivity matrix entries (i.e., weights) before and after training for DFP (the top panel) and IFP (the bottom panel). E, Average pre-synaptic (incoming connections) strength sorted by peak activation of neurons (as in Fig. 4.4) for DFP and IFP, respectively. F, Comparison of mean and variance of elements of task connectivity matrix based on temporal distance of neurons. For IFP (the bottom panel) temporally adjacent neurons are more tightly coupled and a peak can be observed. (The inset shows this peak and i, j denote neurons indices.)

4.5.4 Stable Manifold Encoding Is Forgetful, but Robust.

We sought to better understand the functional advantages of the different mechanism types. In this regard, we interrogated networks by extending delay periods beyond the nominal training requirements, a form of increased memory demand. The main question here is how increasing the memory demand in this way would affect activity and consequently degrade task performance. Fig. 4.10A illustrates the comparison of neural activity patterns for DFP and IFP encoding categories (with extended delay equal to five times the nominal delay interval). For DFP encoding, regardless of the length of the extended delay, the neural activity is unaffected since the network uses fixed points as the invariant structure to encode memory traces. Consequently, after the extended delay ends and the network receives the second stimulus, task computations can be executed correctly. However, for IFP encoding, during the extended delay interval, neural activity gradually drops away which results in loss of function due to deviation from the 'correct' activity pattern upon receiving the second stimulus. Fig. 4.10B summarizes the deviation from the nominally 'correct' post-delay neural activity as a function of delay extension for our two FP mechanisms, as well as LC and Mix. As expected, for DFP encoding this deviation is near zero. In contrast, for IFP the networks can tolerate extended delay up to % 100 of the nominal delay, after which point performance gradually drops, i.e., the correct representation is 'forgotten'.

To assay other functional aspects of these mechanisms, we examined how performance of our networks would tolerate the presence of a distracting noise added to the actual stimulus. Here, we found a counterintuitive functional advantage of 'forgetting', relative to the DFP mechanism. We specifically added uncorrelated noise of differing variance to the first of the two sequential stimuli and examined deterioration from the nominal 'correct' neural representation at trial conclusion. For values of noise variance that are less than the stimulus

variance (vertical dotted line Fig. 4.10C) IFC (and indeed LC) encoding are highly robust to perturbations, and indeed variances in excess of an order of magnitude greater than the stimulus can be tolerated. In stark contrast, DFP encoding is highly fragile with respect to distracting noise, with rapid and near-complete breakdown of the correct neural representation after modest perturbation (Fig. 4.10C). To understand this mechanism we carefully studied the trajectories in low-dimensional space in the presence of distracting noise (Fig. 4.11), from which we ascertained that the distracting noise was essentially placing the trajectory in an erroneous basin of attraction, i.e., causing an incorrect memory fixed point to be induced. This result runs counter to classical Hopfield-type associative memory theory [68], which presumes that basins are useful to rejecting noise and uncertainty. Our finding, in essence, indicates that the high reliance on many fixed points for stable memory representations in DFP encoding makes this mechanism more susceptible to the temporal integration of noise (see also Discussion).

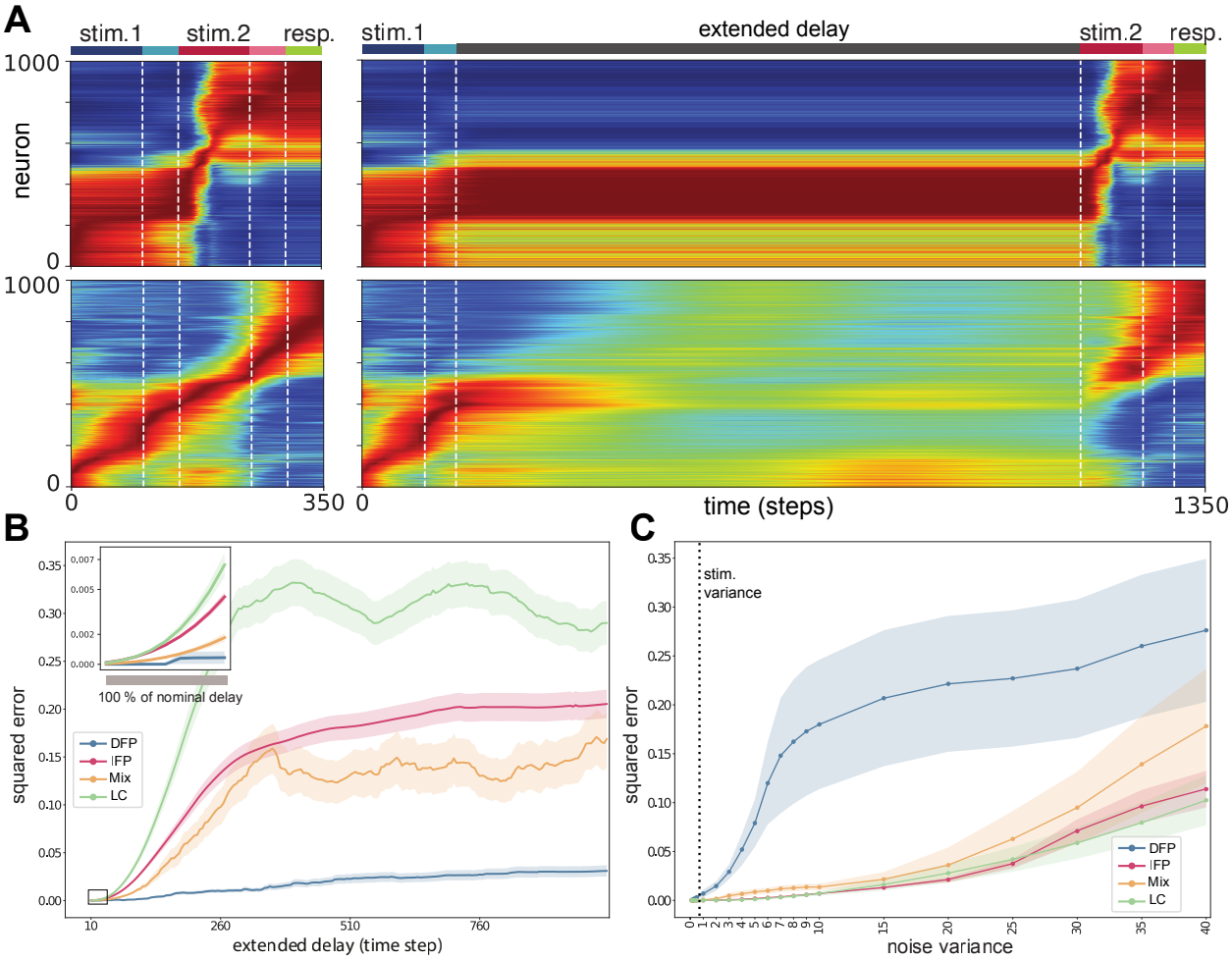

Figure 4.10: **Functional advantages/disadvantages of each mechanism type.** A, Comparison of activity patterns before and after increasing memory demand for DFP (top panel) and IFP (bottom panel). B, Summary of deviation from correct pattern of activity across different values of extended delay for all optimized networks. The squared error shows the difference between correct and deviated trial outputs averaged over all trials and associated networks. C, Summary of deviation from correct pattern of activity across different values of noise variance for all optimized networks. B and C show that IFP, LC and mixed mechanisms are forgetful, but robust to sizable perturbations.

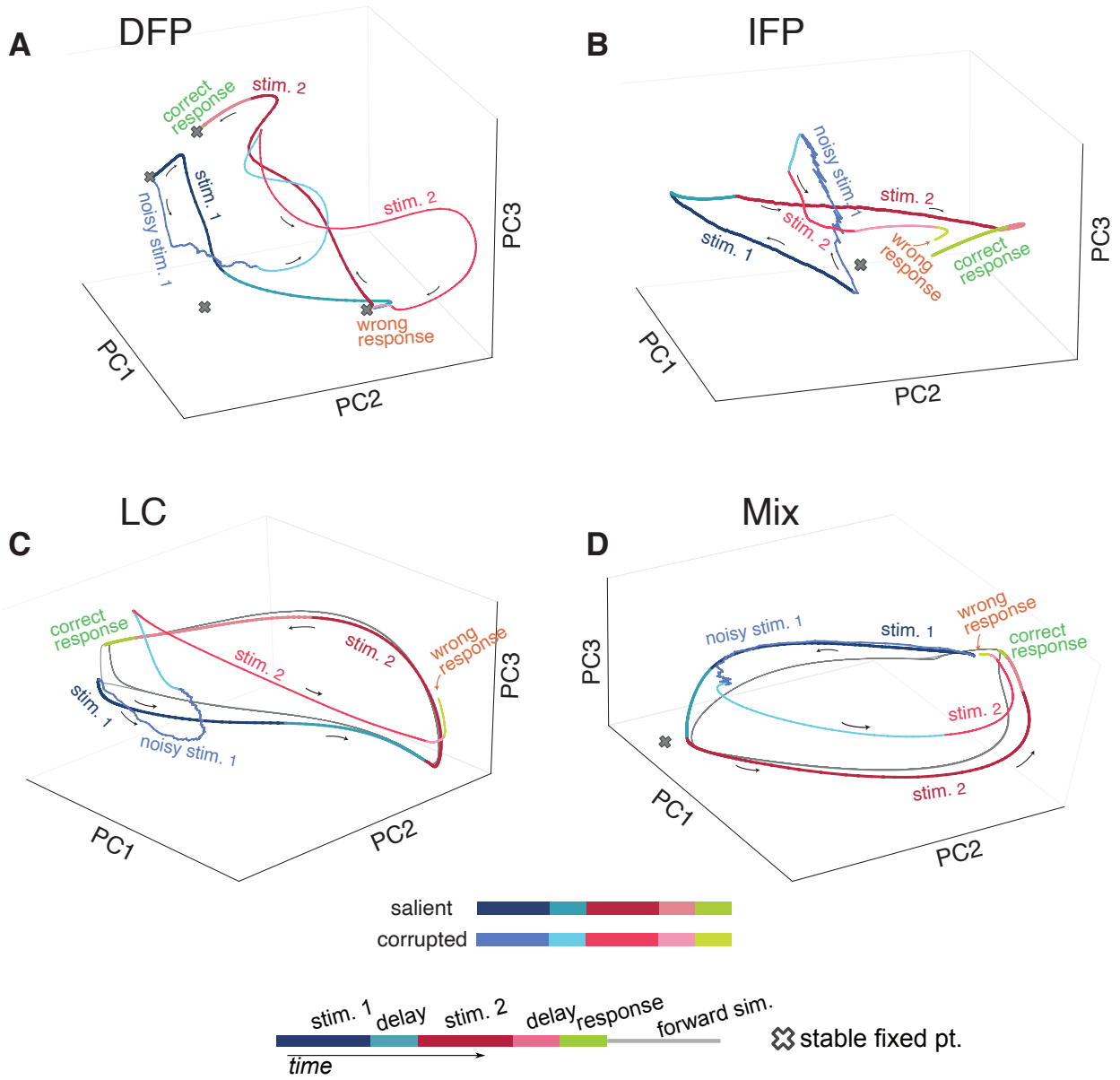

Figure 4.11: **Neural trajectories of perturbed and salient trials** Plot show how noise corrupts the salient trajectory for all four mechanisms (same trial and initial condition). PCs are exclusive to each network. In DFP, distracting noise places the trajectory in an erroneous basin of attraction and thus the network generates an incorrect response; in IFP noise pushes the trajectory away from the 'correct' slow manifold.

4.5.5 Initial Network Properties Dictate the Emergence of Different Solution Dynamics.

Finally, we sought to understand the factors prior to optimization that bias the emergent dynamics towards one type of mechanism versus another. We considered three main network properties: (i) the strength of connectivity, g, (ii) the variance of feedback, σ_f, and (iii) the sparsity of the initial connectivity matrix. We varied these parameters over their possible ranges. Fig. 4.12 illustrates the effect of different parameterizations: for small values of g the trainability of networks is poor, but improves significantly as g increases. In other words, large random initial connectivity facilitates training, consistent with known results [74, 82]. For $g < 1$ the untrained network has one stable fixed point at the origin and the emergent trained dynamics tend to be of DFC or IFC encoding (Fig. 4.12A). Interestingly, the variance of feedback weights, σ_f has a notable effect on the emergent dynamics; for large values of σ_f the networks tend to form DFC models and as σ_f decreases only IFC and LC models arise (Fig. 4.12B). The sparsity of initial connectivity matrix has no significant effect on the trainability of networks nor the emergent dynamics (Fig. 4.12C).

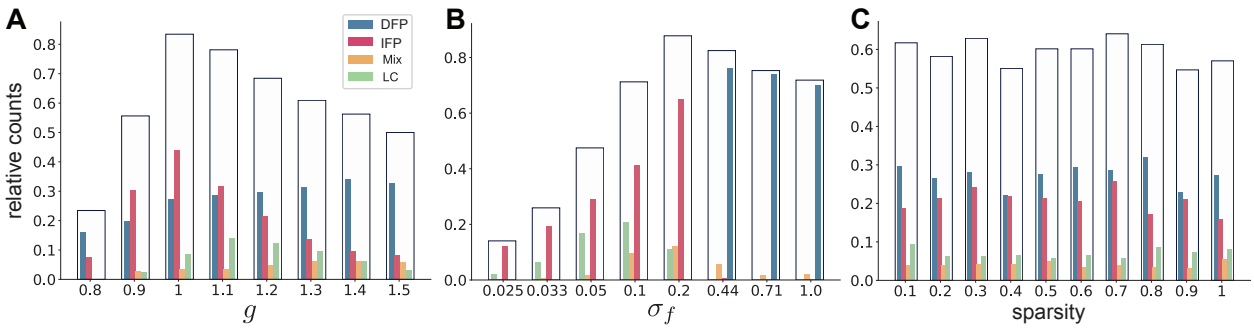

Figure 4.12: **The effect of parameters prior to optimization on the diversity of the emergent solutions.** Relative count shows the number of trainable networks divided by the total number of networks for each specified value of parameters. Transparent bars show the relative count of trainable networks and the inner bars show the corresponding emergent types for each specified value of parameters. A The strength of connections within the initial connectivity matrix **J**. B, The variance of feedback weights. C The sparsity of **J**.

4.6 Conclusion and Discussion

4.6.1 Learning a Diversity of Dynamics for WM Function

In this work we used a top-down optimization-based approach to investigate potential dynamical mechanisms mediating WM function. By training/optimizing RNNs using a modification of the FORCE regression method, we found four qualitatively different types of network dynamics that can mediate function. At a mechanistic level, these solutions are differentiated on the basis of the number of asymptotically stable fixed points manifest in the network vector field and, crucially, how those fixed points are leveraged in the service of the task. We note especially two solution types, one reflecting neural memory representations that are highly persistent corresponding to direct encoding at fixed points (i.e., DFP), versus the other where neural representations are transient and correspond to traversal along slow

manifolds in the network state space (i.e., IFP). At the level of neural activity, DFP produces tonic sustained activity during delay periods, while IFP produces phasic, transient activity.

Our results are related to prior work that has shown that persistent versus transient encoding of memories can manifest in neural networks trained on different WM tasks and under different optimization/learning schemes [93]. Here, we choose to focus on a single, structured task in an effort to go beyond overt activity characterizations and carefully dissect the underlying dynamical mechanisms associated, namely the attractor landscape in the neural state space. Doing so provides not only insight into potential generative circuit processes but also allows us to perform sensitivity analyses to ascertain nuanced functional advantages associated with the different mechanisms.

4.6.2 Tradeoff between Efficiency, Memory Persistence and Robustness

In particular, our results suggest an interesting balance between persistence and robustness of memory representations. Specifically, the DFP mechanism resembles in many ways traditional associative memory attractor dynamics, in the sense of Hopfield networks [68]. Here, each memoranda is associated with a distinct, asymptotically stable fixed point. On the one hand, such a mechanism is able to retain memories for arbitrary lengths of time. Further, the dynamics with the attractor basins can nominally correct for small perturbations to neural trajectories at the onset of memory periods. However, our results suggest that this latter classical interpretation breaks down when considering sequential, time-varying stimuli. In this case, perturbations to stimuli can accrue over time, causing neural representations to stray into errant basins of attraction, ultimately leading to failure of performance.

In contrast, in the IFP encoding mechanism, the network vector field exhibits a smaller number of fixed points that do not encode memoranda directly. Rather, memory representations are formed from projection of neural activity along slow manifolds that are ostensibly shaped through optimization of the network vector field. The fixed points here are, in essence, 'shared' between memoranda. This mechanism turns out to be far more robust to time-varying stimulus perturbations. There are likely two factors related to this robustness. First, noisy perturbations may not be able to easily move trajectories off of the 'correct' slow manifold. Second, there are no competing attractors to absorb errant trajectories, as would be the case in the DFP mechanism. In total, the IFP encoding can be viewed as an overall more efficient use of neural dynamics wherein the lack of a persistent representation (i.e., 'forgetfulness') is offset by both a lighter weight coding scheme in terms of the number of attractors deployed in the state space, leading – perhaps paradoxically – to more robust performance.

4.6.3 Shaping a Landscape with Few Attractors

Expanding on the above point of efficiency, it is of note that the limit cycle and mixed mechanisms are most comparable to IFP in terms of the way in which they attractor landscape is used in the service of the task. In the LC mechanism in particular, the oscillatory cycle is not itself used to encode or sustain the memory, but rather shapes the landscape to create slow manifolds for encoding, similar to IFP. Thus, while the oscillation is not directly functional, it nonetheless is critical in establishing the 'right' landscape for task completion. From an energetic standpoint, the indirect mechanisms are potentially less expensive since most neurons are inactive at any moment in time, in contrast to DFP encoding.

4.6.4 Temporally Restricted Optimization Promotes Solutions that Are Compatible with Observed Dynamics *in vivo*

Our results shed light on the different means by which recurrent networks can embed memory functions within their dynamics. Such a question is highly relevant to understanding how key machine learning and artificial intelligence constructs such as RNNs encode complex context-dependent functions [97]. However, they also suggest mechanistic interpretations for actual WM circuits in the brain, given the seeming prevalence of phasic activity patterns during delay intervals observed *in vivo* [63]. Indeed, it has been observed that neurons in memory-relevant regions such as prefrontal cortex do not necessarily maintain persistent activity throughout long delay periods, but rather may 'ramp' on and off at systematic time points [63], as is compatible with our IFC mechanism. Further, in the IFC mechanism, most neurons are lightly saturated (Fig. 4.9c), meaning that most neurons are within a linear regime, as thought to occur in actual neural circuits [18, 98].

Notably, the IFP dynamical mechanism only arises after using the proposed temporally restricted error kernel. Indeed, we found that using the native FORCE method without such a kernel leads to poor trainability; and further those networks that do manage to be trained are highly fragile to the extended delay and noise perturbations we considered. This fragility ostensibly arises due to the latent outputs being overly constrained in this situation. Indeed, the choice of how to constrain these outputs throughout the task is somewhat arbitrary in the first place. Hence, the temporally restricted error kernel may be allowing for the emergence of more naturalistic dynamics in our RNNs.

4.6.5 Potential for Enhanced Fast Learning and Generalization

An important technical caveat is that we have set up our RNNs to produce activity in the positive quadrant. Hence, our analysis focuses on characterization of the attractor landscape in that region of the state space. However, because we consider an odd activation function, we know analytically that the fixed points analyzed in our networks have 'mirror' negative fixed points that are not directly used in the service of the task, which means that these dynamical features are essentially 'wasted' by construction and network design. A speculative hypothesis is that these fixed points may allow the network to more quickly learn a related task with minimal synaptic modification, i.e., by leveraging the mirror dynamics that are already embedded in the network. Such a concept is related to the idea of meta-learning [99] and may be an interesting line of future study.

Chapter 5

Conclusion and Future Directions

5.1 Concluding Remarks

In this work we contributed our efforts to develop hypotheses regarding the functional role of brain dynamics. Specifically, we directed our focus toward the top-down modeling approaches in engineering neuroscience. To address the notion of functionality, we incorporated optimization methods within two different paradigms in the context of engineering theory. First, we postulated our hypothesis regarding the general functionality of neural circuits for information processing and synthesized the optimal dynamical substrates for maximization of empowerment. In the second part of our work, we considered a more specific function of the neural circuits, i.e. working memory. Because complex cognitive functions lack a well-defined objective function, we developed our hypothesis on the basis of creating generative models of such complex functions using artificial neural networks and thus identified network dynamics that facilitate working memory functions.

In the following, I first make some remarks on the remaining challenges and potential ideas within the top-down modeling approach that we pursued in this dissertation. Then, I provide my opinions on the potential prospects of engineering theory within the context of neuroscience.

5.1.1 Remaining Challenges within the Top-down Modeling Approach Pursued in This Work

Optimization is a fundamental component of top-down modeling approaches and thus optimization methods/techniques could largely impact the nature of inferences about brain functions. In particular, in Chapter 2, we used deep neural networks to approximate unknown probability distributions for empowerment maximization. Here, the decision variables are parameters (weights and biases) of neural networks and thus we face a non-convex optimization problem. As such, different hyperparameters and initializations could potentially lead to different local maxima. Then, how can we ensure the validity of our hypothesis? In our work, we substantiated the validity of our inferences on the basis of intuition and the prevailing observations about neural dynamics. But what about the problems where we lack apriori intuition or further evidences about the logic of our inferences? Should we explore other modeling approaches or try to devise a more reliable optimization strategy? Moreover, in Chapter 3 and 4, we incorporated realistic assumption within the optimization method for minimization of the error regression. As compared to similar previous methods, reinforced regression paradigm led to the emergence of neural activities that are commensurate with neural recordings. However, the update rule for modification of synaptic connections is not biologically plausible. A question that arises here is how the nature of solutions may change if we use a realistic update rule (i.e., Hebbian learning rule)?

In particular in this work, we focused on top-down modeling to link the brain function with neural network/circuit dynamics. In the network model of Chapter 2, we considered several simplifying assumptions including *homogenous* neural networks with *firing-rate* neurons and network time constant equal to 1. However, there remain open questions regarding the mathematical formulations and optimization strategies for non-homogenous neural network models as well as modifying our models by considering the complex dynamics of individual neurons.

Interestingly, in Chapter 4, we observed that if the initial dynamical regime of random RNNs is near chaotic, i.e. $g \geq 1$, the trainability of such RNNs increases to perform WM tasks. Basically, these random RNNs have at least one unstable fixed point as compared to random RNNs with one stable fixed point at the origin (i.e. $g < 1$). Linking this observation back to the results in Chapter 2, at a high-level, we can conclude that chaotic random RNNs may be *better* models for generating mechanistic hypothesis of brain functions since their dynamics are compatible with optimal dynamics obtained via empowerment maximization. In other words, chaotic RNN dynamics may facilitate task-based optimization since task computations can be implemented in a wider swath of state space. Perhaps it is worth exploring the use of empowerment maximization as a preprocessing step for task-based modeling with the aim of limiting the space of model parameters for efficient training.

5.1.2 Future Landscape of Top-down Modeling within Engineering Neuroscience

Indeed, understanding the computational bases of brain functions have been a paramount challenge of neuroscientific research. An essential step to tackle such challenge is directing

effort to a synergy among different theoretical and practical disciplines. Specifically, theoretical and engineering neuroscience are reciprocal to each other; using the fundamental theoretical knowledge one can formulate principles in an abstract fashion as a blueprint for synthesizing/analyzing models of brain functions. In turn, by using tools from engineering one can draw meaningful inferences about the mechanisms underlying the brain computations, which ultimately can lead to novel theories about the link between neural processings and high-level functions.

Engineering neuroscience paradigm can potentially contribute to the following areas:

- Neurostimulation
- Reinforcement Learning
- Neuroscience Inspired Artificial Intelligence

References

[1] Eric R Kandel, James H Schwartz, Thomas M Jessell, Department of Biochemistry, Molecular Biophysics Thomas Jessell, Steven Siegelbaum, and AJ Hudspeth. *Principles of neural science*. Vol. 4. McGraw-hill New York, 2000.

[2] Alan A Baumeister, Kristopher Henderson, Joni Lee Pow, and Claire Advokat. "The early history of the neuroscience of attention-deficit/hyperactivity disorder." In: *Journal of the History of the Neurosciences* 21.3 (2012), pp. 263–279.

[3] Daniel Tranel, Gregory Cooper, and Robert L. Rodnitzky. "Higher Brain Functions." In: *Neuroscience in Medicine*. Ed. by P. Michael Conn. Totowa, NJ: Humana Press, 2003, pp. 621–639. DOI: 10.1007/978-1-59259-371-2_29.

[4] Jon Driver, Patrick Haggard, and Tim Shallice. "Introduction. Mental processes in the human brain." In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 362.1481 (2007), pp. 757–760.

[5] "The Neural Basis of Cognition." In: *Principles of Neural Science, Fifth Edition*. New York, NY: McGraw-Hill Education, 2014.

[6] Lyric A Jorgenson et al. "The BRAIN Initiative: developing technology to catalyse neuroscience discovery." In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 370.1668 (2015), p. 20140164.

[7] Ritchie Chen, Andres Canales, and Polina Anikeeva. "Neural recording and modulation technologies." In: *Nature Reviews Materials* 2.2 (2017), pp. 1–16.

[8] Konrad Körding. "Decision theory: what" should" the nervous system do?" In: *Science* 318.5850 (2007), pp. 606–610.

[9] Peiran Gao and Surya Ganguli. "On simplicity and complexity in the brave new world of large-scale neuroscience." In: *Current opinion in neurobiology* 32 (2015), pp. 148–155.

[10] Steven J Schiff. *Neural control engineering: the emerging intersection between control theory and neuroscience*. MIT Press, 2012.

[11] G Bard Ermentrout, Roberto F Galán, and Nathaniel N Urban. "Relating neural dynamics to neural coding." In: *Physical review letters* 99.24 (2007), p. 248103.

[12] Rodrigo Quian Quiroga and Stefano Panzeri. *Principles of neural coding*. CRC Press, 2013.

[13] Eugene M Izhikevich. *Dynamical systems in neuroscience*. MIT press, 2007.

[14] ShiNung Ching. "Control-Theoretic Approaches for Modeling, Analyzing, and Manipulating Neuronal (In) activity." In: *Dynamic Neuroscience*. Springer, 2018, pp. 219–238.

[15] Henry Markram. "THE HUMAN BRAIN PROJECT." In: *Scientific American* 306.6 (2012), pp. 50–55.

[16] LA Abott, FA Bisby, and DJ Rogers. "Taxonomic analysis in biology, computers, models and databases." In: *New York* (1985).

[17] Nancy J Kopell, Howard J Gritton, Miles A Whittington, and Mark A Kramer. "Beyond the connectome: the dynome." In: *Neuron* 83.6 (2014), pp. 1319–1328.

[18] Guangyu Robert Yang and Xiao-Jing Wang. "Artificial neural networks for neuroscientists: A primer." In: *arXiv preprint arXiv:2006.01001* (2020).

[19] Anthony CC Coolen, Reimer Kühn, and Peter Sollich. *Theory of neural information processing systems*. OUP Oxford, 2005.

[20] Alexander G Dimitrov, Aurel A Lazar, and Jonathan D Victor. "Information theory in neuroscience." In: *Journal of computational neuroscience* 30.1 (2011), pp. 1–5.

[21] Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. "Empowerment: A universal agent-centric measure of control." In: *Evolutionary Computation, 2005. The 2005 IEEE Congress on*. Vol. 1. IEEE. 2005, pp. 128–135.

[22] Maximilian Karl, Maximilian Soelch, Philip Becker-Ehmck, Djalel Benbouzid, Patrick van der Smagt, and Justin Bayer. "Unsupervised Real-Time Control through Variational Empowerment." In: *arXiv preprint arXiv:1710.05101* (2017).

[23] Ahmed H Qureshi and Michael C Yip. "Adversarial Imitation via Variational Inverse Reinforcement Learning." In: *arXiv preprint arXiv:1809.06404* (2018).

[24] Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. "Variational intrinsic control." In: *arXiv preprint arXiv:1611.07507* (2016).

[25] David Barber Felix Agakov. "The IM algorithm: a variational approach to information maximization." In: *Advances in Neural Information Processing Systems* 16 (2004), p. 201.

[26] Takashi Hayakawa, Takeshi Kaneko, and Toshio Aoyagi. "A biologically plausible learning rule for the Infomax on recurrent neural networks." In: *Frontiers in computational neuroscience* 8 (2014), p. 143.

[27] Taro Toyoizumi, Jean-Pascal Pfister, Kazuyuki Aihara, and Wulfram Gerstner. "Generalized Bienenstock–Cooper–Munro rule for spiking neurons that maximizes information transmission." In: *Proceedings of the National Academy of Sciences* 102.14 (2005), pp. 5239–5244.

[28] Niru Maheswaranathan, Alex Williams, Matthew Golub, Surya Ganguli, and David Sussillo. "Universality and individuality in neural dynamics across large populations of recurrent networks." In: *Advances in neural information processing systems*. 2019, pp. 15629–15641.

[29] Guangyu Robert Yang, Madhura R Joglekar, H Francis Song, William T Newsome, and Xiao-Jing Wang. "Task representations in neural networks trained to perform many cognitive tasks." In: *Nature neuroscience* 22.2 (2019), pp. 297–306.

[30] David Sussillo, Mark M Churchland, Matthew T Kaufman, and Krishna V Shenoy. "A neural network that finds a naturalistic solution for the production of muscle activity." In: *Nature neuroscience* 18.7 (2015), pp. 1025–1033.

[31] H Francis Song, Guangyu R Yang, and Xiao-Jing Wang. "Training excitatory-inhibitory recurrent neural networks for cognitive tasks: a simple and flexible framework." In: *PLoS computational biology* 12.2 (2016), e1004792.

[32] Nelson Cowan. "What are the differences between long-term, short-term, and working memory?" In: *Progress in brain research* 169 (2008), pp. 323–338.

[33] Shintaro Funahashi, Charles J Bruce, and Patricia S Goldman-Rakic. "Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex." In: *Journal of neurophysiology* 61.2 (1989), pp. 331–349.

[34] Daniel J Amit, Stefano Fusi, and Volodya Yakovlev. "Paradigmatic working memory (attractor) cell in IT cortex." In: *Neural computation* 9.5 (1997), pp. 1071–1092.

[35] Albert Compte, Nicolas Brunel, Patricia S Goldman-Rakic, and Xiao-Jing Wang. "Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model." In: *Cerebral cortex* 10.9 (2000), pp. 910–923.

[36] Clayton E Curtis and Mark D'Esposito. "Persistent activity in the prefrontal cortex during working memory." In: *Trends in cognitive sciences* 7.9 (2003), pp. 415–423.

[37] Eelke Spaak, Kei Watanabe, Shintaro Funahashi, and Mark G Stokes. "Stable and dynamic coding for working memory in primate prefrontal cortex." In: *Journal of Neuroscience* 37.27 (2017), pp. 6503–6516.

[38] Mitchell R Riley and Christos Constantinidis. "Role of prefrontal persistent activity in working memory." In: *Frontiers in systems neuroscience* 9 (2016), p. 181.

[39] Xiao-Jing Wang. "Synaptic reverberation underlying mnemonic persistent activity." In: *Trends in neurosciences* 24.8 (2001), pp. 455–463.

[40] John D Murray, Alberto Bernacchia, Nicholas A Roy, Christos Constantinidis, Ranulfo Romo, and Xiao-Jing Wang. "Stable population coding for working memory coexists with heterogeneous neural dynamics in prefrontal cortex." In: *Proceedings of the National Academy of Sciences* 114.2 (2017), pp. 394–399.

[41] Christopher D Harvey, Philip Coen, and David W Tank. "Choice-specific sequences in parietal cortex during a virtual-navigation decision task." In: *Nature* 484.7392 (2012), pp. 62–68.

[42] Shaul Druckmann and Dmitri B Chklovskii. "Neuronal circuits underlying persistent representations despite time varying activity." In: *Current Biology* 22.22 (2012), pp. 2095–2103.

[43] Xiaofan Zhang, Hu Yi, Wenwen Bai, and Xin Tian. "Dynamic trajectory of multiple single-unit activity during working memory task in rats." In: *Frontiers in computational neuroscience* 9 (2015), p. 117.

[44] Hugo Touchette and Seth Lloyd. "Information-theoretic approach to the study of control systems." In: *Physica A: Statistical Mechanics and its Applications* 331.1-2 (2004), pp. 140–172.

[45] Christoph Salge, Cornelius Glackin, and Daniel Polani. "Empowerment–an introduction." In: *Guided Self-Organization: Inception*. Springer, 2014, pp. 67–114.

[46] Shakir Mohamed and Danilo Jimenez Rezende. "Variational information maximisation for intrinsically motivated reinforcement learning." In: *Advances in neural information processing systems*. 2015, pp. 2125–2133.

[47] Tobias Jung, Daniel Polani, and Peter Stone. "Empowerment for continuous agent?environment systems." In: *Adaptive Behavior* 19.1 (2011), pp. 16–39.

[48] Christoph Salge, Cornelius Glackin, and Daniel Polani. "Approximation of empowerment in the continuous domain." In: *Advances in Complex Systems* 16.02n03 (2013), p. 1250079.

[49] Richard Blahut. "Computation of channel capacity and rate-distortion functions." In: *IEEE transactions on Information Theory* 18.4 (1972), pp. 460–473.

[50] Joseph A O?Sullivan. "Alternating minimization algorithms: from Blahut-Arimoto to expectation-maximization." In: *Codes, curves, and signals*. Springer, 1998, pp. 173–192.

[51] Ishmael Belghazi, Sai Rajeswar, Aristide Baratin, R Devon Hjelm, and Aaron Courville. "MINE: mutual information neural estimation." In: *arXiv preprint arXiv:1801.04062* (2018).

[52] Tapani Raiko and Matti Tornio. "Variational Bayesian learning of nonlinear hidden state-space models for model predictive control." In: *Neurocomputing* 72.16-18 (2009), pp. 3704–3712.

[53] Rajesh Ranganath, Sean Gerrish, and David Blei. "Black box variational inference." In: *Artificial Intelligence and Statistics*. 2014, pp. 814–822.

[54] Daniel Ritchie, Paul Horsfall, and Noah D Goodman. "Deep amortized inference for probabilistic programs." In: *arXiv preprint arXiv:1610.05735* (2016).

[55] Rui Shu, Hung H Bui, Shengjia Zhao, Mykel J Kochenderfer, and Stefano Ermon. "Amortized Inference Regularization." In: *arXiv preprint arXiv:1805.08913* (2018).

[56] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes." In: *arXiv preprint arXiv:1312.6114* (2013).

[57] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic back-propagation and approximate inference in deep generative models." In: *arXiv preprint arXiv:1401.4082* (2014).

[58] Hugh R Wilson and Jack D Cowan. "Excitatory and inhibitory interactions in localized populations of model neurons." In: *Biophysical journal* 12.1 (1972), pp. 1–24.

[59] Hugh R Wilson and Jack D Cowan. "A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue." In: *Kybernetik* 13.2 (1973), pp. 55–80.

[60] Peter Dayan, L Abbott, et al. "Theoretical neuroscience: computational and mathematical modeling of neural systems." In: *Journal of Cognitive Neuroscience* 15.1 (2003), pp. 154–155.

[61] Moritz Helmstaedter, Kevin L Briggman, Srinivas C Turaga, Viren Jain, H Sebastian Seung, and Winfried Denk. "Connectomic reconstruction of the inner plexiform layer in the mouse retina." In: *Nature* 500.7461 (2013), pp. 168–174.

[62] David Zipser and Richard A Andersen. "A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons." In: *Nature* 331.6158 (1988), pp. 679–684.

[63] Jong Chan Park, Jung Won Bae, Jieun Kim, and Min Whan Jung. "Dynamically changing neuronal activity supporting working memory for predictable and unpredictable durations." In: *Scientific reports* 9.1 (2019), pp. 1–10.

[64] Richard H Bauer and Joaquin M Fuster. "Delayed-matching and delayed-response deficit from cooling dorsolateral prefrontal cortex in monkeys." In: *Journal of comparative and physiological psychology* 90.3 (1976), p. 293.

[65] Pierre Enel, Emmanuel Procyk, René Quilodran, and Peter Ford Dominey. "Reservoir computing properties of neural dynamics in prefrontal cortex." In: *PLoS computational biology* 12.6 (2016), e1004967.

[66] Razvan Pascanu and Herbert Jaeger. "A neurodynamical model for working memory." In: *Neural networks* 24.2 (2011), pp. 199–207.

[67] Valerio Mante, David Sussillo, Krishna V Shenoy, and William T Newsome. "Context-dependent computation by recurrent dynamics in prefrontal cortex." In: *nature* 503.7474 (2013), pp. 78–84.

[68] John J Hopfield. "Neural networks and physical systems with emergent collective computational abilities." In: *Proceedings of the national academy of sciences* 79.8 (1982), pp. 2554–2558.

[69] Ronald J Williams and David Zipser. "A learning algorithm for continually running fully recurrent neural networks." In: *Neural computation* 1.2 (1989), pp. 270–280.

[70] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors." In: *nature* 323.6088 (1986), pp. 533–536.

[71] Mantas Lukoševičius and Herbert Jaeger. "Reservoir computing approaches to recurrent neural network training." In: *Computer Science Review* 3.3 (2009), pp. 127–149.

[72] Thomas Natschläger, Wolfgang Maass, and Henry Markram. "The" liquid computer": A novel strategy for real-time computing on time series." In: *Special issue on Foundations of Information Processing of TELEMATIK* 8.ARTICLE (2002), pp. 39–43.

[73] Herbert Jaeger. "The ?echo state? approach to analysing and training recurrent neural networks-with an erratum note." In: *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148.34 (2001), p. 13.

[74] David Sussillo and Larry F Abbott. "Generating coherent patterns of activity from chaotic neural networks." In: *Neuron* 63.4 (2009), pp. 544–557.

[75] James Martens and Ilya Sutskever. "Learning recurrent neural networks with hessian-free optimization." In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. Citeseer. 2011, pp. 1033–1040.

[76] Haim Sompolinsky, Andrea Crisanti, and Hans-Jurgen Sommers. "Chaos in random neural networks." In: *Physical review letters* 61.3 (1988), p. 259.

[77] Francesca Mastrogiuseppe and Srdjan Ostojic. "Linking connectivity, dynamics, and computations in low-rank recurrent neural networks." In: *Neuron* 99.3 (2018), pp. 609–623.

[78] Alan Edelman and N Raj Rao. "Random matrix theory." In: *Acta numerica* 14 (2005), p. 233.

[79] Kanaka Rajan and Larry F Abbott. "Eigenvalue spectra of random matrices for neural networks." In: *Physical review letters* 97.18 (2006), p. 188104.

[80] Kanaka Rajan, LF Abbott, and Haim Sompolinsky. "Stimulus-dependent suppression of chaos in recurrent neural networks." In: *Physical Review E* 82.1 (2010), p. 011903.

[81] Kenneth D Harris and Thomas D Mrsic-Flogel. "Cortical connectivity and sensory coding." In: *Nature* 503.7474 (2013), pp. 51–58.

[82] Friedrich Schuessler, Alexis Dubreuil, Francesca Mastrogiuseppe, Srdjan Ostojic, and Omri Barak. "Dynamics of random recurrent networks with correlated low-rank structure." In: *Physical Review Research* 2.1 (2020), p. 013111.

[83] Peter Dayan. "Reinforcement learning." In: *Stevens' handbook of experimental psychology* (2002).

[84] Rishidev Chaudhuri and Ila Fiete. "Computational principles of memory." In: *Nature neuroscience* 19.3 (2016), p. 394.

[85] Omri Barak and Misha Tsodyks. "Working models of working memory." In: *Current opinion in neurobiology* 25 (2014), pp. 20–24.

[86] Joel Zylberberg and Ben W Strowbridge. "Mechanisms of persistent activity in cortical circuits: possible neural substrates for working memory." In: *Annual review of neuroscience* 40 (2017).

[87] Gustavo Deco and Edmund T Rolls. "Attention and working memory: a dynamical model of neuronal activity in the prefrontal cortex." In: *European Journal of Neuroscience* 18.8 (2003), pp. 2374–2390.

[88] Warasinee Chaisangmongkon, Sruthi K Swaminathan, David J Freedman, and Xiao-Jing Wang. "Computing by robust transience: how the fronto-parietal network performs sequential, category-based decisions." In: *Neuron* 93.6 (2017), pp. 1504–1517.

[89] Giulio Bondanelli and Srdjan Ostojic. "Coding with transient trajectories in recurrent neural networks." In: *PLoS computational biology* 16.2 (2020), e1007655.

[90] Timo Nachstedt and Christian Tetzlaff. "Working memory requires a combination of transient and attractor-dominated dynamics to process unreliably timed inputs." In: *Scientific reports* 7.1 (2017), pp. 1–14.

[91] Pierre Enel, Joni D Wallis, and Erin L Rich. "Stable and dynamic representations of value in the prefrontal cortex." In: *Elife* 9 (2020), e54313.

[92] Sean E Cavanagh, John P Towers, Joni D Wallis, Laurence T Hunt, and Steven W Kennerley. "Reconciling persistent and dynamic hypotheses of working memory coding in prefrontal cortex." In: *Nature communications* 9.1 (2018), pp. 1–16.

[93] A Emin Orhan and Wei Ji Ma. "A diverse range of factors affect the nature of neural representations underlying short-term memory." In: *Nature neuroscience* 22.2 (2019), pp. 275–283.

[94] Katsuyuki Sakai. "Task set and prefrontal cortex." In: *Annu. Rev. Neurosci.* 31 (2008), pp. 219–245.

[95] Omri Barak, David Sussillo, Ranulfo Romo, Misha Tsodyks, and LF Abbott. "From fixed points to chaos: three models of delayed discrimination." In: *Progress in neurobiology* 103 (2013), pp. 214–222.

[96] Chen Beer and Omri Barak. "One step back, two steps forward: interference and learning in recurrent neural networks." In: *Neural Computation* 31.10 (2019), pp. 1985–2003.

[97] Niru Maheswaranathan, Alex Williams, Matthew Golub, Surya Ganguli, and David Sussillo. "Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics." In: *Advances in neural information processing systems*. 2019, pp. 15696–15705.

[98] Daniel B Rubin, Stephen D Van Hooser, and Kenneth D Miller. "The stabilized supralinear network: a unifying circuit motif underlying multi-input integration in sensory cortex." In: *Neuron* 85.2 (2015), pp. 402–417.

[99] Jane X Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Demis Hassabis, and Matthew Botvinick. "Prefrontal cortex as a meta-reinforcement learning system." In: *Nature neuroscience* 21.6 (2018), pp. 860–868.

[100] Martín Abadi et al. "Tensorflow: a system for large-scale machine learning." In: *OSDI*. Vol. 16. 2016, pp. 265–283.

[101] Joshua V Dillon et al. “TensorFlow Distributions.” In: *arXiv preprint arXiv:1711.10604* (2017).

Appendix A

Model Parameters for Empowerment Maximization

In this section, we provide the details of parameters used in our simulations. Particularly, the dynamical model parameters for pendulum and the Wilson-Cowan model are obtained from the corresponding references [22, 58]. For the remaining, i.e. linear and nonlinear systems, the model parameter are chosen to highlight the key idea of empowerment maximization with respect to the system dynamics.

A.0.1 Dynamical Models Parameters:

In Fig. 2.4

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

In Fig. 2.5, the parameter set-up for pendulum are the same as the model provided in [22].

In Fig. 2.6, $c_1 = 12, c_2 = 4, c_3 = 13, c_4 = 11, a_e = 1.2, a_i = 1, \theta_e = 2.8, \theta_i = 4$ and $P = Q = 0$. We have set $[I_e, I_i] = [u_1, u_2]$.

We obtained excitatory and inhibitory nullclines by setting $\dot{s}_e = 0$ and $\dot{s}_i = 0$, respectively. The fixed points of the system are located where the nullclines intersect.

In Fig. 2.7 (a)

$$\mathbf{A} = \begin{bmatrix} -0.5 & 0 \\ 0 & -0.5 \end{bmatrix}$$

In Fig. 2.7 (b)

$$\mathbf{K} = \begin{bmatrix} 0.626 & 0.032 \\ 0.022 & 0.674 \end{bmatrix}$$

In Fig. 2.8(a)

$$f(\boldsymbol{s}) = \begin{cases} s_x^2 - 0.5 \\ s_y^2 - 0.5 \end{cases}$$

In Fig. 2.8 (b)

$$K = \begin{bmatrix} 1.533 & -0.414 \\ -0.430 & 1.566 \end{bmatrix}$$

In Fig. 2.9, $c_1 = 16, c_2 = 12, c_3 = 15, c_4 = 4, a_e = 1.3, a_i = 2, \theta_e = 4\theta_i = 3.7$. We have set $[I_e, I_i] = [u_1, u_2]$. In Fig. 2.9(b), $k = P$.

A.0.2 Simulation Parameters:

We have used TensorFlow [100, 101] framework for our implementations. The NNs structure for our simulations are as follows:

Linear system: $d = 2$,

$\omega(\mathrm{u}_t|\mathrm{s}_t) : 16 \text{ ELU} + 16 \text{ ELU} + \{d \text{ idenity}, d \exp\}$

$q(\mathrm{u}_t|\mathrm{s}_t, \mathrm{s}_{t+n}) : 16 \text{ ELU} + 16 \text{ ELU} + \{d \text{ idenity}, d \exp\}$

Pendulum: $d = 2$,

$\omega(\mathrm{u}_t|\mathrm{s}_t) : 128 \tanh + 128 \tanh + 128 \tanh + 128 \tanh + \{1 \text{ idenity}, 1 \exp\}$

$q(\mathrm{u}_t|\mathrm{s}_t, \mathrm{s}_{t+n}) : 128 \tanh + 128 \tanh + 128 \tanh + 128 \tanh + \{1 \text{ idenity}, 1 \exp\}$

Nonlinear system: We have used the same structure as the linear one.

The Wilson-Cowan model: $d = 2$,

$\omega(\mathrm{u}_t|\mathrm{s}_t) : 16 \text{ ELU} + 16 \text{ ELU} + 16 \text{ ELU} + \{d \text{ idenity}, d \exp\}$

$q(\mathrm{u}_t|\mathrm{s}_t, \mathrm{s}_{t+n}) : 16 \text{ ELU} + 16 \text{ ELU} + 16 \text{ ELU} + \{d \text{ idenity}, d \exp\}$

Appendix B

Details of Update Rules

In this section, we provide the mathematical details of obtaining parameter update rules as per FORCE method.

The goal here is to minimize the squared error during training time T, with respect to readout weights $\mathbf{W}_o$:

$$E(t) = \frac{1}{2}\int_0^T e(t)^2 dt \tag{B.1}$$

$$\frac{\partial E(t)}{\partial \mathbf{W}_{o_i}} = \int_0^T e(t)\frac{\partial z(t)}{\partial \mathbf{W}_{o_i}} dt \tag{B.2}$$

$$\frac{\partial z(t)}{\partial \mathbf{W}_{o_i}} = \mathbf{r}_i(t) + \sum_{m=1}^{N} \mathbf{W}_{o_m}\mathbf{r}_i'(t)\frac{\partial \mathbf{x}_m(t)}{\partial \mathbf{W}_{o_i}} \tag{B.3}$$

Difficulties in training RNNs arise from the taxing computations of $\frac{\partial \mathbf{x}_m(t)}{\partial \mathbf{W}_{o_i}}$ through time. In contrast to back propagation-based methods of training RNN, such as BPTT, it is assumed

that $\frac{\partial \mathbf{x}_m(t)}{\partial \mathbf{W}_{o_i}}$ is close to zero, throughout training. This assumption is valid as long as $\epsilon(t)$ is close to zero, where:

$$z(t) = f(t) + \epsilon(t) \tag{B.4}$$

By considering

$$z(t) = \mathbf{W}_o(t - \Delta t)\mathbf{r}(t) \tag{B.5}$$

where Δt is the update time step, we obtain:

$$e_-(t) = \mathbf{W}_o(t - \Delta t)\mathbf{r}(t) - f(t) \tag{B.6}$$

Using (B.3), we have the gradient-based update rule for $\mathbf{W}_o$ as:

$$\mathbf{W}_o(t) = \mathbf{W}_o(t - \Delta t) - \eta(t)e_-(t)\mathbf{r}(t) \tag{B.7}$$

where $\eta(t)$ is a time-dependent learning rate.

Using (B.7) and (B.6), we can rewrite $z(t)$ as :

$$z(t) = \mathbf{W}_o(t)f(t) = f(t) + e_-(t)(1 - \eta(t)\mathbf{r}(t)^T(t)\mathbf{r}(t)) \tag{B.8}$$

Thus, $\epsilon(t)$ will be much smaller than $1 - \eta(t)\mathbf{r}(t)^T(t)\mathbf{r}(t) \ll 1$ and thus remains as a valid assumption throughout training. Then, recursive least square method can be used to obtain the update rules presented in Chapter 3.

Appendix C

Update Rules for Training RNNs to Perform SPM Task Using Reinforced Regression

Update rule for modifying output readout weights and the inverse correlation matrix:

$$\begin{aligned} \mathbf{W}_o(t) &= \mathbf{W}_o(t-\Delta t) - e_o(t)P(t)\mathbf{r}(t) \\ \boldsymbol{P}(t) &= \boldsymbol{P}(t-\Delta t) - \frac{\boldsymbol{P}(t-\Delta t)\mathbf{r}(t)\mathbf{r}^T(t)\boldsymbol{P}(t-\Delta t)}{1+\mathbf{r}^T(t)\boldsymbol{P}(t-\Delta t)\mathbf{r}(t)} \end{aligned} \tag{C.1}$$

$$\boldsymbol{P}(t) = \int_0^T \mathbf{r}(t)\mathbf{r}^T(t)dt + \alpha \boldsymbol{I}_N \tag{C.2}$$

Update rule for modifying dummy weights we have:

$$\mathbf{W}_d(t) = \mathbf{W}_d(t-\Delta t) - e_d(t)P(t)\mathbf{r}(t). \tag{C.3}$$

Note that, using reinforced regression we update the inverse correlation matrix during training intervals shown in . Δt denotes the update time steps.

Appendix D

Model Parameters for SPM Task Learning and Dynamical Analysis

In this section, we provide the details of parameters used in our simulations.

In SPM task, we set the stimulus interval to 100 time steps, delay intervals 50 time steps and response interval to 50 time steps. From the low dimensional representation of MNIST digits (obtained from training Variational Auto Encoder) we selected digits 1 and 0 as the stimuli. We encoded the summation outcomes (i.e. 0, 1, 2) as 0.5, 1 and 1.5 respectively for training the networks. We encoded dummy outputs as the two dimensional mean vector for each digit representation.

For all simulations the value of α is initialized to 1 and $\boldsymbol{P}$ is initialized to identity matrix for both readout and dummy weights training. The number of neurons $N = 1000$ and elements of $\mathbf{W}_o$ and $\mathbf{W}_d$ are initialized to zero. Input weights were drawn randomly from zero mean Gaussian distribution with variance 50. For 4 different initialization seeds we considered all possible combination of feasible values for g, σ_f and sparsity (in Fig. 4.12). The value of σ_f were chosen

proportional to the size of network, i.e. $\sigma_f \in \{\frac{1}{1.5N}, \frac{1}{N}, \frac{1}{0.5N}, \frac{1}{0.1N}, \frac{1}{0.02N}, \frac{1}{0.005N}, \frac{1}{0.002N}, \frac{1}{0.0001N}\}$. Training was terminated if the average root mean squared error between target and output was less that 0.01 for all trials.