Neural Computation via Linearised Dynamics of RNNs and Linear Readouts

Contents

1 Introduction

We use theoretical analysis and simulation to study the dynamics of input-driven random recurrent neural networks (RNNs), aiming to provide an interpretable account of how the emergent geometry of neural activity supports computation — including tasks in decision making, context dependence, and memory recall. We derive closed-form expressions for the steady states and covariances of network activity under given input regimes, by linearising the system around fixed points and solving the associated Lyapunov equation. The results of this approach, which align closely with simulations of the discrete-time dynamics and dynamical mean-field theory, enable a geometric interpretation of network responses under different input regimes, and clarify how these geometries facilitate computation (such as memory and signal separation) via an *interpretable* linear readout layer.

2 Theoretical Framework

2.1 Network Dynamics

We consider the dynamics of a discrete-time recurrent neural network with N units governed by the update equation:

$$\boldsymbol{x}(t+1) = \tanh\left(\boldsymbol{W}\boldsymbol{x}(t) + \boldsymbol{W}_{\text{in}}\boldsymbol{u}(t)\right), \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^N$ is the network state at time t, $\boldsymbol{W} \in \mathbb{R}^{N\times N}$ is the recurrent connectivity matrix, $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{N\times M}$ is the input weight matrix, and $\boldsymbol{u}(t) \in \mathbb{R}^M$ is the input. The entries of $\boldsymbol{W}$ are drawn independently from a Gaussian distribution $\mathcal{N}(0, \frac{1}{N})$, and then the matrix is rescaled to have a desired spectral radius ρ. This may be related to the standard formulation in dynamical mean-field theory (DMFT) (see Section 2.6), where the entries of the weight matrix are drawn independently from a Gaussian distribution $\mathcal{N}(0, \frac{g^2}{N})$, and the effective coupling strength is governed by the parameter g. In our formulation of $\boldsymbol{W}$, we first sample $W_{ij} \sim \mathcal{N}(0, \frac{1}{N})$, and after sampling, we normalise $\boldsymbol{W}$ by dividing by its largest eigenvalue modulus and then multiplying by ρ, so that the final matrix satisfies $\rho(\boldsymbol{W}) = \rho$. In the

large limit $N \to \infty$ in DMFT, we have that $\rho \approx g$. DMFT predicts a phase transition at $g = 1$: for $g < 1$, the autonomous network settles to a stable fixed point, while for $g > 1$, it exhibits chaotic activity. By focusing on the $\rho < 1$ regime, our analysis investigates the formation and geometry of stable representations. The input weights $\boldsymbol{W}_{\text{in}}$ are generally drawn independently from a Gaussian distribution $\mathcal{N}(0, \frac{1}{N})$ and scaled by an input scaling factor ι, which will be elaborated upon in Section 2.3. However, in the case of implementing DMFT (Section 2.6 and 3), we will set $\boldsymbol{W}_{\text{in}}$ to the particular case of $\boldsymbol{W}_{\text{in}} = \frac{\iota}{\sqrt{N}}\mathbf{1}$, where $\mathbf{1}$ is a vector of ones, to avoid an ensemble-average mean activity of 0.

2.2 Fixed Point Analysis

To characterise the steady-state response of the network to a stochastic (Gaussian distributed) input with mean $\boldsymbol{\mu} \in \mathbb{R}^M$ and covariance $\boldsymbol{\Sigma}_{in} \in \mathbb{R}^{M \times M}$, we analyse the fixed points of the input-driven system. For theoretical calculation, we consider the input as constant in time, $\boldsymbol{u}(t) = \boldsymbol{\mu}$. A fixed point $\boldsymbol{x}^*$ satisfies:

$$\boldsymbol{x}^* = \tanh\left(\boldsymbol{W}\boldsymbol{x}^* + \boldsymbol{W}_{\text{in}}\boldsymbol{\mu}\right), \tag{2}$$

We can approximate the fixed point by linearising this equation around $\boldsymbol{x}^*$, under assumptions and constraints discussed in Section 2.3, yielding:

$$\boldsymbol{x}^* \approx \left(\boldsymbol{I} - \boldsymbol{W}\right)^{-1} \boldsymbol{W}_{\text{in}}\boldsymbol{\mu}. \tag{3}$$

This provides an analytical estimate of the steady-state network activity vector in response to a constant input mean $\boldsymbol{\mu}$.

2.3 Assumptions and Constraints of the Linearisation Approach

The linearisation approach around fixed points in the preceding section is valid under certain assumptions and constraints. In particular, it relies on the approximation $\tanh(z) \approx z$, which holds when z is small. In our setup, we have $\tanh(\boldsymbol{W}\boldsymbol{x}^* + \boldsymbol{W}_{in}\boldsymbol{\mu})$, and so we want to keep $\|\boldsymbol{W}\boldsymbol{x}^* + \boldsymbol{W}_{in}\boldsymbol{\mu}\|$ close to 0.

We have that $\boldsymbol{x}^* \approx (\boldsymbol{I} - \boldsymbol{W})^{-1}\boldsymbol{W}_{\text{in}}\boldsymbol{\mu}$ (assuming that $\boldsymbol{I} - \boldsymbol{W}$ is invertible, which is valid for $\rho < 1$), and that $\|\boldsymbol{W}_{in}\boldsymbol{\mu}\| \approx \iota\|\boldsymbol{\mu}\|$, so we can write

$$\begin{aligned}
\|\boldsymbol{W}\boldsymbol{x}^* + \boldsymbol{W}_{\text{in}}\boldsymbol{\mu}\| &\approx \|\boldsymbol{W}(\boldsymbol{I} - \boldsymbol{W})^{-1}\boldsymbol{W}_{\text{in}}\boldsymbol{\mu} + \boldsymbol{W}_{\text{in}}\boldsymbol{\mu}\| \\
&\leq \|\boldsymbol{W}(\boldsymbol{I} - \boldsymbol{W})^{-1}\boldsymbol{W}_{\text{in}}\boldsymbol{\mu}\| + \|\boldsymbol{W}_{\text{in}}\boldsymbol{\mu}\| \\
&\leq \|\boldsymbol{W}(\boldsymbol{I} - \boldsymbol{W})^{-1}\| \cdot \|\boldsymbol{W}_{\text{in}}\boldsymbol{\mu}\| + \|\boldsymbol{W}_{\text{in}}\boldsymbol{\mu}\| \\
&= \|\boldsymbol{W}_{\text{in}}\boldsymbol{\mu}\| \cdot \left(\|\boldsymbol{W}(\boldsymbol{I} - \boldsymbol{W})^{-1}\| + 1\right) \\
&\leq \|\boldsymbol{W}_{\text{in}}\boldsymbol{\mu}\| \cdot \left(\|\boldsymbol{W}\| \cdot \|(\boldsymbol{I} - \boldsymbol{W})^{-1}\| + 1\right)
\end{aligned}$$

For our purposes here, we can consider the comparison between $\|\boldsymbol{W}\| \cdot \|(\boldsymbol{I} - \boldsymbol{W})^{-1}\|$ and $\frac{\rho}{1-\rho}$, where ρ is the maximum eigenvalue (the spectral radius) of $\boldsymbol{W}$. This is appropriate, as given a non-normal matrix $\boldsymbol{A}$, $\|\boldsymbol{A}\| \gg \rho(\boldsymbol{A})$, which means that if $\frac{\rho(\boldsymbol{A})}{1-\rho(\boldsymbol{A})}$ upper bounds $\mathcal{A}$, then $\|\boldsymbol{A}\| \cdot \|(\boldsymbol{I} - \boldsymbol{A})^{-1}\|$ does too. Therefore, let

$$\|\boldsymbol{W}\boldsymbol{x}^* + \boldsymbol{W}_{\text{in}}\boldsymbol{\mu}\| \lesssim \iota\|\boldsymbol{\mu}\| \cdot \left(\frac{\rho}{1-\rho} + 1\right)$$

and, provided we ensure $\iota\|\boldsymbol{\mu}\| \cdot \left(\frac{\rho}{1-\rho} + 1\right) \ll 1$, then $\|\boldsymbol{W}\boldsymbol{x}^* + \boldsymbol{W}_{\text{in}}\boldsymbol{\mu}\| \ll 1$, and the linearisation approach is suitable. Using these constraints, we can formulate a restriction on the input scaling factor ι:

$$\iota\|\boldsymbol{\mu}\| \cdot \left(\frac{\rho}{1-\rho} + 1\right) \ll 1 \Rightarrow \iota \ll \frac{1-\rho}{\|\boldsymbol{\mu}\|}$$

2.4 Stability Analysis

To assess the local stability of a fixed point $\boldsymbol{x}^*$ under input $\boldsymbol{u}(t) = \boldsymbol{\mu} \in \mathbb{R}^M$, we compute the Jacobian of the network dynamics linearised around $\boldsymbol{x}^*$. The Jacobian $\boldsymbol{J} \in \mathbb{R}^{N \times N}$ at the fixed point is given by:

$$\boldsymbol{J} = \text{diag}\left[1 - \tanh^2\left(\boldsymbol{W}\boldsymbol{x}^* + \boldsymbol{W}_{\text{in}}\boldsymbol{\mu}\right)\right]\boldsymbol{W}, \tag{4}$$

where the diagonal matrix contains the derivatives of the tanh nonlinearity at each neuron. The spectral radius of the Jacobian $\rho(\boldsymbol{J})$ is then calculated to assess local stability. A fixed point is locally stable if $\rho(\boldsymbol{J}) < 1$, implying all perturbations decay over time, ensuring that the system returns to $\boldsymbol{x}^*$ after small perturbations. We note that in our construction of $\boldsymbol{W}$ with $\rho < 1$, and $\boldsymbol{W}_{in}$ with small scaling factor $\iota \ll 1$, as mentioned in the preceding section, it means that for $\tanh^2(z_i)$, most z_i are close to zero, and

$$\tanh^2(z_i) \approx z_i^2 \ll 1 \quad \Rightarrow \quad 1 - \tanh^2(z_i) \approx 1.$$

This gives the approximation

$$\text{diag}[1 - \tanh^2(\boldsymbol{z})] \approx \boldsymbol{I} \quad \Rightarrow \quad \boldsymbol{J} \approx \boldsymbol{W}.$$

Thus, the Jacobian $\boldsymbol{J}$ has spectral radius approximately equal to that of $\boldsymbol{W}$:

$$\rho(\boldsymbol{J}) \approx \rho(\boldsymbol{W}).$$

This holds particularly when the network operates near the origin and receives weak or constant input. Since we initialise our networks with $\rho < 1$, the fixed points calculated by the approach outlined in Section 2.2 have local stability.

2.5 Theoretical Covariance Calculation

To understand the *variability* of the dynamics around the theoretical fixed points calculated above, we consider the Jacobian of the system dynamics (Equation 4) around $\boldsymbol{x}^*$. Assuming stationarity, the covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{N \times N}$ of the state $\boldsymbol{x}(t)$ satisfies the discrete-time Lyapunov equation:

$$\boldsymbol{\Sigma} = \boldsymbol{J}\boldsymbol{\Sigma}\boldsymbol{J}^\top + \boldsymbol{W}_{\text{in}}\boldsymbol{\Sigma}_{\text{in}}\boldsymbol{W}_{\text{in}}^\top. \tag{5}$$

where $\boldsymbol{\Sigma}_{\text{in}} \in \mathbb{R}^{M \times M}$ is the input covariance matrix. Solving this equation yields a theoretical prediction of the variance around each fixed point. We use vectorisation which transforms the equation into a linear system that can be solved by linear algebra techniques. We start

with the discrete-time Lyapunov equation:

$$\boldsymbol{\Sigma} = \boldsymbol{J}\boldsymbol{\Sigma}\boldsymbol{J}^\top + \boldsymbol{Q}, \tag{6}$$

where $\boldsymbol{Q} = \boldsymbol{W}_{\text{in}}\boldsymbol{\Sigma}_{\text{in}}\boldsymbol{W}_{\text{in}}^\top$. Applying the $\text{vec}(\cdot)$ operator, which stacks the columns of a matrix into a vector, and using the property

$$\text{vec}(\boldsymbol{ABC}) = (\boldsymbol{C}^\top \otimes \boldsymbol{A})\,\text{vec}(\boldsymbol{B}), \tag{7}$$

where $\otimes$ denotes the Kronecker product, we obtain:

$$\text{vec}(\boldsymbol{\Sigma}) = \text{vec}(\boldsymbol{J}\boldsymbol{\Sigma}\boldsymbol{J}^\top) + \text{vec}(\boldsymbol{Q}) = (\boldsymbol{J} \otimes \boldsymbol{J})\,\text{vec}(\boldsymbol{\Sigma}) + \text{vec}(\boldsymbol{Q}). \tag{8}$$

Rearranging terms gives a linear system:

$$(\boldsymbol{I} - \boldsymbol{J} \otimes \boldsymbol{J})\,\text{vec}(\boldsymbol{\Sigma}) = \text{vec}(\boldsymbol{Q}), \tag{9}$$

where $\boldsymbol{I}$ is the $N^2 \times N^2$ identity matrix. Thus, the solution for $\text{vec}(\boldsymbol{\Sigma})$ is

$$\text{vec}(\boldsymbol{\Sigma}) = (\boldsymbol{I} - \boldsymbol{J} \otimes \boldsymbol{J})^{-1}\,\text{vec}(\boldsymbol{Q}). \tag{10}$$

Finally, reshaping $\text{vec}(\boldsymbol{\Sigma})$ into a $N \times N$ matrix yields the desired covariance matrix $\boldsymbol{\Sigma}$. It is important to note that, even if the input covariance is isotropic, the resulting state covariance $\boldsymbol{\Sigma}$ will generally not be isotropic. This is because the Jacobian $\boldsymbol{J}$, which reflects the recurrent connectivity, selectively amplifies variance along certain directions, creating anisotropy in the network activity. Thus, recurrent dynamics shape the covariance geometry away from isotropy.

2.6 Dynamical Mean-Field Theory

While the preceding sections provide exact solutions for the fixed points and covariances of a *single realisation* of a network, dynamical mean-field theory [1, 2] offers a powerful,

complementary approach. DMFT characterises the macroscopic statistical properties of an *ensemble* of random networks in the large $N \to \infty$ limit. It provides a benchmark for the typical behaviour of such systems, allowing us to validate that our finite-N results are not artifacts of a specific random draw, but are representative of the general case.

The central assumption of DMFT is that in the large N limit, the total input h_i to any given neuron becomes a Gaussian random variable, due to the central limit theorem applied to the sum of many weakly correlated recurrent inputs. Instead of tracking N individual activities, DMFT solves for a few self-consistent "order parameters" that describe the entire population [2]. For a network driven by a constant mean input, the key order parameters are:

- The **mean activity**, $m = \mathbb{E}[x_i]$, which represents the average activation across the neural population.
- The **mean-squared activity**, $q = \mathbb{E}[x_i^2]$, which represents the total power of the neural activity, including both the mean and its fluctuations.

From these, the average variance of neural activity can be found as $\mathrm{Var}(x_i) = q - m^2$. While input weights $\boldsymbol{W}_{\text{in}}$ are often drawn from a random centred Gaussian or Uniform distribution, this leads to an ensemble-average mean activity of zero (in DMFT). For the DMFT-related analysis of this work, the input weights $\boldsymbol{W}_{\text{in}}$ are defined as a constant vector $\boldsymbol{W}_{\text{in}} = \frac{\iota}{\sqrt{N}}\mathbf{1}$, where $\mathbf{1}$ is a vector of ones. This choice ensures that the external input drives all neurons coherently, eliciting a non-zero mean network activity that can be directly compared with predictions from dynamical mean field theory.

These order parameters m, q are therefore determined by solving a set of self-consistent equations. First, the mean and variance of the Gaussian input field h are expressed in terms of m and q. For our network, where the recurrent gain is $g \approx \rho$ and the input vector is $\boldsymbol{W}_{\text{in}} = \frac{\iota}{\sqrt{N}}\mathbf{1}$, the moments of the field are:

$$\mathbb{E}[h] = \mathbb{E}[W_{\text{in},i}u_i] = \frac{\iota}{\sqrt{N}}\mu + gm \tag{11}$$

$$\mathrm{Var}(h) = g^2 q + \mathrm{Var}(W_{\text{in},i}u_i) = g^2 q + \frac{\iota^2}{N}\sigma_{\text{in}}^2 \tag{12}$$

where σ_{in}^2 is the variance of the input signal $u(t)$. The order parameters must, in turn, be generated by the network's response to this field. This closes the loop, yielding the self-consistency equations:

$$m = \int \mathcal{D}z \, \tanh\left(\mathbb{E}[h] + \sqrt{\text{Var}(h)}z\right) \tag{13}$$

$$q = \int \mathcal{D}z \, \tanh^2\left(\mathbb{E}[h] + \sqrt{\text{Var}(h)}z\right) \tag{14}$$

where $\mathcal{D}z = \frac{e^{-z^2/2}}{\sqrt{2\pi}}dz$ represents integration over a standard normal variable. These equations are typically solved numerically by iterating until convergence.

Connecting DMFT to our Analysis

We use the solutions to these DMFT equations, m_{DMFT} and q_{DMFT}, as a theoretical benchmark for the results from our finite-N simulations and our specific linearised theory. The connection between the order parameters is as follows:

- The **mean activity** from our theory and simulations should match the DMFT prediction:

$$m_{\text{DMFT}} \approx \frac{1}{N}\sum_{i=1}^{N} x_i^* = \text{mean}(\boldsymbol{x}^*) \tag{15}$$

  where $\boldsymbol{x}^*$ is the fixed-point vector calculated in Section 2.2.

- The **mean-squared activity** from DMFT corresponds to the sum of the mean-squared fixed point activity and the average variance of the fluctuations around it. This connects all of our theoretical quantities:

$$q_{\text{DMFT}} \approx \frac{1}{N}\left(\sum_{i=1}^{N}(x_i^*)^2 + \sum_{i=1}^{N}\Sigma_{ii}\right) = \frac{1}{N}\left(\|\boldsymbol{x}^*\|^2 + \text{Tr}(\boldsymbol{\Sigma})\right) \tag{16}$$

  where $\boldsymbol{\Sigma}$ is the covariance matrix from the Lyapunov equation in Section 2.5.

This comparison will allow us to rigorously validate our linearisation approach and demonstrate that our detailed geometric analysis captures the behaviour of a typical network from the ensemble described by DMFT.

2.7 Analysis of Network Timescales

The ability of a network to integrate inputs over time or maintain short-term memory is determined by its intrinsic timescales. A long timescale allows perturbations to persist, enabling computation on temporally extended signals. The dominant timescale can be derived from both the single-instance linearised theory and from DMFT.

Timescale from Linearised Theory

The local dynamics around a fixed point $\boldsymbol{x}^*$ are governed by the Jacobian $\boldsymbol{J}$ (Equation 4). A small perturbation $\delta\boldsymbol{x}(t)$ from the fixed point evolves according to the linear map $\delta\boldsymbol{x}(t+1) = \boldsymbol{J}\,\delta\boldsymbol{x}(t)$. The dynamics are limited by the slowest mode – the one whose corresponding eigenvalue of $\boldsymbol{J}$ has a magnitude closest to 1. The characteristic timescale τ_k of a mode k with eigenvalue λ_k is the time it takes for its amplitude to decay by a factor of $1/e$:

$$|\lambda_k|^{\tau_k} = e^{-1} \quad \Longrightarrow \quad \tau_k = -\frac{1}{\ln|\lambda_k|}. \tag{17}$$

The dominant, or longest, intrinsic timescale of the linearised network, τ_{lin}, is therefore set by the spectral radius of the Jacobian, which we denote $\rho_J = \rho(\boldsymbol{J})$:

$$\tau_{\text{lin}} = -\frac{1}{\ln(\rho_J)}. \tag{18}$$

This provides an exact expression for the timescale of the linearised system.

Approximate Scaling of the Timescale

To gain a more intuitive understanding of how this timescale behaves, we can derive an approximate scaling relationship for networks operating near the critical point, where $\rho_J \to 1$. In this regime, we can use the first-order Taylor expansion of the natural logarithm around $x = 1$, which is $\ln(x) \approx x - 1$.

Let $\rho_J = 1-\epsilon$, where ϵ is a small positive number representing the distance from criticality

$(0 < \epsilon \ll 1)$. Substituting this into the exact expression for the timescale (Equation 18):

$$\begin{aligned}\tau_{\text{lin}} &= -\frac{1}{\ln(1-\epsilon)} \\ &\approx -\frac{1}{(1-\epsilon)-1} \quad (\text{using } \ln(x) \approx x-1) \\ &= -\frac{1}{-\epsilon} = \frac{1}{\epsilon}.\end{aligned}$$

Since $\epsilon = 1 - \rho_J$, we arrive at the well-known scaling approximation [2]:

$$\tau_{\text{lin}} \approx \frac{1}{1-\rho_J}. \tag{19}$$

Furthermore, under the conditions of our linear analysis (Section 2.4), the Jacobian's spectral radius is close to that of the connectivity matrix, $\rho_J \approx \rho(\boldsymbol{W}) = \rho$. This yields the simple scaling law:

$$\tau \approx \frac{1}{1-\rho}. \tag{20}$$

This result establishes a direct and intuitive link between the network's spectral radius $(1-\rho)$ and its memory capacity (τ): as the network is tuned towards the "edge of chaos", its intrinsic timescale diverges with the inverse of its distance from the critical point. We validate this scaling relationship empirically in Section 3.

Timescale from Dynamical Mean-Field Theory

Within the DMFT framework, the intrinsic timescale can be extracted by analysing the decay of the population-averaged activity. For the zero-mean connectivity matrices used in our model, the mean activity m is driven purely by the external input. However, the decay of fluctuations around the mean is governed by recurrent dynamics. The two-time autocorrelation function, $C(\tau) = \mathbb{E}[x_i(t)x_i(t+\tau)]$, decays with a rate determined by the product of the gain g and the network's susceptibility χ. The susceptibility, $\chi = \int \mathcal{D}z \, \text{sech}^2(\mu_h + \sigma_h z)$, measures the average linear response of the neurons.

The effective decay factor for fluctuations is $g\chi$. The DMFT timescale τ_{DMFT} is defined

by $(g\chi)^{\tau_{\text{DMFT}}} = e^{-1}$, which yields:

$$\tau_{\text{DMFT}} = -\frac{1}{\ln(g\chi)}. \tag{21}$$

This is the direct mean-field analogue of the expression derived from the Jacobian's spectral radius. As the system approaches the critical point where $g\chi \to 1$, the timescale diverges as $\tau_{\text{DMFT}} \approx \frac{1}{1-g\chi}$.

2.8 Theoretical Basis for State-Space Separation

Here we establish theoretically why distinct input signals produce separable trajectories in the network's state space. This principle is general and underpins the network's ability to discriminate between different stimuli, whether they are static or time-varying. We show that a difference in inputs acts as a driving force that pushes the network states apart, a separation that is then propagated and shaped by the recurrent dynamics.

Consider two different input signals, $\boldsymbol{u}_A(t)$ and $\boldsymbol{u}_B(t)$, which generate two corresponding state trajectories, $\boldsymbol{x}_A(t)$ and $\boldsymbol{x}_B(t)$. To perform a linear analysis, we approximate the network dynamics by assuming the system operates in the linear regime of the activation function, i.e., $\tanh(\boldsymbol{z}) \approx \boldsymbol{z}$. Under this assumption, the state evolution for each trajectory becomes:

$$\boldsymbol{x}_A(t+1) \approx \boldsymbol{W}\boldsymbol{x}_A(t) + \boldsymbol{W}_{\text{in}}\boldsymbol{u}_A(t) \tag{22}$$

$$\boldsymbol{x}_B(t+1) \approx \boldsymbol{W}\boldsymbol{x}_B(t) + \boldsymbol{W}_{\text{in}}\boldsymbol{u}_B(t) \tag{23}$$

To analyse their separation, we examine the evolution of their difference vector, $\Delta\boldsymbol{x}(t) = \boldsymbol{x}_A(t) - \boldsymbol{x}_B(t)$. Subtracting the two equations yields a simple linear time-invariant system for the difference vector:

$$\Delta\boldsymbol{x}(t+1) \approx \boldsymbol{W}\Delta\boldsymbol{x}(t) + \boldsymbol{W}_{\text{in}}\Delta\boldsymbol{u}(t) \tag{24}$$

where $\Delta\boldsymbol{u}(t) = \boldsymbol{u}_A(t) - \boldsymbol{u}_B(t)$. This simplified linear system reveals two key mechanisms:

1. **Input-Driven Separation:** The term $\boldsymbol{W}_{\text{in}}\Delta\boldsymbol{u}(t)$ acts as a direct external driving

force. So long as the input signals are different ($\Delta \boldsymbol{u}(t) \neq \mathbf{0}$), this term continuously pushes the trajectories apart.

2. **Recurrent Propagation:** The term $\boldsymbol{W}\Delta \boldsymbol{x}(t)$ describes how any existing separation is propagated by the fixed recurrent dynamics. In the stable regime ($\rho(\boldsymbol{W}) < 1$), these dynamics will transform the separation vector without causing it to explode.

Assuming the network starts from the same initial state ($\Delta \boldsymbol{x}(0) = \mathbf{0}$), any subsequent difference between the inputs will create a non-zero separation vector. By induction, the separation at any time T is a cumulative result of the entire history of input differences, each propagated through the network's fixed linear dynamics. Therefore, even under this model with linearised dynamics, we can conclude that distinct input histories will generate unique and separable trajectories. This fundamental property is what we leverage in the classification/decision-making tasks (Section 5).

2.9 Signal Separation

The preceding section has outlined that different signals are separated in the state-space, and the *signal separation capacity* (or discriminability) is used to measure how easy or hard it is to discriminate or distinguish between two signals (or signal distributions). This is crucial for tasks such as classification, decision making, and memory, where separability of neural representations determines performance. In the context of this present work, we want to analyse the the ability of the network to produce distinct internal states in response to different input conditions, and compare how the signal discriminability of various input regimes compares with the discriminability of the resulting internal neural dynamics. Assuming two multivariate Gaussian signal distributions with means $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^N$ and covariances $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2 \in \mathbb{R}^{N \times N}$, the discriminability d' between these signals is defined as:

$$d'^2 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2), \tag{25}$$

where $\boldsymbol{\Sigma}$ is an estimate of the shared noise covariance (such as pooled covariance, or empirically estimated covariance). This metric can be used both in input space (to assess input

discriminability) and in the state space of the network (to assess representational discriminability). Comparisons between the two indicate how well the network geometry preserves or amplifies task-relevant distinctions.

Improved Signal Separation in Network States

To build an intuition for how the network transforms signal discriminability, we can develop a scaling argument by approximating the amplification of the signal (the separation between fixed points) and the noise (the state covariance). Given two input signals with means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ (and equal covariance $\boldsymbol{\Sigma}_{\text{in}}$), the difference between input means is $\Delta\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$. The difference between the corresponding network fixed points is:

$$\begin{aligned}\Delta\boldsymbol{x}^* &:= \boldsymbol{x}^*(\boldsymbol{\mu}_1) - \boldsymbol{x}^*(\boldsymbol{\mu}_2) \\ &\approx (\boldsymbol{I} - \boldsymbol{W})^{-1}\boldsymbol{W}_{\text{in}}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) = (\boldsymbol{I} - \boldsymbol{W})^{-1}\boldsymbol{W}_{\text{in}}\Delta\boldsymbol{\mu}.\end{aligned}$$

To understand how the signal is amplified, we consider the norm of this difference. The amplification is governed by the resolvent matrix $(\boldsymbol{I} - \boldsymbol{W})^{-1}$. While for a general non-normal matrix $\boldsymbol{W}$ the norm $\|(\boldsymbol{I} - \boldsymbol{W})^{-1}\|$ can be much larger than $1/(1-\rho)$, we can use the spectral radius to establish a heuristic scaling that is particularly relevant as $\rho \to 1$. We approximate that the signal amplification scales with this dominant factor:

$$\|\Delta\boldsymbol{x}^*\| \propto \frac{\iota}{1-\rho}\|\Delta\boldsymbol{\mu}\|. \tag{26}$$

Next, to estimate how variability (i.e., noise) in the inputs is amplified, we again consider the discrete Lyapunov equation (Equation 5), $\boldsymbol{\Sigma} = \boldsymbol{J}\boldsymbol{\Sigma}\boldsymbol{J}^\top + \boldsymbol{W}_{\text{in}}\boldsymbol{\Sigma}_{\text{in}}\boldsymbol{W}_{\text{in}}^\top$. Unrolling this equation recursively and approximating $\boldsymbol{J}$ with $\boldsymbol{W}$ yields:

$$\boldsymbol{\Sigma} = \sum_{k=0}^{\infty} \boldsymbol{W}^k \boldsymbol{Q} (\boldsymbol{W}^\top)^k, \tag{27}$$

where $\boldsymbol{Q} = \boldsymbol{W}_{\text{in}}\boldsymbol{\Sigma}_{\text{in}}\boldsymbol{W}_{\text{in}}^\top$. By approximating the effect of the sum with a geometric series based on the spectral radius $\rho = \rho(\boldsymbol{W})$, we can approximate the scaling of the total variance.

Taking norms, the state covariance norm can be approximated as:

$$\|\boldsymbol{\Sigma}\| \approx \frac{1}{1-\rho^2}\|\boldsymbol{Q}\| \leq \frac{\iota^2}{1-\rho^2}\|\boldsymbol{\Sigma}_{in}\|. \tag{28}$$

This suggests that an estimate for the noise variance in the network state scales approximately with $\frac{\iota^2}{1-\rho^2}$. Combining these scaling approximations allows us to estimate the network discriminability, d'_{net}. If we define an effective input discriminability based on norms as ${d'_{in}}^2 := \frac{\|\Delta\boldsymbol{\mu}\|^2}{\|\boldsymbol{\Sigma}_{in}\|}$, we can write:

$$\begin{aligned}
{d'_{net}}^2 &= (\Delta\boldsymbol{x}^*)^\top \boldsymbol{\Sigma}^{-1} \Delta\boldsymbol{x}^* \\
&\approx \frac{\|\Delta\boldsymbol{x}^*\|^2}{\|\boldsymbol{\Sigma}\|} \\
&\propto \frac{\left(\frac{\iota}{1-\rho}\|\Delta\boldsymbol{\mu}\|\right)^2}{\frac{\iota^2}{1-\rho^2}\|\boldsymbol{\Sigma}_{in}\|} \\
&= \frac{1-\rho^2}{(1-\rho)^2} \cdot \frac{\|\Delta\boldsymbol{\mu}\|^2}{\|\boldsymbol{\Sigma}_{in}\|} \\
&= \frac{1+\rho}{1-\rho}{d'_{in}}^2.
\end{aligned}$$

Thus, this heuristic argument suggests that the network can amplify discriminability by a factor that scales with $\frac{1+\rho}{1-\rho}$. It is important to treat this as a scaling relationship rather than a strict upper bound. The analysis relies on norm-based approximations and does not capture the crucial potential geometric alignment between the signal vector $\Delta\boldsymbol{x}^*$ and the eigenspectrum of the noise covariance $\boldsymbol{\Sigma}$. Due to the non-normal

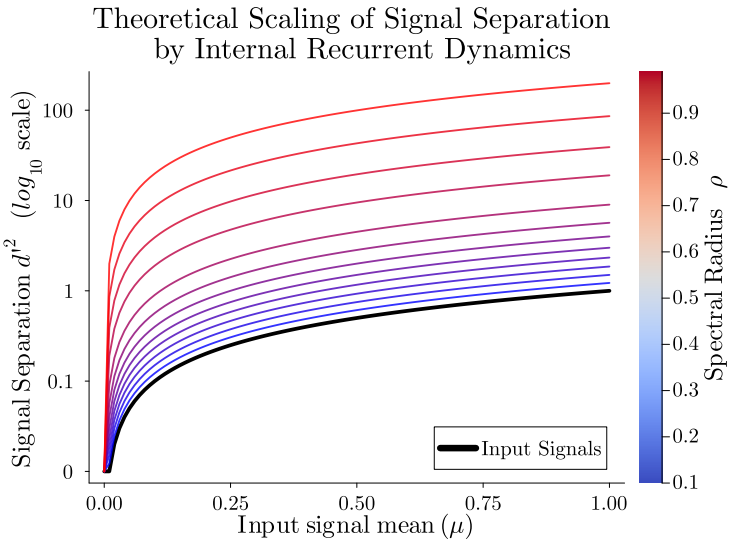

Figure 1: Theoretical scaling of the signal separation for internal network states relative to inputs, as a function of input mean μ and spectral radius ρ.

nature of $\boldsymbol{W}$, the network can amplify the signal along directions of relatively low noise, achieving a d' value that exceeds this simple estimate. Nonetheless, this result provides valuable intuition for the substantial representational advantage conferred by recurrent dynamics operating near the edge of stability.

2.10 Dimensionality Reduction via PCA

To understand the geometry of the network dynamics, we apply Principal Component Analysis (PCA) which reduces the dimensionality and helps to visualise the high-dimensional activity. Given the zero-mean data matrix $\boldsymbol{X} \in \mathbb{R}^{N \times T}$ composed of network states over T time steps, we compute the covariance matrix:

$$\boldsymbol{C} = \frac{1}{T} \boldsymbol{X} \boldsymbol{X}^\top, \tag{29}$$

and extract the leading eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots$ to define the principal components. The projection of the data onto the k-dimensional PCA space is then:

$$\boldsymbol{X}_{\text{PCA}} = \boldsymbol{V}_k^\top (\boldsymbol{X} - \bar{\boldsymbol{X}}), \tag{30}$$

where $\boldsymbol{V}_k = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k]$ and $\bar{\boldsymbol{X}}$ is the activity mean vector. The projected fixed points are therefore:

$$\boldsymbol{x}^*_{\text{PCA}} = \boldsymbol{V}_k^\top (\boldsymbol{x}^* - \bar{\boldsymbol{X}}), \tag{31}$$

and the projected covariance matrices become:

$$\boldsymbol{\Sigma}_{\text{PCA}} = \boldsymbol{V}_k^\top \boldsymbol{\Sigma} \boldsymbol{V}_k. \tag{32}$$

2.11 Biophysically-Realistic Rule for Learning

While PCA provides a powerful tool for analysing high-dimensional network dynamics, its standard implementation involves global computations (e.g., eigenvalue decomposition) that are not biologically plausible. A more realistic mechanism is given by a generalised Hebbian algorithm, also known as *Sanger's rule* – an online, local learning rule that approximates

PCA through synaptic updates based only on pre- and post-synaptic activity.

Let $\boldsymbol{x}(t) \in \mathbb{R}^N$ denote the zero-mean network state at time t, and let $\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{k \times N}$ be a matrix whose rows $\boldsymbol{w}_i^\top$ represent synaptic weights projecting onto k readout neurons. The activity of these neurons is $\boldsymbol{y}(t) = \boldsymbol{W}_{\text{out}}\boldsymbol{x}(t)$. Sanger's rule updates the weights as:

$$\Delta \boldsymbol{w}_i(t) = \eta y_i(t) \left[\boldsymbol{x}(t) - \sum_{j=1}^{i} y_j(t) \boldsymbol{w}_j(t) \right], \tag{33}$$

where $\eta > 0$ is a small learning rate. This rule ensures that weight updates are local while maintaining orthogonality between weight vectors (via the subtractive projection term). Thus, Sanger's rule provides a biologically plausible mechanism for learning a low-dimensional representation of dynamics that aligns with the geometry captured by PCA; this serves as a biophysically-based approach for constructing interpretable readout layers.

3 Comparisons between Theory and Simulation

To validate our theoretical framework, we compare its predictions against direct numerical simulations of the network dynamics defined in Equation 1. These simulations represent the "ground truth" for a network of finite size ($N = 100, \rho = 0.99$) and serve as a benchmark for our theoretical approximations. We compare statistics from three sources:

1. **Simulation:** The empirical mean and mean-squared activity measured directly from the time series of a simulated network.

2. **Linearised Theory:** The theoretical prediction for the same network, derived from its specific fixed point $\boldsymbol{x}^*$ and covariance matrix $\boldsymbol{\Sigma}$ (Sections 2.2 & 2.5).

3. **Dynamical Mean-Field Theory:** The theoretical prediction for the *ensemble average* of networks in the $N \to \infty$ limit, using the self-consistent equations (Equations 13 & 14).

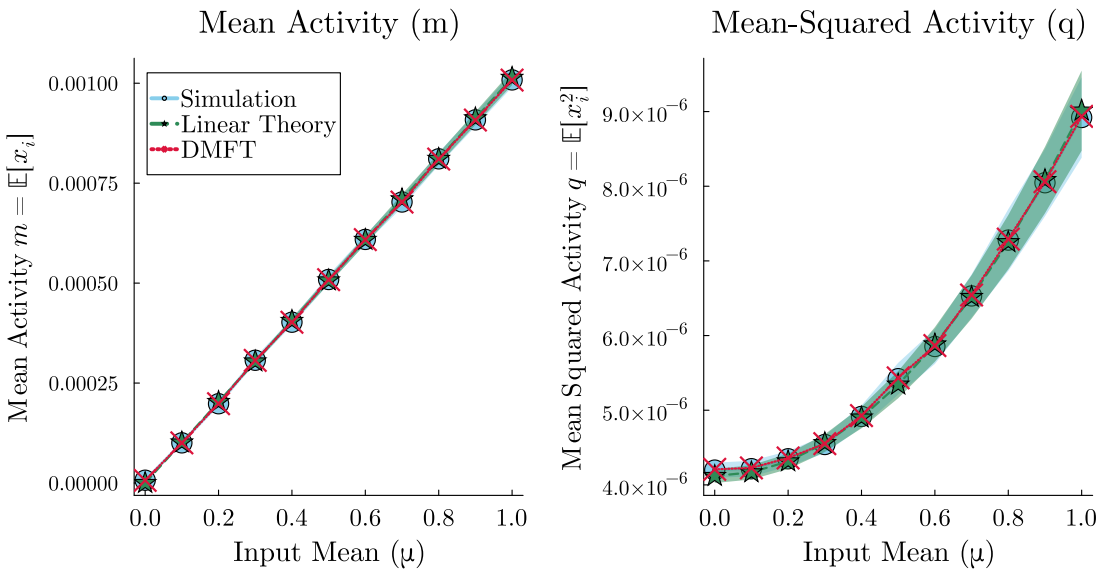

Figure 2: **Comparison of Network Activity Statistics.** Mean activity (m, left) and mean-squared activity (q, right) as a function of the mean input μ. Blue circles show results from direct simulation (ground truth), averaged over 100 trials. Green squares show the average prediction from the linearised theory for each network instance. Red crosses show the prediction from the simple Dynamical Mean-Field Theory.

3.1 Analysis of Mean and Mean-Squared Activity

Figure 2 demonstrates close agreement between the direct simulation, linearised theory, and the ensemble-averaged DMFT for mean values of input between 0 and 1. The left panel shows the population-averaged mean activity, $m = \mathbb{E}[x_i]$, and the right panel displays the mean-squared activity, $q = \mathbb{E}[x_i^2]$, both as a function of the mean input μ (with a standard deviation of 1.0). All three methods produce closely aligning curves, confirming that they accurately capture the network's first and second-order statistics in response to the external drive. The close match between linearised theory and the full simulation validates our use of a linear approximation for the fixed point in this parameter regime.

In summary, this comparison provides a strong validation of our entire theoretical approach. It shows that our linearised theory for a single instance accurately reflects the behaviour of the full non-linear simulation, and that both are well-described by the predictions of a simple, classic dynamical mean-field theory, demonstrating that the behaviour of one typical network instance indeed represents the ensemble average.

3.2 Fixed Point Geometry

To understand the geometry of network dynamics for different inputs, we consider the steady-state (fixed point) activity of the recurrent network for constant input $\mu \in \mathbb{R}$. As presented earlier, under the linear approximation around the fixed point, the steady-state equation becomes: $\boldsymbol{x}^*(\mu) \approx (\boldsymbol{I} - \boldsymbol{W})^{-1}\boldsymbol{W}_{\text{in}}\mu$. This expression is a *linear* function of the input mean μ. Thus, the family of fixed points lies on a one-

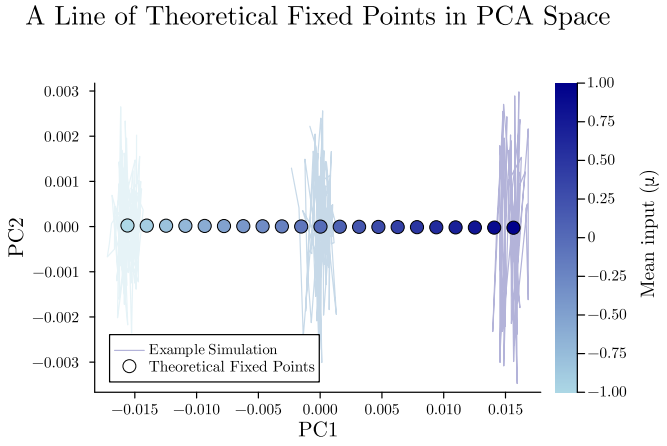

Figure 3: Theoretical steady state/fixed points for input mean values μ from -1 to 1.

dimensional subspace of $\mathbb{R}^N$, parametrised by $\boldsymbol{x}^*(\mu) = \boldsymbol{A}\mu$, where $\boldsymbol{A} = (\boldsymbol{I} - \boldsymbol{W})^{-1}\boldsymbol{W}_{\text{in}}$. Consequently, the fixed points for all input values $\mu \in \mathbb{R}$ lie on a line in neuron space. Since the PCA space is a linear projection, the low-dimensional representations of the steady states of the neural dynamics during this task also lie on a line in principal component space (e.g. Figure 3).

3.3 Analysis of Intrinsic Timescales

We investigate the network's intrinsic timescale, a crucial property for memory and temporal processing. We compare simulations against the theoretical predictions from both the single-instance linearised theory and DMFT. The empirical timescale, τ_{sim}, is extracted by fitting an exponential decay to the population-averaged autocorrelation function of the simulated network activity. The theoretical timescales, τ_{lin} and τ_{DMFT}, are calculated from the Jacobian and the DMFT susceptibility, respectively, as described in Section 2.7.

Figure 4 shows these comparisons as a function of the spectral radius ρ. It demonstrates that as the network is tuned closer to the edge of chaos ($\rho \to 1$), its intrinsic timescale grows exponentially, a phenomenon known as critical slowing down. This confirms that tuning the network's stability provides a direct mechanism for controlling its memory capacity. Crucially, direct simulation, linearised theory, and DMFT are in excellent agreement across the entire range of ρ values.

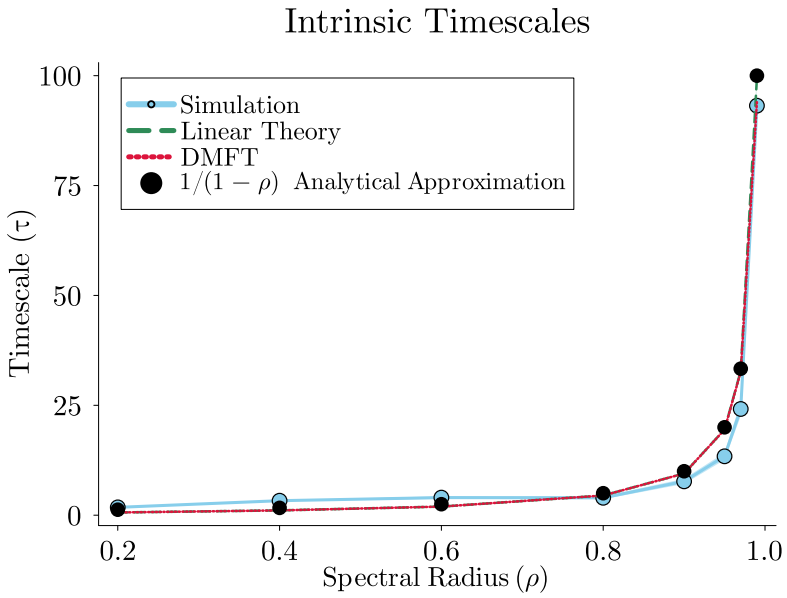

Figure 4: **Comparison of Network Timescales.** The intrinsic timescale τ as a function of the spectral radius ρ. The timescale grows exponentially as the network approaches the critical point at $\rho = 1$. Simulation results (blue) are tightly matched by predictions from single-instance linear theory (green), DMFT (red), and the analytical approximation given by $\frac{1}{1-\rho}$ (black circles).

This result unifies our analysis, showing that the same mathematical properties of the recurrent connectivity that determine the static geometry of network representations (and thus signal separation) also govern the network's dynamic properties, such as its intrinsic timescale for temporal integration.

3.4 Signal Separation

We evaluate how the internal dynamics of random networks improve signal discriminability relative to the inputs. Specifically, we generate Gaussian input signals S_1 with varying means $\mu \in [0, 1]$ and fixed variance $\sigma^2_{\text{in}} = 1$, and compare each to a baseline input S_2 with mean $\mu_2 = 0$, and also fixed variance $\sigma^2_{\text{in}} = 1$. We compute the full d' discriminability between network responses to S_1 and S_2, defined via the Mahalanobis distance (Equation 25) between their respective means and covariances. This accounts for the full geometry of variability in the state space and allows us to better quantify how much more separable the network states become, compared to the input signals alone. This is compared with the theoretical scaling factor derived in Section 2.9. Figure 5 exhibits how much the signal separation is amplified in the internal network dynamics over the inputs, and also demonstrates close alignment with the theoretical scaling factor. Section 5.1 will later demonstrate how this increased separation facilitates decision-making.

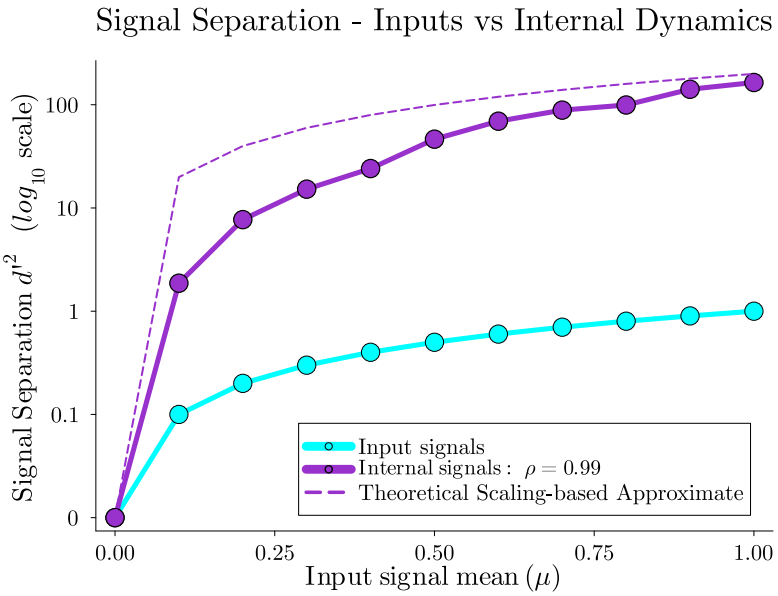

Figure 5: **Signal Separation Capacity.** A comparison between signal discriminability of input signals and internal dynamics (using Mahalanobis metric) across varying input mean μ_1 relative to $\mu_2 = 0$, for $\rho = 0.99$. The comparison demonstrates an amplification of signal separation within the network, and also indicates close agreement with the approximate scaling factor developed in Section 2.9.

4 Example Task: n-back Sequence Recall

4.1 Task Description

In this section, we consider a task that tests the ability of a network to maintain and recall a history of its past inputs. A sequence of random values, sampled *iid* from a Gaussian distribution with mean μ and variance σ_{in}^2, is fed sequentially into the network. The task is to reproduce the input from n time steps earlier. As n increases, the network must retain information over longer timescales, making the task more challenging. We investigate how the internal dynamics of the network encode this temporal history, and how the current neural state allows the network to "look into the past."

4.2 How Current Neural Activity Encodes History

To understand how the network encodes its past inputs in the current activity, we consider the linearised dynamics of the recurrent neural network with input $u(t)$ (assuming without loss of generality $\mu = 0, \sigma_{in}^2 = 1$, spectral radius $\rho < 1$, and $\boldsymbol{W}_{in}$ is scaled appropriately). The dynamics are:

$$\boldsymbol{x}(t+1) = \tanh\left(\boldsymbol{W}\boldsymbol{x}(t) + \boldsymbol{W}_{\text{in}}u(t)\right) \approx \boldsymbol{W}\boldsymbol{x}(t) + \boldsymbol{W}_{\text{in}}u(t).$$

Unrolling this recurrence relation gives:

$$\boldsymbol{x}(t) = \sum_{\tau=0}^{t-1} \boldsymbol{W}^{\tau}\boldsymbol{W}_{\text{in}}u(t-1-\tau).$$

This expression shows that the current state $\boldsymbol{x}(t)$ is a linear combination of past inputs $u(t-1), u(t-2), \ldots$, each mapped through increasingly high powers of $\boldsymbol{W}$. Assuming $\boldsymbol{W}$ is diagonalisable with eigenvalues λ_i, right eigenvectors $\boldsymbol{v}_i$, and left eigenvectors $\boldsymbol{z}_i$ (i.e. $\boldsymbol{W} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{Z}^{-1}$), we can write:

$$\boldsymbol{W}^{\tau}\boldsymbol{W}_{\text{in}} = \sum_i \lambda_i^{\tau}\langle \boldsymbol{z}_i, \boldsymbol{W}_{\text{in}}\rangle \boldsymbol{v}_i,$$

which shows that each past input $u(t-\tau)$ contributes to the network state in the direction of $\boldsymbol{v}_i$, scaled by λ_i^τ and the alignment of the input direction with the corresponding left eigenvector. Since $|\lambda_i| < 1$, older inputs contribute less to the current activity, demonstrating an intrinsic memory decay ("fading memory") due to the spectral radius of $\boldsymbol{W}$. This temporal filtering induces a structured covariance in the network states $\boldsymbol{X}$, where earlier inputs are attenuated and recent inputs dominate. When applying PCA to $\boldsymbol{X}$, we therefore expect the principal components to reflect these temporal contributions.

We analyse the low-dimensional geometry of the neural activity using PCA. As the above indicates, we indeed find that temporal contributions are reflected in the principal components, and that remarkably, the kth principal component of the network state is most strongly correlated with the input k time steps in the past. This indicates a structured encoding of temporal information: the current neural activity $\boldsymbol{x}(t)$ projects onto different principal axes that selectively encode past inputs at varying lags. In particular, the correlation matrix between lagged inputs and PCs reveals a diagonally dominant structure, reflecting a temporal stratification of memory across the PC axes (Figure 6). PCA recovers directions in state space that encode inputs at specific lags.

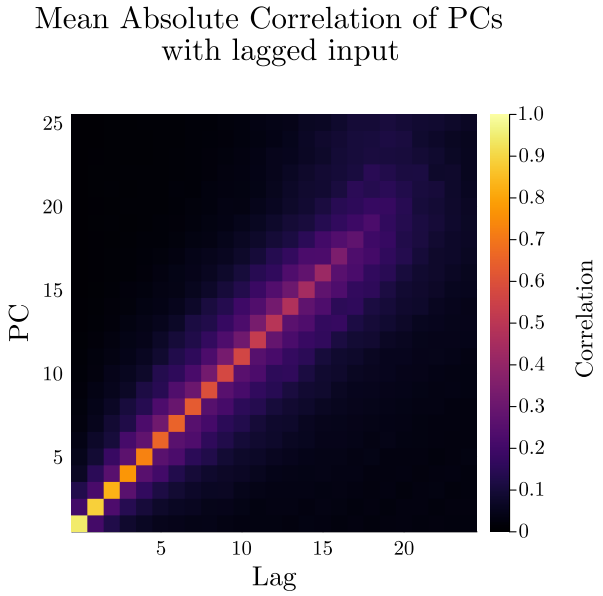

Figure 6: Mean absolute value of correlations between lagged input and principal components of neural activity over multiple trials.

4.3 Constructing $\boldsymbol{W}_{\text{out}}$ from the Geometry of Dynamics

4.3.1 Using the Eigenstructure of $\boldsymbol{W}$

From the insights outlined above, we can directly construct a theoretical readout layer $\boldsymbol{W}_{\text{out}}$ by leveraging the eigenstructure of the recurrent connectivity matrix. As shown earlier, the

contribution of the input $u(t-n)$ to the current state is $\boldsymbol{W}^n \boldsymbol{W}_{\text{in}} = \sum_i \lambda_i^n \langle \boldsymbol{z}_i, \boldsymbol{W}_{\text{in}} \rangle \boldsymbol{v}_i$, where λ_i, $\boldsymbol{v}_i$, and $\boldsymbol{z}_i$ are the eigenvalues, right eigenvectors, and left eigenvectors of $\boldsymbol{W}$, respectively. This motivates the following theoretical construction of the readout vector, for an n-back sequence recall:

$$\boldsymbol{W}_{\text{out}} = \sum_i \lambda_i^n \langle \boldsymbol{z}_i, \boldsymbol{W}_{\text{in}} \rangle \boldsymbol{v}_i.$$

This readout direction maximally extracts the component of $\boldsymbol{x}(t)$ that contains information about the input n steps ago, and does so entirely from known system parameters, without simulation or training. The readout at time t is then given by $y(t) = \boldsymbol{W}_{\text{out}} \boldsymbol{x}(t)$, which approximates the input from $t-n$. See Figure 7 for an example of the network output based on a readout layer $\boldsymbol{W}_{\text{out}}$ constructed using this approach.

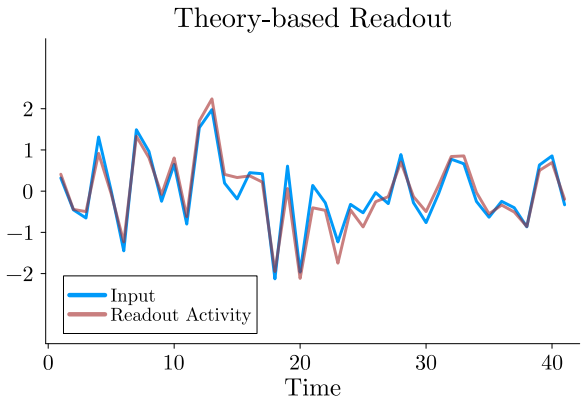

Figure 7: Example n-back input and network output using eigenstructure-derived $\boldsymbol{W}_{\text{out}}$.

4.3.2 Using Simulated Trajectories and PCA

Leveraging the observed correspondence between lag and principal component, we can construct a theoretically-informed readout vector without "training" (in the normal online or offline, supervised way). For a target lag n, we identify the principal component $\boldsymbol{p}_k$ that most strongly correlates with the input n time steps ago. We then define the readout vector $\boldsymbol{W}_{\text{out}}$ as:

$$\boldsymbol{W}_{\text{out}} = \text{sign}(\rho_{k,n}) \cdot \boldsymbol{p}_k,$$

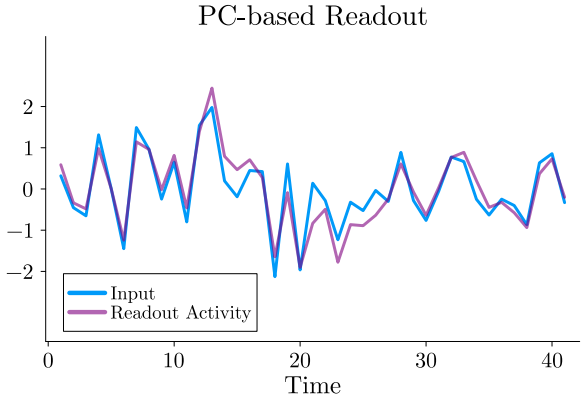

Figure 8: Example n-back input and network output using a PC derived $\boldsymbol{W}_{\text{out}}$.

where $\rho_{k,n}$ is the empirical correlation between PC_k and the lag-n input. This sign adjustment ensures that the readout aligns positively with the target signal. This approach constructs a decoding scheme grounded in the intrinsic geometry of the network dynamics, rather than relying on supervised training, and achieves reliable memory-based output from the network. See Figure 8 for an example of the network output based on a readout layer $\boldsymbol{W}_{\text{out}}$ constructed using the method outlined above.

4.3.3 Using a Biophysically-Realistic Learning Rule

As outlined in Section 2.11, we can implement a local learning rule to approximately learn the principal components of the network's activity. We then apply the same approach as above, where we select the principal component $\boldsymbol{p}_k$ that most strongly correlates with the lag-n input, and use this to construct $\boldsymbol{W}_{\text{out}}$. See Figure 9 for an example of the network output based on a readout layer $\boldsymbol{W}_{\text{out}}$ constructed using Sanger's rule.

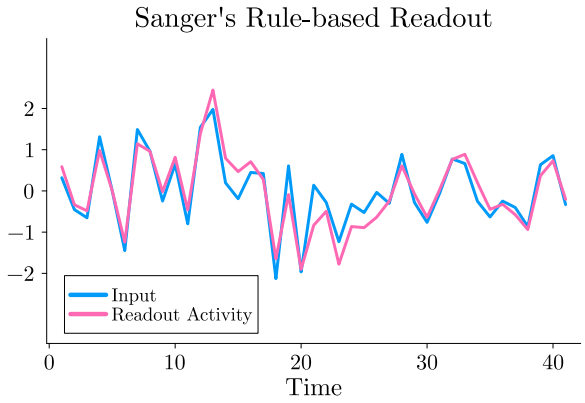

Figure 9: Example n-back input and network output using Sanger's rule.

5 Example Task: Classification / Decision-Making

5.1 (a) Simple Decision-Making

5.1.1 Task Description

In this section, we describe a simple and standard decision making task, demonstrating how the theoretical fixed points and variance derived in Sections 2.2 and 2.5 align with simulated dynamics. We also demonstrate how signal discriminability between two example inputs increases in the internal network states compared with the inputs themselves. Furthermore, we explore the role of the PCA state space and how the low dimensional geometry of the neural dynamics can be used to effectively complete this task, constructing a readout layer $\boldsymbol{W}_{\text{out}}$ using the first principal component loadings.

In this task, the network receives a scalar noisy input stimulus with mean μ and variance σ_{in}^2. The network must integrate the signal over time and determine whether the *mean average* of the noisy input is greater than or less than 0, giving a positive output if $\mu > 0$ and a negative output if $\mu < 0$. The difficulty of the task is governed by the two parameters μ and σ_{in}^2, i.e. how close the signal mean is to 0, and how variable the signal is.

5.1.2 Theoretical Fixed Points and Simulation

We demonstrate here, as an example, the agreement between simulation and theory when driving the network (number of neurons $N = 100$) activity by three different input means, μ_1, μ_2, μ_3, each with $\sigma_{in}^2 = 1.0$. Figure 10 (A) displays the example stochastic signals with $\mu_1 = 1.0, \mu_2 = 0.0, \mu_3 = -1.0$. In this example, the average sign (+1 or -1) of Input 1 and Input 3 are relatively easy to discriminate, unlike Input 2 which has a mean of 0, and is therefore impossible to discriminate as being positive or negative on average.

We observe, as the theory predicts (Section 3.2) that the fixed points for different input are collinear, and also that signals in the state space (Figure 10, Panel B) are more discriminated and separated than the inputs (Figure 10, Panel A). This was theoretically demonstrated in Section 3.4, and can easily be observed and confirmed by this example. This increased discriminability between signals is fundamental to how the network is able to "pick apart"

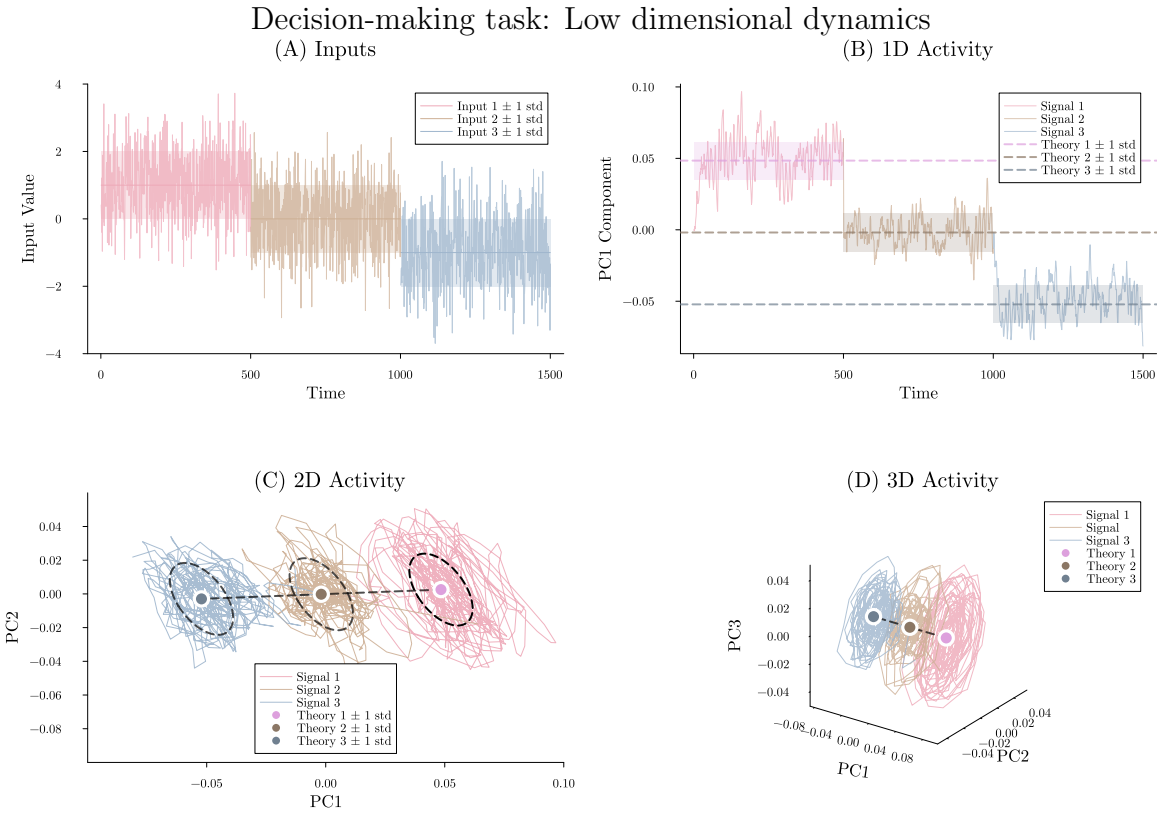

Figure 10: Example inputs (A) and the corresponding internal network dynamics in low dimensional state space (B-D).

signals that are harder to discern in the input space. We further demonstrate this in the following section.

5.1.3 Constructing W_{out} from the Geometry of Dynamics

To understand how the network performs decision-making, we examine the relationship between the low-dimensional geometry of the neural trajectories and the linear readout used for classification. Specifically, we show how a readout layer can be constructed using both the *theoretical* fixed points and covariances, and using the first principal component of the *simulated* network activity (by PCA and Sanger's rule)

Using Theoretical Fixed Points and Covariance

To analytically derive the decision readout, we consider the fixed points and covariances of the network dynamics under different input conditions. We have previously shown that

varying input μ results in a line of fixed points in neural space (Section 5.3.2). The difference vector between two fixed points $\Delta \boldsymbol{x}^*$, captures the principal direction in state space that distinguishes the two fixed points of the decision states. When the covariance of the neural activity around each fixed point is approximately isotropic, the difference vector $\boldsymbol{W}_{\text{out}} = \frac{\Delta \boldsymbol{x}^*}{\|\Delta \boldsymbol{x}^*\|}$ would provide a natural readout direction. However, as already discussed, the covariance matrices $\boldsymbol{\Sigma}_+$ and $\boldsymbol{\Sigma}_-$ around these fixed points are anisotropic, reflecting structured variability in network activity. In this case, simply using the difference vector neglects the geometry of noise and variability that can degrade discriminability. To incorporate this anisotropy, we formulate the readout as a Fisher discriminant direction,

$$\boldsymbol{W}_{\text{out}} = \frac{(\boldsymbol{\Sigma}_+ + \boldsymbol{\Sigma}_-)^{-1} \Delta \boldsymbol{x}^*}{\|(\boldsymbol{\Sigma}_+ + \boldsymbol{\Sigma}_-)^{-1} \Delta \boldsymbol{x}^*\|},$$

which weights the difference by the inverse of the total covariance, optimally separating the two states by maximising signal-to-noise ratio.

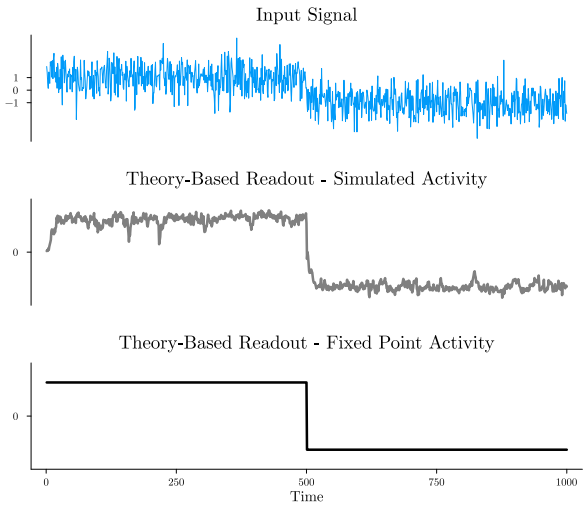

Figure 11: Example input and theoretically-based readout.

This improved readout accounts for the full noise geometry around the fixed points, thereby improving classification performance over the naive difference vector. The scalar output is then computed as $y(t) = \boldsymbol{W}_{\text{out}} \boldsymbol{x}(t)$, projecting the network state onto this noise-aware task-relevant axis. This approach reveals that the readout layer can be constructed purely from the geometry of network dynamics and their variability, without requiring PCA on simulation data.

Using Simulated Trajectories

We can also construct a geometry-based readout layer by using a low dimensional projection of the simulated trajectories. The PCA projection matrix $\boldsymbol{P}$ provides a set of orthogonal directions in state space, ordered by the variance explained. The first principal component $\boldsymbol{p}_1$ (i.e., the first column of $\boldsymbol{P}$) captures the direction of maximum variance. We can use

this component to define a readout weight vector $\boldsymbol{W}_{\text{out}} = \boldsymbol{p}_1$ which is applied linearly to the network state. This allows us to interpret the readout layer as a projection onto a meaningful low-dimensional axis that aligns with task-relevant structure. We evaluate the contributions of individual neurons to the first principal component via the loadings, identifying which neurons are most influential for the readout. Figure 12 demonstrates how an example $\boldsymbol{W}_{\text{out}}$ constructed in this way performs in deciding between two example input stimuli, clearly distinguishing between two noisy signals.

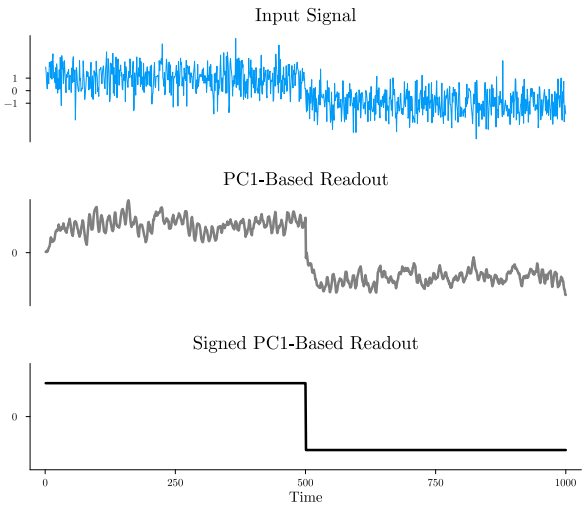

Figure 12: Example input and simulation (via PC1) based readout.

Using a Biophysically-Realistic Learning Rule

As with the memory task (Section 4.3), we can implement a local learning rule to approximate the principal components of the activity. As above indicates, the first principal component is sufficient to effectively distinguish between input signals. Using Sanger's rule (Section 2.11) to calculate the first PC, we use the loadings of this as the readout layer. Figure 13 shows an example of network output based on $\boldsymbol{W}_{\text{out}}$ constructed via Sanger's rule.

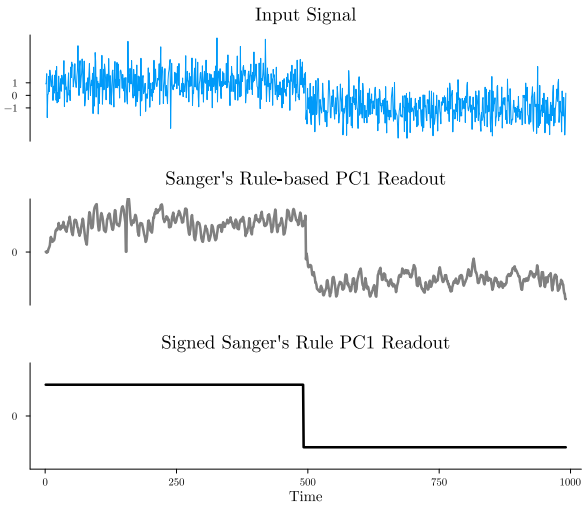

Figure 13: Example input and Sanger's Rule-based readout.

5.2 (b) Delayed Decision-Making

5.2.1 Task Description

Here we briefly turn our attention to a task that is very similar to the decision-making task studied in Section 5.1. The "delayed" version we now explore introduces a small difference in task implementation – namely, fixation and response periods – which results in an interesting added challenge for understanding the network dynamics and how this gives rise to successful task completion.

The setup of this task is very similar to that described in Section 5.1. The main difference is that the task is divided into a "fixation phase" (where a noisy stimulus with mean μ and variance σ_{in}^2 drives the system, and the desired output is a fixation on/around 0), and a "response phase" (when the stimulus is removed, and the network must then make a decision as to whether or not the mean of the past noisy input was greater than or less than 0). Once again, the difficulty of this task is controlled by the parameters of μ and σ_{in}^2; however, as we will outline, added difficulty arises from the demands placed by the introduction of these two distinct phases.

5.2.2 Theoretical Fixed Points and Relaxation Dynamics

The theoretical fixed points and variance in this task are exactly the same as those from the previous task, including the fact that the fixed points (for varying $\mu \in \mathbb{R}$) lie on a line in neural space. An extra consideration here is the fact that, from every fixed point the network state occupies (arising from the mean input μ during the fixation phase), the network state is then released and decays towards $\mathbf{0}$ (during the response phase). Owing to the construction of $\boldsymbol{W}$ with $\rho < 1$, we are guaranteed that the dynamics will decay to $\mathbf{0}$. This is true even outside the linear approximation regime, as a consequence of the fact that the origin is a globally attracting fixed point, because the non-linear function tanh is contractive, with Lipschitz constant < 1. Our interest here, therefore, is in the relaxation dynamics during the response phase, as this is where the output decision is encoded.

If we denote the fixed point for a given input stimulus (mean μ) by $\boldsymbol{x}^*_\mu$, then we can – in the linearised system – theoretically calculate the relaxation dynamics from this point by $\boldsymbol{x}(t) \approx \boldsymbol{W}^t \boldsymbol{x}^*_\mu$, where t is the time from onset of the response period. The trajectory during the response phase is thus fully determined by the fixed point state $\boldsymbol{x}^*_\mu$, and by the eigenstructure of $\boldsymbol{W}$, where the dominant components of the dynamics are aligned with the leading eigenvectors of $\boldsymbol{W}$ (see the following section). Figure 14 demonstrates example simulation and theory dynamics for the delayed decision-making task. We note (e.g. in the two-dimensional PCA state space in Panels (B) and (D)) that the principal line of relaxation dynamics differs from the line of fixed points during fixation. This is a crucial aspect of the dynamics geometry for constructing $\boldsymbol{W}_{\text{out}}$.

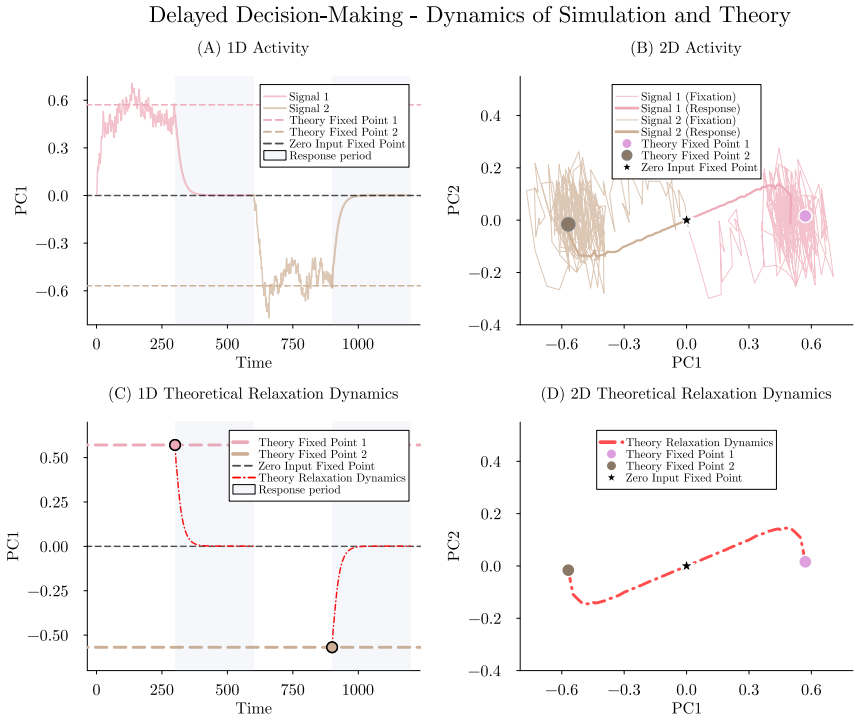

Figure 14: Example low dimensional dynamics of the delayed-decision-making task.

5.2.3 Constructing $\boldsymbol{W}_{\text{out}}$ from the Geometry of Dynamics

To construct a readout layer $\boldsymbol{W}_{\text{out}}$ that successfully outputs 0 during the fixation period and a "decision" (+1 or −1) during the response period, we can leverage the observation that the internal dynamics during relaxation differ geometrically from that during fixation. We therefore want to identify a direction (a "coding direction") in the N-dimensional state space that optimally distinguishes fixation-period activity from response-period activity. The idea is to find a vector (as in Section 5.1.3) which captures the direction in state space that

most *distinguishes between the two decisions during response dynamics*, but is *orthogonal* (or as close to orthogonal as possible) *to fixation dynamics*. Formally, we seek a projection vector, $\mathbf{w} \in \mathbb{R}^N$, that maximises the variance of the neural states during the response period relative to the variance during the fixation period. This is difficult to do in low dimensions, but becomes easier in higher dimensions.

As alluded to previously, since the relaxation dynamics are determined by $\boldsymbol{W}$, the dominant components must align with the leading eigenvectors of $\boldsymbol{W}$. We also showed earlier (Section 5.1.2) that fixed points corresponding to different inputs lie on a line in neural space, defined by $\boldsymbol{A} = (\boldsymbol{I} - \boldsymbol{W})^{-1}\boldsymbol{W}_{\text{in}}$. This means that the dominant directions of dynamics during fixation and response are defined by $\boldsymbol{A}$ and $\boldsymbol{W}$, respectively (Figure 15). The issue is, however, that the line of fixed points determined by $\boldsymbol{A}$ may overlap significantly with the eigenspace of $\boldsymbol{W}$, thus the projection onto the leading eigenvector of $\boldsymbol{W}$ may not effectively separate the classes unless the lines are well-separated. This can be addressed in the following two ways.

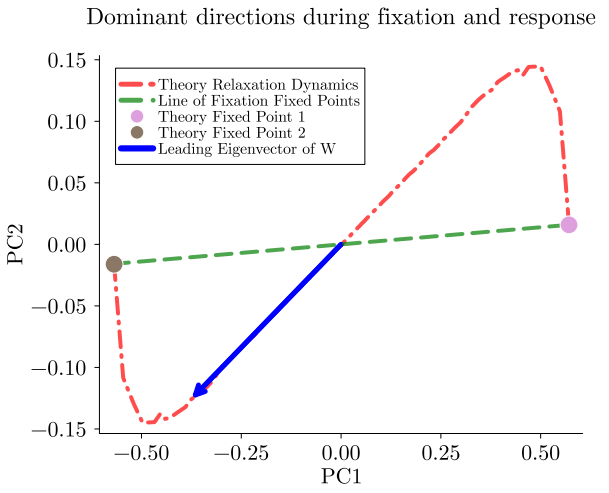

Figure 15: Example dominant directions of dynamics during fixation (green dashed line) and response (blue arrow).

1. Constructing $\boldsymbol{W}_{\text{in}}$ to maximise orthogonality

We can tackle this with a simple approach: given $\boldsymbol{W}$, can we construct $\boldsymbol{W}_{\text{in}}$ so as to maximise orthogonality between $\boldsymbol{A} = (\boldsymbol{I} - \boldsymbol{W})^{-1}\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{v}_1$, the leading eigenvector of $\boldsymbol{W}$? Specifically, we require:

$$\boldsymbol{v}_1^\top (\boldsymbol{I} - \boldsymbol{W})^{-1}\boldsymbol{W}_{\text{in}} = 0. \tag{34}$$

which means that $\boldsymbol{W}_{\text{in}}$ must lie in the nullspace of $\boldsymbol{v}_1^\top (\boldsymbol{I} - \boldsymbol{W})^{-1}$. This can be interpreted as a linear constraint on $\boldsymbol{W}_{\text{in}}$, which can be satisfied by projecting any initial input vector

$\boldsymbol{W}_{\text{in}}^{(0)}$ onto the nullspace of the constraint:

$$\boldsymbol{W}_{\text{in}} = \boldsymbol{W}_{\text{in}}^{(0)} - \frac{\left[\boldsymbol{v}_1^\top (\boldsymbol{I} - \boldsymbol{W})^{-1} \boldsymbol{W}_{\text{in}}^{(0)}\right]}{\left\|(\boldsymbol{I} - \boldsymbol{W})^{-\top} \boldsymbol{v}_1\right\|^2} \cdot (\boldsymbol{I} - \boldsymbol{W})^{-\top} \boldsymbol{v}_1. \tag{35}$$

This yields a choice of $\boldsymbol{W}_{\text{in}}$ such that the input-driven fixed points lie on a subspace maximally orthogonal to the dominant relaxation mode. In doing so, the geometry of fixation and response dynamics becomes naturally separable, improving the conditions for constructing effective linear readouts without relying on simulated trajectories. As $\boldsymbol{v}_1$ is the vector with which the response dynamics predominantly align, we may then employ this as a readout layer for a theoretically motivated decoding of the network's decision.

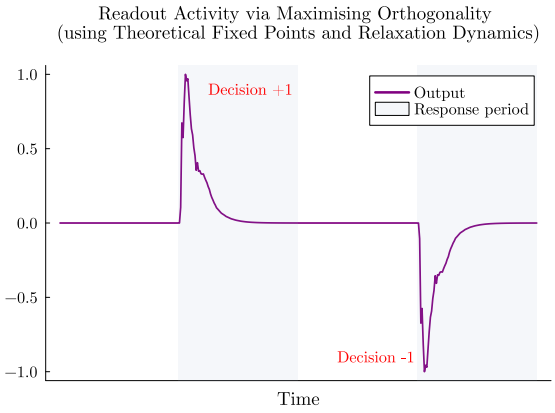

Figure 16: Example output using theoretical fixed points and relaxation dynamics, orthogonally-constructed $\boldsymbol{W}_{\text{in}}$, and the leading eigenvector of $\boldsymbol{W}$ as $\boldsymbol{W}_{\text{out}}$.

2. Constructing an orthogonal projection from activity

A second approach is by using simulation data itself, attempting to find an optimal projection vector using this actual neural activity. We partition the time indices for the simulated neural activity into two disjoint sets: $\mathcal{T}_{\text{fix}}$, corresponding to all fixation periods, and $\mathcal{T}_{\text{resp}}$, corresponding to all response periods. All state vectors are centred by subtracting the global mean state, $\bar{\mathbf{x}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{x}(t)$. We then compute the covariance matrices for each epoch, via

$$\mathbf{Cov}_{\text{fix}} = \frac{1}{|\mathcal{T}_{\text{fix}}|} \sum_{t \in \mathcal{T}_{\text{fix}}} (\boldsymbol{x}(t) - \bar{\boldsymbol{x}})(\boldsymbol{x}(t) - \bar{\boldsymbol{x}})^\top \tag{36}$$

$$\mathbf{Cov}_{\text{resp}} = \frac{1}{|\mathcal{T}_{\text{resp}}|} \sum_{t \in \mathcal{T}_{\text{resp}}} (\boldsymbol{x}(t) - \bar{\boldsymbol{x}})(\boldsymbol{x}(t) - \bar{\boldsymbol{x}})^\top \tag{37}$$

or using the theoretical covariance calculations presented in Section 2.5. Our objective is to find the vector $\mathbf{w}$ that maximises the Rayleigh Quotient:

$$R(\mathbf{w}) = \frac{\mathbf{w}^\top \mathbf{Cov}_{\text{resp}} \mathbf{w}}{\mathbf{w}^\top \mathbf{Cov}_{\text{fix}} \mathbf{w}} \tag{38}$$

The maximisation of this ratio is a classic problem that is solved by the generalised eigenvalue problem:

$$\mathbf{Cov}_{\text{resp}} \mathbf{w} = \lambda \mathbf{Cov}_{\text{fix}} \mathbf{w} \tag{39}$$

The vector $\mathbf{w}$ that maximises Equation 38 is the generalised eigenvector corresponding to the largest generalised eigenvalue, $\lambda_{\max}$. The readout vector, $\mathbf{W}_{\text{out}}$, can then be set to the principal generalised eigenvector found by solving Equation 39. The scalar readout is then computed, as before, by applying $\boldsymbol{W}_{\text{out}}$ linearly to the state activity $\boldsymbol{x}(t)$. This procedure yields a readout that, by construction, isolates the neural dynamics most relevant to the decision-making phase of the task.

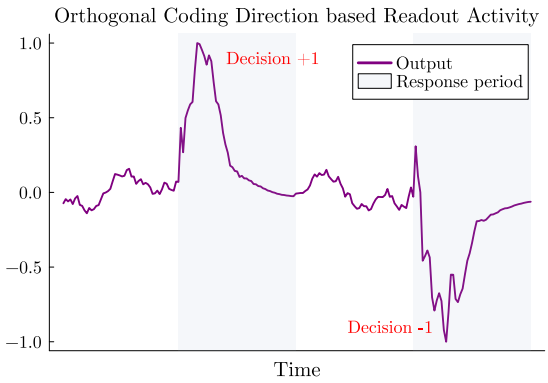

Figure 17: Example output using simulated activity and the orthogonal coding direction-based W_{out}.

Using a Biophysically-Realistic Learning Rule

As an alternative to solving the generalised eigenvalue problem, we can derive an effective readout vector $\boldsymbol{W}_{\text{out}}$ using the online learning mechanism of Sanger's rule in a structured, two-stage process. First, we apply Sanger's rule to the neural activity recorded exclusively during the fixation periods ($\mathcal{T}_{\text{fix}}$). This initial stage learns the principal components of the fixation dynamics ($\boldsymbol{P}_{\text{fix}}$), effectively identifying the subspace that our readout should ideally ignore. Second, we create a new, orthogonalised dataset for the response activity, by projecting the response dynamics into the nullspace of $\boldsymbol{P}_{\text{fix}}$, i.e. for each state vector $\boldsymbol{x}(t)$ during the response period, we calculate $\boldsymbol{x}_{\text{ortho}}(t) = \boldsymbol{x}(t) - \boldsymbol{P}_{\text{fix}}\boldsymbol{x}(t)$. While this may require additional biophysically plausible mechanisms to Sanger's rule, this projection into

the nullspace could readily be implemented by a feedforward inhibitory circuit, where an interneuron learns to detect and subsequently cancel the predictable nuisance dynamics of the fixation period from the input to the readout neuron. Then, we apply Sanger's rule a second time to the orthogonalised response activity. The first principal component found in this step becomes our readout vector, $\boldsymbol{W}_{\text{out}}$. By construction, this vector captures the direction of maximal variance within the response dynamics that is also maximally orthogonal to the dominant fixation dynamics. Figure 18 shows an example output with $\boldsymbol{W}_{\text{out}}$ generated by the above approach.

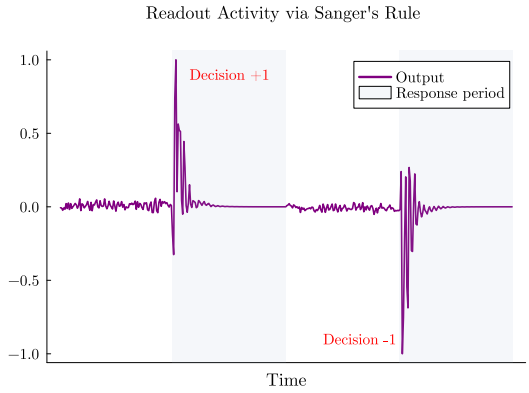

Figure 18: Example output using simulated activity and Sanger's rule-based W_{out}.

5.3 (c) Context-Dependent Decision-Making

5.3.1 Task Description

The context-dependent task is similar to the decision-making task already analysed (Section 5.1), however, in this case there are three inputs, two of which are noisy signals (with means μ_1, μ_2 and variance σ_{in}^2, as in the decision-making case), and the third presents a "contextual cue" to the network, instructing which of the two other signals to attend to. The network must integrate the signal indicated by the context cue, and determine whether the mean average of the attended-to-input is greater than or less than 0, giving a positive output is $\mu_i > 0$ and a negative output is $\mu_i < 0$. The additional complexity in this task arises from the fact that the input is higher dimensional, giving rise to a more interesting state space and landscape of fixed points. We note that this task can be augmented as a "delayed" version (with fixate and response periods) as the task in Section 5.2, but here we limit our attention to the non-delayed version.

5.3.2 Theoretical Fixed Points and Simulation

In the context-dependent task, we consider two context conditions (+1 and −1) with two stimuli, s_1 and s_2, giving two possible input vectors:

$$\mathbf{u}_1^* = [+1, s_1, s_2]^T \quad \text{(attend stimulus 1)}$$
$$\mathbf{u}_2^* = [-1, s_1, s_2]^T \quad \text{(attend stimulus 2)}$$

The corresponding fixed points are:

$$\boldsymbol{x}_1^* = (\boldsymbol{I} - \boldsymbol{W})^{-1}\boldsymbol{W}_{\text{in}}[+1, s_1, s_2]^T$$
$$\boldsymbol{x}_2^* = (\boldsymbol{I} - \boldsymbol{W})^{-1}\boldsymbol{W}_{\text{in}}[-1, s_1, s_2]^T$$

If we let $\boldsymbol{A} = (\boldsymbol{I} - \boldsymbol{W})^{-1}$ and denote the columns of $\boldsymbol{W}_{\text{in}}$ as $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3$, then:

$$\boldsymbol{x}_1^* = \boldsymbol{A}\boldsymbol{w}_1 + s_1\boldsymbol{A}\boldsymbol{w}_2 + s_2\boldsymbol{A}\boldsymbol{w}_3$$
$$\boldsymbol{x}_2^* = -\boldsymbol{A}\boldsymbol{w}_1 + s_1\boldsymbol{A}\boldsymbol{w}_2 + s_2\boldsymbol{A}\boldsymbol{w}_3$$

A key insight is that the difference between corresponding fixed points is constant:

$$\boldsymbol{x}_1^* - \boldsymbol{x}_2^* = \boldsymbol{A}\boldsymbol{w}_1 - (-\boldsymbol{A}\boldsymbol{w}_1) = 2\boldsymbol{A}\boldsymbol{w}_1$$

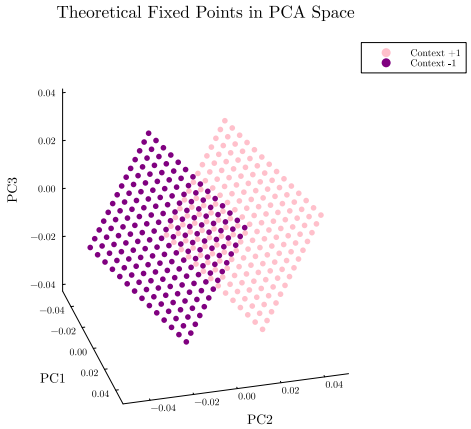

Figure 19: Example of how the fixed points of different stimuli μ_1, μ_2 (between −1 and 1) lie on parallel planes in 3D PCA space for the two different context cues.

This difference is independent of the stimulus values s_1 and s_2. Therefore, the fixed points for the two contexts form two affine subspaces (or planes) that differ by a translation along the vector $2\boldsymbol{A}\boldsymbol{w}_1$. Because this translation is constant, the two planes are parallel. This geometric structure emerges directly from the system properties and is independent of coordinate system choice. The context signal (± 1) determines which plane contains the fixed point, while the stimulus combinations (s_1, s_2) determine the specific location within each plane. Figure 19 demonstrates these parallel planes for an example network $\boldsymbol{W}$ and theoretic

fixed points in PCA space for mean stimuli μ_1, μ_2 both ranging from -1 to 1.

If we set $\mu_1 = \pm\zeta_1, \mu_2 = \pm\zeta_2$, then we have eight different possible input configurations, which will give rise to eight different possible fixed points in the state space. If we consider Context $+1$ fixed points on one plane (e.g. the pink plane in Figure 19), then we can focus on four possible fixed points, namely, $\boldsymbol{x}_{+\zeta_1,+\zeta_2}, \boldsymbol{x}_{+\zeta_1,-\zeta_2}, \boldsymbol{x}_{-\zeta_1,+\zeta_2}, \boldsymbol{x}_{-\zeta_1,-\zeta_2}$. We will denote these as $\boldsymbol{x}_{++}, \boldsymbol{x}_{+-}, \boldsymbol{x}_{-+}, \boldsymbol{x}_{--}$ for simplicity.

Let us take the $\mathbf{0}$ stimulus ($\mu_1 = \mu_2 = 0$) as a reference point: $\boldsymbol{x}_{0,0} = \boldsymbol{A}\boldsymbol{w}_1$. We then study the four fixed points *relative* to $\boldsymbol{x}_{0,0}$, which are:

$$\begin{aligned}
\boldsymbol{x}_{++} - \boldsymbol{x}_{0,0} &= \zeta_1 \boldsymbol{A}\boldsymbol{w}_2 + \zeta_2 \boldsymbol{A}\boldsymbol{w}_3 \\
\boldsymbol{x}_{+-} - \boldsymbol{x}_{0,0} &= \zeta_1 \boldsymbol{A}\boldsymbol{w}_2 + -\zeta_2 \boldsymbol{A}\boldsymbol{w}_3 \\
\boldsymbol{x}_{-+} - \boldsymbol{x}_{0,0} &= -\zeta_1 \boldsymbol{A}\boldsymbol{w}_2 + \zeta_2 \boldsymbol{A}\boldsymbol{w}_3 \\
\boldsymbol{x}_{--} - \boldsymbol{x}_{0,0} &= -\zeta_1 \boldsymbol{A}\boldsymbol{w}_2 + -\zeta_2 \boldsymbol{A}\boldsymbol{w}_3
\end{aligned}$$

These four vectors form a grid or parallelogram in the subspace spanned by $\boldsymbol{A}\boldsymbol{w}_2$ and $\boldsymbol{A}\boldsymbol{w}_3$, which is centred at $\boldsymbol{x}_{0,0}$. The vertices of this grid are located at coordinates $(\pm\zeta_1, \pm\zeta_2)$ in the $(\boldsymbol{A}\boldsymbol{w}_2, \boldsymbol{A}\boldsymbol{w}_3)$ coordinate system. Since, as already alluded to, the affine subspaces of fixed points for Context 1 and -1 differ by a translation along the vector $2\boldsymbol{A}\boldsymbol{w}_1$, we have that the orientation of the four fixed points in the Context 1 plane is preserved in the four fixed points in the Context -1 plane. This is illustrated in Figure 20, which displays the grids of theoretical fixed points for combinations of input means

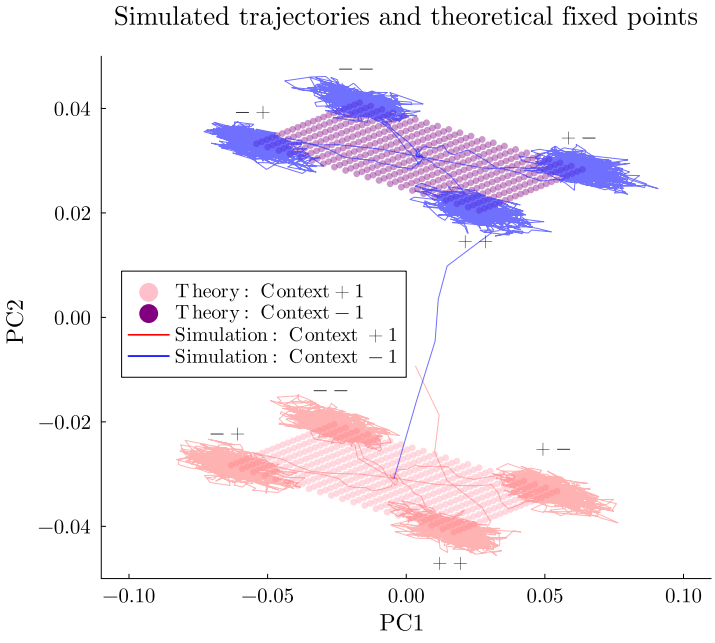

Figure 20: Example simulated trajectories of network states for $\mu_{1,2} = \pm 1$ on top of a grid of theoretical fixed points for values of $\mu_{1,2}$ from -1 to $+1$.

from -1 to $+1$, along with simulated trajectories around all eight possible configurations of input, where $\mu_{1,2} = \pm 1$ and variance $\sigma_{in}^2 = 0.5$. The couplets of $+/-$ symbols denote the configuration of input, for example, $+-$ signifies the fixed point for $\mu_1 = +1$, $\mu_2 = -1$. This enables us to see that the orientation of the geometry of fixed points is preserved between the two planes. This presents an intriguing challenge as to how the linear readout layer can extract relevant information from the dynamics in order to correctly "decide" and solve the task. This is elaborated upon in Section 5.3.3.

5.3.3 Constructing $\boldsymbol{W}_{\text{out}}$ from the Geometry of Dynamics

Given the geometric view of the network dynamics presented in Section 5.3.2, we observe that the orientation of fixed point coordinates is preserved between the two contextual subspaces/planes, as seen in Figure 20. If we use the same example, we can consider how each fixed point relates to the correct/target output "decision" of the network (Figure 21). In order to correctly complete the task, the readout layer $\boldsymbol{W}_{out}$ must effectively compute

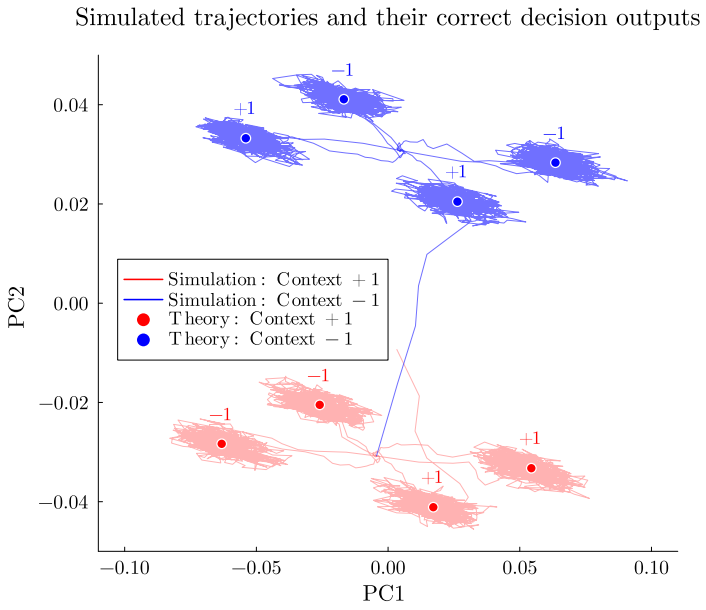

Figure 21: The same trajectories as in Figure 20, but with target output annotated.

6 Example Task: Time-Series Prediction

7 Conclusion

References

[1] J. Kadmon and H. Sompolinsky, "Transition to chaos in random neuronal networks," *Physical Review X*, vol. 5, no. 4, p. 041030, 2015.

[2] H. Sompolinsky, A. Crisanti, and H.-J. Sommers, "Chaos in random neural networks," *Physical review letters*, vol. 61, no. 3, p. 259, 1988.