A THEORY OF PARAMETER IDENTIFIABILITY IN DATA-CONSTRAINED RECURRENT NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

An increasingly common approach in neuroscience seeks to understand the brain by training recurrent neural networks (RNNs) to reproduce observed neural activity. Unlike brains, these RNNs can be computationally poked and perturbed to reveal principles central to their function. However, whether the insights gained from these RNNs truly apply to biological neural circuits remains an open question. The answer hinges on a key distinction: which RNN parameters are uniquely determined by the data they are trained on (*i.e.*, identifiable), and which are unconstrained? To this end, we develop a framework that isolates identifiable subspaces of the RNN parameters, leading to several key findings: (i) commonly used RNN estimators have unconstrained parameters and the dimensionality of training data, *i.e.*, the trajectories in neural state space, dictates the extent of parameter constraints; (ii) we can design RNN estimators to remain confined to identifiable components; (iii) we propose intervention experiments to expand the identifiable subspace; and (iv) we prove that changes in non-identifiable components preserve dynamics on identifiable subspaces but can introduce spurious structure elsewhere. Together, these results delineate regions of state space where RNN predictions are reliable and pinpoint where they are not. Our theory shows that current RNNs are not valid proxies of neural circuits, as their predictions and interpretation can be swayed by non-identifiable components. These results define guidelines for the responsible use of RNN models in neuroscience.

1 INTRODUCTION

Recent advances in large-scale neural recording allow researchers to measure brain-wide activity in animals (Kim & Schnitzer, 2022; Manley et al., 2024; Bounds & Adesnik, 2024; Stringer et al., 2019). Computational neuroscientists have developed methods to analyze these high-dimensional recordings and gain mechanistic insights into neural computation (Schneider et al., 2023; Gardner et al., 2022; Mante et al., 2013; Sussillo & Barak, 2013; Pandarinath et al., 2018). A key conceptual advance is the view that the brain represents information at the level of neural populations, rather than individual neurons (Saxena & Cunningham, 2019; Pouget et al., 2000; Kira et al., 2023; Churchland et al., 2012; Averbeck et al., 2006). This view analyzes dynamical properties of population-activity patterns to understand how the brain solves the task at hand (Liu et al., 2024; Nair et al., 2023; Langdon et al., 2023; Vyas et al., 2020; Khona & Fiete, 2022), revealing, for instance, how a line attractor in the hypothalamus might encode aggression in male mice (Vinograd et al., 2024). Hence, modeling population dynamics, even from partially observed neural activity, may offer a path to reverse-engineering algorithms in the brain (Durstewitz et al., 2023; Dinc et al., 2025).

A prominent approach in systems neuroscience fits recurrent neural networks (RNNs) as dynamical models to reproduce recorded neural activity trajectories. This approach then treats the resulting trained RNNs as in silico analogues of the biological circuits (Rajan et al., 2016; Daie et al., 2021; Perich et al., 2021; Perich & Rajan, 2020; Valente et al., 2022; Duncker & Sahani, 2021; Cohen et al., 2020; Finkelstein et al., 2021; Dinc et al., 2023; Kim et al., 2023; Liu et al., 2024; Linderman et al., 2017). These RNN models have been used to analyze the structure of population dynamics, including attractors underlying neural activity (Valente et al., 2022; Nair et al., 2023), flow fields governing responses to perturbations (Kim et al., 2023; Linderman et al., 2017), and communication patterns across brain regions (Perich et al., 2021; Perich & Rajan, 2020). Critically, their predictions are increasingly used to guide causal experiments Walker et al. (2019); Liu et al. (2024); Vinograd et al. (2024), though their internal structure is not guaranteed to reflect ground truth mechanisms (Das & Fiete, 2020; Qian et al., 2024; Brinkman et al., 2018; Göring et al., 2024). A key failure

mode arises when some model parameters are not constrained by the data distribution; this is the problem of parameter identifiability, the focus of this work.

Our contributions are as follows: we examine when data-constrained RNN (dRNN) parameters are constrained by their neural datasets. We then address estimation from finite and noisy recordings, suggesting how estimation can be engineered to confine parameters to their identifiable components. Finally, we derive two experimental insights: (i) targeted interventions can reveal non-identifiable parameter subspaces, and (ii) there exist parameter subspaces where variation yields the same predictions, and those where they differ. Understanding when and why such divergences occur is essential because treating unconstrained parameters as mechanistic insight can mislead interpretation and waste experimental effort.

2 BACKGROUND

We introduce the RNNs used for our theorems, along with essential concepts from estimation theory.

Recurrent Neural Networks (RNNs) The focus of this work is to analyze RNNs that are commonly used to reproduce observed neural activity (Perich et al., 2021; Valente et al., 2022; Duncker & Sahani, 2021; Cohen et al., 2020; Finkelstein et al., 2021; Dinc et al., 2023). Accordingly, we consider a biologically motivated and interpretable class of RNNs (Perich et al., 2021; Dinc et al., 2025), characterized by the time evolution equation:

$$\tau \dot{r}(t) = -r(t) + \phi(W^{\mathrm{rec}} r(t) + W^{\mathrm{in}} u(t) + \epsilon_{\mathrm{in}}(t)) + \epsilon_{\mathrm{conv}}(t), \tag{1}$$

where $\tau \in \mathbb{R}$ refers to the neuronal decay time constant, $r(t) \in \mathbb{R}^{N_{\mathrm{rec}}}$ the neural activities and $\dot{r}(t) \in \mathbb{R}^{N_{\mathrm{rec}}}$ their time derivatives, $u(t) \in \mathbb{R}^{N_{\mathrm{in}}}$ the inputs, $W^{\mathrm{rec}} \in \mathbb{R}^{N_{\mathrm{rec}} \times N_{\mathrm{rec}}}$ the recurrent weights, $W^{\mathrm{in}} \in \mathbb{R}^{N_{\mathrm{rec}} \times N_{\mathrm{in}}}$ the input weights, $\epsilon_{\mathrm{in/conv}}(t) \in \mathbb{R}^{N_{\mathrm{rec}}}$ some unknown input/conversion noise terms, and $\phi(\cdot)$ a monotonic nonlinearity. In this definition, we omit the bias term without loss of generality, as it can be incorporated by fixing one of the inputs to one.

For notational convenience, we define the concatenated vector $x(t) = [r(t), u(t)] \in \mathbb{R}^{N_{\mathrm{rec}}+N_{\mathrm{in}}}$ and the parameter matrix $\theta = [W^{\mathrm{rec}}, W^{\mathrm{in}}] \in \mathbb{R}^{N_{\mathrm{rec}} \times (N_{\mathrm{rec}}+N_{\mathrm{in}})}$, so that the input to the nonlinearity becomes $z(t) = \theta x(t) + \epsilon_{\mathrm{in}}$. In what follows, we denote $N_X = N_{\mathrm{rec}} + N_{\mathrm{in}}$. In practice, neural activity data is discretized in time. Hence, we introduce discrete RNN models resulting from the Euler discretization of Eq. 1:

$$r[s+1] = (1-\alpha) r[s] + \alpha \phi(z[s]) + \epsilon_{\mathrm{conv}}, \tag{2}$$

where we perform the discretization via $r[s] = r(s \cdot \Delta t)$, where we denote the discretized time scale as $\alpha = \Delta t / \tau$ and $s \in \mathbb{N}$ refers to the discretized time.

Identifiability. We introduce the notion of identifiability and explain how it applies to these RNNs. Intuitively, identifiability is about whether you can uniquely determine the parameters of a model from the observed data. If a model is identifiable, then, given enough data, there is only one set of parameters that could produce that data. If a model is non-identifiable, then there are multiple different sets of parameters that could produce the same observations.

Definition 1 (Identifiability (Lehmann & Casella, 2006))**.** *Let $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$ be a model, or family of parameterized probability distributions, with parameter space Θ. $\mathcal{P}$ is identifiable if and only if the mapping $\theta \mapsto P_\theta$ is injective, i.e., if*

$$P_{\theta_1} \hat{=} P_{\theta_2} \quad \Rightarrow \quad \theta_1 = \theta_2 \quad \textit{for all } \theta_1, \theta_2 \in \Theta, \tag{3}$$

where $\hat{=}$ means equal in distribution.

Identifiability of θ can often be achieved under certain "identification conditions." For instance, for a family of distributions P_θ that satisfy the condition $P_\theta \hat{=} P_{-\theta} \hat{=} P_{|\theta|}$, one may enforce $\theta \geq 0$ as an identification condition. Identifiability in RNN parameters often requires an identification condition, which we will expand upon in Section 4.

3 RELATED WORKS

We review related work on models trained to reproduce neural activity and their ability to generate biological insights, followed by control and systems theory on linking observable behavior of

a system (*i.e.*, data) to underlying model states and dynamics. An extended review, including additional background on identifiability in nonlinear systems and task-trained RNNs, is provided in Appendix A.

Models of neural activity. Models trained to reproduce neural activity have long promised insight into biological and computational mechanisms, though their interpretability remains debated. Recovering synaptic connectivity from observed dynamics is generally ill-posed (Das & Fiete, 2020; Brinkman et al., 2018), and functional properties, such as presumed underlying attractors, can be unreliable when inferred from data alone (Qian et al., 2024; Göring et al., 2024). Even physically constrained RNNs with a one-to-one mapping between recorded and modeled neurons (Perich & Rajan, 2020; Perich et al., 2021; Dinc et al., 2023) remain poorly understood in terms of identifiability. Nevertheless, RNNs and other predictive models have been used to uncover putative mechanistic features, including population-level gating mechanisms (Finkelstein et al., 2021), inter-area communication motifs (Perich & Rajan, 2020), and low-dimensional attractor dynamics (Valente et al., 2022). In some cases, the predictions of these models have been refined and causally validated through intervention (Daie et al., 2021; Liu et al., 2024; Vinograd et al., 2024; Walker et al., 2019), but in others models can result in misleading interpretations. This ambiguity highlights a paradox: RNNs trained on neural data can yield genuine mechanistic insights, misleading interpretations, and sometimes both.

Identifiability of nonlinear systems. The question of whether models are uniquely determined by data has long been studied in control theory. Classical realization theory results show that any external behavior generated by a finite-dimensional system can be represented by a "minimal" and unique system, which must be controllable and observable (Sussmann, 1976). Such a minimal model can often be found by restricting the parameters to the quotient space of the original model space and the equivalence relation of indistinguishability, or, equivalently, by reducing the state space to the manifold occupied by the lower-dimensional underlying system (Crouch, 1979; Brockett, 2005). For neural networks specifically, identifiability has been examined under specific conditions (Sussmann, 1992; Poznyak et al., 2001; Albertini & Sontag, 1993). This analysis excluded "degenerate situations", such as those with parameter dependencies, nonobservability, and underlying low-dimensionality–all of which occur in real-world neural data (Dubreuil et al., 2020; Perich et al., 2025). Parameter symmetries have been characterized in both recurrent (Al-Falou & Trummer, 2003; Biswas & Fitzgerald, 2022) and feedforward architectures (Bui Thi Mai & Lampert, 2020; Bona-Pellissier et al., 2023). Recent studies have highlighted how RNN dynamics and behavior are only partially constrained by partial input–output observations (Rajan et al., 2010; Kepple et al., 2022), leading to parameter ambiguity, and have proposed frameworks to measure, understand, and intervene on solution degeneracy in task-trained RNNs (Huang et al., 2025)

4 THEORY

Consider a ground-truth RNN model with parameters θ^* and dynamics as defined in Eq 2. We consider two empirically relevant questions: i) Are the parameters θ^* identifiable in the absence of noise? ii) Can we define identification conditions such that an estimation procedure recovers an identifiable solution $\hat{\theta}$? Here, we answer both questions.

4.1 CONDITIONAL IDENTIFIABILITY FOR RNNS

To apply the framework of identifiability as defined in Definition 1 to RNNs, we associate a probability distribution with the discretized dynamics of Eq. 2. The model parameters θ govern the mapping $x[s] = [r[s], u[s]] \mapsto r[s+1]$ under these dynamics. We therefore define the family of conditional probability distributions $\mathcal{P}_{\mathcal{X}} = \{P(Y|x;\theta), \forall x \in \mathcal{X}, \theta \in \Theta\}$, on a random variable Y, where $\mathcal{X}$ denotes the domain of conditioning variables. In the case of RNNs, $\mathcal{X}$ corresponds to the joint space of inputs and neural states, $x = [u, r] \in \mathcal{X}$. This motivates the following definition:

Definition 2 (Conditional Identifiability)**.** *Given a conditioning domain $\mathcal{X}$, let $\mathcal{P}_{\mathcal{X}} = \{P(Y|x;\theta), \forall x \in \mathcal{X}, \theta \in \Theta\}$ be a model with parameter space Θ. The model is conditionally identifiable if and only if, for every $x \in \mathcal{X}$, the mapping $\theta \mapsto P(Y \mid x;\theta)$ is injective.*

When θ is multidimensional (e.g., $\theta = [W^{\text{rec}}, W^{\text{in}}]$ in RNNs) it may be that only certain components of, or directions within, θ are identifiable. In such cases, the model as a whole is not identifiable, but identifiable parameter combinations can still be defined. In this paper, we will characterize which subsets of θ are identifiable.

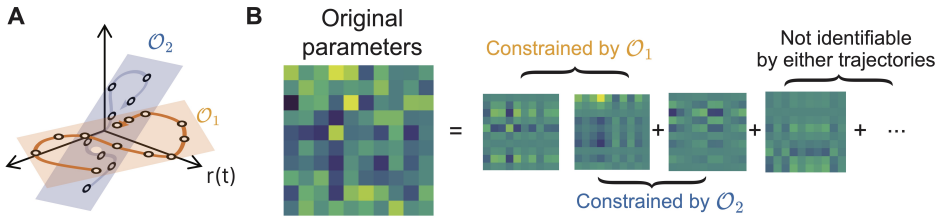

Figure 1: **Neural trajectory subspaces constrain parameters. A** Observed neural trajectories $\mathcal{O}_i$ for $i \in \{1, 2\}$ are confined to linear subspaces. **B** Parameters can be decomposed into components constrained by the observed neural data and an unconstrained, non-identifiable remainder.

4.2 Neural activity subspaces constrain linear combinations of parameters

We establish a theorem that characterizes the identifiable and non-identifiable parameter directions in $\theta^* = [W^{\text{rec}}, W^{\text{in}}]$, the recurrent and input weight matrices of the ground-truth RNN. We formulate this notion in terms of conditional identifiability, where the conditioning domain is the set of observed input–activity data $\mathcal{X} = \left\{x^{(m)}[s] = [u^{(m)}[s], r^{(m)}[s]] \mid s = 1, \ldots, T, m = 1, \ldots, M\right\}$ at any time $s = 1, \ldots, T$ and across all experimental trials $m = 1, \ldots, M$. We define $X \in \mathbb{R}^{TM \times N_X}$ to be the observation matrix, each row containing an $x^{(m)}[s]$ for $s = 1, \ldots, T$ and $m = 1, \ldots, M$.

Theorem 1 (Conditional Identifiability in RNNs)**.** *Consider the noiseless RNN defined by Eq. 2 with parameters θ^*. Consider X an observation matrix defining the conditioning space $\mathcal{X}$ and denote $P \in \mathbb{R}^{N_X \times N_X}$ the projection matrix onto its column space. Then, any RNN parametrized by θ such that:*

$$\theta = \theta^* P + \Delta\theta \quad \textit{for some } \Delta\theta \textit{ that verifies} \quad \Delta\theta P = 0, \tag{4}$$

gives the same conditional probability distribution as the ground-truth RNN. θ^* is conditionally identifiable if one of the following holds: (i) P is full rank, in which case $P = I$, or (ii) the parameter space is restricted to $\{\theta \in \Theta : \theta(I - P) = 0\}$ (identifiability condition).

The proof for theorem 1, and a Corollary extending it to networks with unobserved influence, including partial observations, are provided in Appendix C. Since Theorem 1 applies to noiseless neural data, the concept of whether a parameter is conditionally identifiable under X simplifies into determining whether changing this parameter will change the neural trajectories that have been observed in X. Theorem 1 characterizes the equivalence class of RNNs that generate X.

As illustrated in Fig. 1, we find that the linear subspaces of observed neural activities constrain linear combinations of RNN parameters. In what follows, we refer to the subspace defined by the relationship $\theta = \theta P$ as the identifiable parameter subspace, and $\tilde{\theta} := \theta^* P$ as the identifiable component of the ground truth parameters θ^*.

Theorem 1 gives practical insights. Even with noiseless data, the richness of the neural dataset (quantified by the rank of the projection matrix P) imposes fundamental limits on parameter recovery. It determines when and in which regions of state space the *novel* predictions of estimated models can be trusted. Any RNN with estimates $\hat{\theta}$ that are such that $\hat{\theta}(I - P) \neq 0$ will make predictions that are unconstrained by the observed data, i.e., it will "hallucinate" dynamics. Next, we will discuss how to avoid such hallucinations.

4.3 Estimation of RNN parameters using only identifiable components

Theorem 1 ties identifiability to the rank of P. Higher rank means fewer RNNs can reproduce the data X. Lower rank means that many can and that the trained RNN model is just one of them. To evaluate in which scenario we are, we propose a method to estimate the rank of P.

For simplicity of exposition, we consider here that the entries of the observation matrix for one trial are i.i.d. distributed as $X_{ij} = \tilde{X}_{ij} + \epsilon_{ij}$, where $\tilde{X} \in \mathbb{R}^{T \times N_X}$ is now the noiseless component from Theorem 1 and ϵ_{ij} is noise with mean 0 and variance σ_ϵ^2. We present the analysis with spatiotemporal correlations and unobserved influence in Appendix B.2.

Because P is the orthogonal projection to the column space of the noiseless observation matrix $\tilde{X}$, we have that $\text{rank}(P) = \text{rank}(\tilde{X})$. To estimate this rank, we consider the singular value de-

composition $X = \sum_{r=1}^{R_X} \sigma_r v^{(r)} u^{(r)T}$, where $X \in \mathbb{R}^{MT \times N_X}$, $v^{(r)} \in \mathbb{R}^{MT}$, $u^{(r)} \in \mathbb{R}^{N_X}$, and $R_X = \min(MT, N_X)$. In the absence of noise $X = \tilde{X}$, finding the rank of P reduces to counting the nonzero singular values σ_r of X. With noise, however, every σ_r becomes nonzero, obscuring the rank of the noiseless $\tilde{X}$.

To address this issue, we consider the Gram matrix of the columns of X, denoted $G = X^T X$, which has expectation:

$$\mathbb{E}[G] = \mathbb{E}[X^T X] = \mathbb{E}\left[(\tilde{X} + \epsilon)^T (\tilde{X} + \epsilon)\right] = \mathbb{E}\left[\tilde{X}^T \tilde{X} + \epsilon^T \tilde{X} + \tilde{X}^T \epsilon + \epsilon^T \epsilon\right] = \mathbb{E}[\tilde{G}] + MT\sigma_\epsilon^2 I, \tag{5}$$

with $\tilde{G} = \tilde{X}^T \tilde{X}$ and σ_ϵ^2 the noise variance. Notably, $\tilde{G}$ scales linearly with MT, as it involves summation over MT terms. Cross-terms involving X_{ij} and ϵ_{ij} vanish in expectation due to zero-mean, data-independent noise, and $T\sigma_\epsilon^2 I$ follows from the i.i.d. assumption. Accordingly, the singular value decomposition of G is:

$$G = X^T X = \sum_{r=1}^{R_X} \sigma_r^2 u^{(r)} u^{(r)T} \quad \xrightarrow{\mathbb{E}} \quad \tilde{G} + \sigma_\epsilon^2 I = \sum_{r=1}^{R_X} \left(\tilde{\sigma}_r^2 + TM\sigma_\epsilon^2\right) \tilde{u}^{(r)} \tilde{u}^{(r)T}, \tag{6}$$

where σ_ϵ^2 is the noise variance, σ_r and $\tilde{\sigma}_r$ are the singular values (scaling linearly with TM) of X and $\tilde{X}$, respectively, and u and $\tilde{u}$ are the eigenvectors of G and $\tilde{G}$ respectively. Once again, we note that it follows that (i) the eigenvectors of the empirical Gram matrix G coincide in expectation with those of the noiseless Gram matrix $\tilde{G}$, and (ii) noise shifts the eigenvalues by σ_ϵ^2. Thus, in expectation, G separates the contributions of signal and noise. Its leading eigenvectors define the column space of $\tilde{X}$, which defines P and accordingly characterizes the identifiable subspace of parameters defined by $\theta = \theta P$, and its eigenvalues provide the rank of P by subtracting the noise variance σ_ϵ^2. In practice, the expectation of the Gram matrix will be estimated by its sample mean across the M trials. For the remainder of this section, we consider that the singular values $\tilde{\sigma}_r$ are ordered and are such that $\tilde{\sigma}_r \gg \sigma_\epsilon$ for all $r \leq R$ and $\tilde{\sigma}_r = 0$ for all $r > R$ where $R = \text{rank}(\tilde{X})$ is the rank of the noiseless $\tilde{X}$.

Theorem 2 (Identifiable estimation of RNN parameters)**.** *Under the assumptions and notation of this section, consider an RNN whose parameters $\theta \in \mathbb{R}^{N_{\text{rec}} \times N_X}$ is estimated by gradient descent of a differentiable loss $L(\theta)$. Assume that the gradient satisfies $v^{(r)T} \nabla L(\theta) = O(\sigma_r^n)$ for every $\theta \in \Theta$ and some integer n. If $\theta^{(k)} P = \theta^{(k)}$ at iteration k of the gradient descent, then for any λ with $TM\sigma_\epsilon^2 \ll \lambda \ll \tilde{\sigma}_R^2$, and for any step $\alpha > 0$, the update*

$$\theta^{(k+1)} = \theta^{(k)} - \alpha \nabla L(\theta) \left(X^T X + \lambda I\right)^{-1}, \tag{7}$$

is a descent direction that satisfies $\theta^{(k+1)} P = \theta^{(k+1)} + O(\sigma_\epsilon^n / \lambda)$.

Theorem 2 relies on some common assumptions. Specifically, the condition that the gradients projected onto singular components $v^{(r)}$ scale as $O(\sigma_r^n)$ follows from the behavior of common estimation methods (*e.g.*, $n = 1$ for least squares, see Corollary 1 below) near their solutions, stemming from the linear parameter-data interaction in Eq. 1. Notably, in this formulation, both λ and $\tilde{\sigma}_R^2$ scale linearly with MT, as the latter involves summation over MT components following Eq. 6.

Corollary 1 (Identifiability in nonlinear regression via local weighted least squares)**.** *In the setting of Theorem 2, suppose the loss is*

$$\mathcal{L}(\theta) = \sum_{i=1}^{TM} L\big(Y_i, g(\theta^T X_i)\big) + \lambda \|\theta\|_2^2, \tag{8}$$

where L is twice differentiable in its second argument, g is smooth, and $\lambda \geq 0$ is a regularization parameter. In the neighborhood of a minimizer $\bar{\theta}$, the objective locally reduces to weighted ridge regression:

$$\bar{\mathcal{L}}(\theta) = \|W^{1/2}(Y - X\theta)\|_2^2 + \lambda \|\theta\|_2^2, \tag{9}$$

with W determined by the curvature of the loss at $\bar{\theta}$. Define $\tilde{\sigma}_\epsilon$ as the singular values of the (noise-free) matrix $W^{1/2}X$ and σ_ϵ the noise component such that $TM\sigma_\epsilon^2 \ll \lambda \ll \tilde{\sigma}_R^2$. Then, the regularized solution approximates an unregularized estimator with only identifiable parameter components.

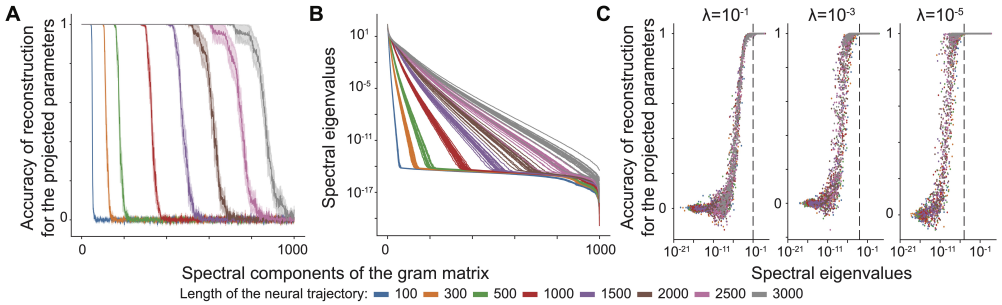

Figure 2: **Gram-matrix spectra determine identifiable parameter subspaces. A** Reconstruction accuracy of the projected parameters vs. spectral components with varying trajectory lengths (T). Solid lines: mean; shaded regions: s.e.m. over 20 randomly initialized RNNs. **B** Eigenvalues of the Gram matrix. Each solid line corresponds to a distinct seed and/or length T. **C** A scatter plot showing reconstruction accuracy of the parameters projected onto spectral components of the Gram matrix ($G = \frac{1}{T}X^TX$) versus the corresponding eigenvalues, where each dot represents one projected parameter estimate. Parameters: Refer to Appendix D.2.6

Proofs for Theorem 2 and a more detailed version of Corollary 1 are provided in Appendix C. Intuitively, Corollary 1 requires only that the loss is twice differentiable and parameters interact with X multiplicatively. Under these conditions, the update rule of Theorem 2 locally coincides with a Hessian-preconditioned (second-order) update, where λ acts as ℓ_2 regularization.

In practice, this setting arises when RNNs in Eq. 1 are trained on single-step prediction: given state–input pair $X_i = [r[i], u[i]]$, the next state $Y_i = r[i+1]$ is independent of history. Thus, *locally*, identifiable RNN training reduces to the same spectral conditions as in linear regression, with the effective Gram matrix determined by the observed neural dataset $\mathcal{D}$.

5 Results

Theorems 1 and 2, along with their corollaries, constitute the backbone of this section. We use them to (i) analyze and categorize existing empirical estimators, (ii) show that standard overfitting controls do not necessarily resolve non-identifiability, and (iii) establish two additional theorems with direct relevance to practical applications of data-constrained RNNs Rajan et al. (2016); Perich et al. (2021); Valente et al. (2022); Duncker & Sahani (2021); Cohen et al. (2020); Finkelstein et al. (2021); Dinc et al. (2023). We note that, throughout our empirical verifications, we normalize the eigenvalues of the Gram matrix by $1/MT$ to account for scaling over number of samples, and use average loss function values during training.

5.1 Empirical validation of identifiability and estimation theory

We use empirical simulations to validate Theorems 1–2 and Corollary 1, and to provide numerical evidence consistent with Theorem 2. We use chaotic RNNs as generators to produce datasets, as they sustain rich but mechanistically ambiguous activity without input, isolating identifiability effects. To estimate our data-constrained RNN (dRNNs), we adopt CORNN (Dinc et al., 2023), a convex optimization method that belongs to the estimator class of Theorem 2 (which stems from the connection this approach to least-squares covered by our Corollary 1) and admits a global minimum. Later and in Appendix B.2, we extend these analyses to task-trained RNNs and extend to alternative estimators.

Gram-matrix spectrum defines identifiable components. We empirically verify that the identifiable component of the parameter space is controlled by the nonzero spectral modes of the Gram matrix (Theorem 1). First, we initialize a ground-truth RNN with randomly sampled weights and collect T samples by Eq. 2 without any noise or input. We use CORNN to estimate the dRNN parameters (see Fig. 2**A-B**). For each dataset, we compute the empirical Gram matrix $G = \frac{1}{T}X^TX$, where X is the observation matrix of concatenated neural activities $\{r[0], r[1], \ldots, r[T-1]\}$. In the noiseless case, G has nonzero eigenvalues along the directions spanned by P and zero eigenvalues along its kernel. Because the spectral components of G capture how strongly different directions are represented in the dataset, larger eigenvalues align with directions within P, while smaller eigenvalues correspond to directions outside P. Theorem 1 predicts projections onto larger eigenvalue directions are identifiable, which we verify.

Firstly, all models trained to reproduce the dataset achieve single-step root mean square error of $\leq 10^{-7} \pm O(10^{-8})$. The projected learned parameters correlate strongly with the projected ground-truth parameters in the directions associated with the leading spectral components, dropping sharply as the eigenvalues approach zero (Fig. 2**A**). The sharp drop shifts as the trajectory length increases; longer trajectories increase G's effective rank. Consistent with this, Fig. 2**B** shows the eigenvalue spectra of G. As T increases, the rank of G increases due to the appearance of additional nonzero eigenvalues, showing that richer trajectories expand the identifiable subspace.

Regularization parameter controls the spectral components used in the estimation. Theorem 2 predicts that estimators using the second-order preconditioning in Eq. 7 suppress contributions from non-identifiable directions, yielding reconstructions confined to the identifiable subspace when the regularization parameter λ is correctly chosen. CORNN is one such estimator as we have shown above. In the absence of noise, the regularization parameter λ directly determines which spectral components contribute to the estimate. As shown in Figs. 2**C** and S3**A**, reconstruction accuracy closely follows the eigenvalue spectrum; components above the effective cutoff set by λ are retained in the solution, whereas those below are suppressed. Increasing λ progressively discards informative spectral modes, reducing reconstruction accuracy. This effect is reflected in Fig. S1**B**, where stronger regularization worsens single-step prediction errors despite perfect training at minimal $\lambda \sim O(10^{-13})$, and in Fig. S3**A** with near perfect reconstruction accuracies up to spectral eigenvalues of the same order. Together, these results confirm Theorem 2, that λ acts as a spectral filter, preventing spurious contributions from near-zero eigenmodes but risking eliminating genuine signal when too high.

Noise permanently limits what is identifiable, regularization mitigates artifacts. With noise, clean separation between identifiable and non-identifiable components is no longer possible. As shown in Fig. S3**A** vs **B**, spectral components accurately reconstructed without noise are now corrupted with noise. Modes with large enough eigenvalues remain reliable, but not lower-eigenvalue directions. Thus, noise irreversibly shrinks the effective dimensionality of the identifiable subspace. At the same time, Fig. S4 shows that without regularization, noise inflates parameter estimates along corrupted directions, producing spurious growth in Frobenius norm. Regularization cannot recover lost components, but it can prevent this uncontrolled amplification. Choosing $\lambda \geq O(\sigma_\epsilon^2)$ suppresses noise-driven contributions, confining the solution to the remaining identifiable subspace (Fig. S3**B**).

5.2 Why standard overfitting controls do not resolve non-identifiability

Standard ℓ_2 (or rank) regularization alone does not guarantee confinement of the solution to the identifiable subspace for a general estimator. Above, we demonstrated a case where the estimator satisfies the conditions of Theorem 2. We now consider estimation approaches that do not in fact resolve the issue of non-identifiability. In Appendix B, we show that low-rank regularization does not eliminate non-identifiability and prove an analog of Theorem 1 for this case. Here, we illustrate how an ℓ_2 penalty on the weights does not guarantee estimation confined to identifiable subspaces.

L2 regularization does not eliminate non-identifiable components in FORCE learning. Most earlier works on dRNNs employ FORCE learning during training Rajan et al. (2016); Perich et al. (2021); Cohen et al. (2020); Finkelstein et al. (2021). This uses an online algorithm to adjust the recurrent weights to match predicted trajectories to observed neural activity. In essence, FORCE approximates the Gram matrix and updates parameters with an implicit ridge penalty on the recursive least-squares Sussillo & Abbott (2009). Although the FORCE update rule follows the same structure as Eq. 7, which stems from the use of recursive least-squares (Sussillo & Abbott, 2009; Rajan et al., 2016; Perich et al., 2021), the network begins from chaotic random initializations. Since FORCE attains low error after few iterations by the use of recursive least-squares (Sussillo & Abbott, 2009), subsequent updates remain in the identifiable subspace (Theorem 2).

Fig. 3 shows the direct implication of this process. We repeat the noise-free estimation from Fig. 2 using a FORCE learner Perich et al. (2021) across normalized regularization strengths λ. In contrast to CORNN, FORCE learning did not suppress the non-identifiable components even when λ was scaled to very large values that decreased the accuracy (Fig. 3**A–B**). As learning converged to correct predictions within the top spectral components, only identifiable parameters continued to receive updates, whereas projections onto the lowest spectral components retained their norms (Fig. 3**B**, contrast FORCE vs. CORNN) and remained highly correlated with their initialization (Fig. 3**C**). This outcome confirms Theorem 2, showing how non-identifiable components present at

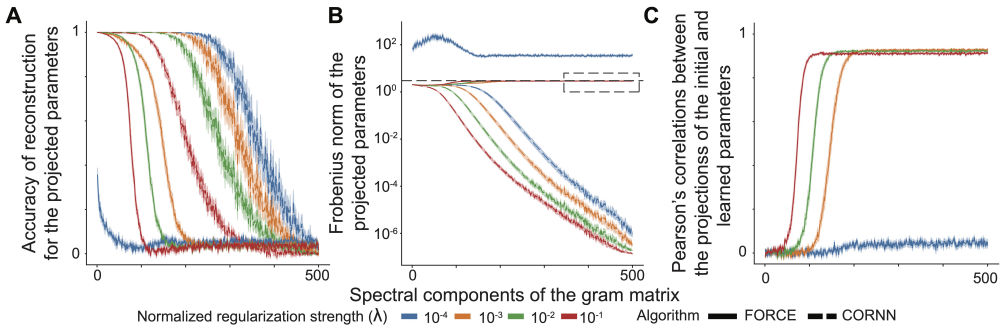

Figure 3: **FORCE learning inevitably learns non-identifiable RNN parameters. A** Reconstruction accuracy of the projected parameters as a function of the spectral components of the training-sample Gram matrix. **B** Frobenius norm of the projected parameters along the spectral components. The boxed region highlights that, for FORCE, the Frobenius norms converge to the (expected) values at initialization (horizontal dashed black line). **C** Pearson correlations between the projections of the initial and learned parameters by FORCE. Parameters: Refer to Appendix D.2.6.

initialization remain largely unchanged. Together, these findings provide conclusive evidence that a popular estimator relies on parameters unconstrained by observed data (being effectively determined by initialization), and ℓ_2 penalties on the weights alone do not resolve identifiability issues.

5.3 Revealing non-identifiable components with targeted interventions

We have shown above that a common estimator (Sussillo & Abbott, 2009; Perich & Rajan, 2020, FORCE) projects arbitrary weights into a randomly initialized subspace, even if it contains non-identifiable directions. In contrast, estimators consistent with Theorem 2 assign zero to those directions. As shown in Figs. S3 and S4 (and discussed further in Appendix B.2), adding random noise to the dynamics can degrade performance. Thus, to resolve the zero-eigenvalue modes of the Gram matrix, we turn to a powerful tool: *deliberate experimental interventions*.

The intervention setup. We focus on generator RNNs trained on the 3-bit flip-flop task, whose neural activities are reconstructed with dRNNs (Fig. 4). Here, the network receives three separate binary input streams, each of which can flip or hold the value of an independent memory bit. This requires the RNN to maintain one of $2^3 = 8$ possible internal states. The networks state is output through three linear readouts. First, we use the generator RNNs to create an observational dataset $\mathcal{D}_{\text{obs}}$ over 5 distinct trials and compute the Gram matrix. We then construct an intervention dataset $\mathcal{D}_{\text{int}}$ by sampling new points $x^{(j)}$ in four cases: (i) additional trials of the network under normal operation (blue in Fig. 4), (ii) projections restricted to the bottom spectral eigenvectors (green), (iii) projections restricted to the top spectral eigenvectors (orange), or (iv) random projections along the spectral components of the Gram matrix (red). Finally, we combine $\mathcal{D}_{\text{obs}}$ and $\mathcal{D}_{\text{int}}$, train dRNNs on the combined datasets, and analyze the reconstruction accuracies of the RNN parameters.

Intervention results showcase the empirical utility of Theorem 1. Intervention strategy critically determines the performance of the dRNNs (Fig. 4**A-B**). When 500 intervention samples are available, all three Gram-matrix–based strategies (but not the one involving extra observational samples; blue in Fig. 4**A**) achieve near-perfect accuracies. This is consistent with Theorem 1; all parameters are identifiable when the data are full rank (*i.e.*, $P = I$ in Theorem 1, albeit with noisy samples). Here, interventions along the top eigenvectors provide little to no benefit ("worst-case"), whereas selecting the bottom eigenvectors ("optimal") accelerates recovery relative to random choices. The lowest eigenvectors of the Gram matrix are more likely to correspond to the null or noisy dimensions (Theorem 1). Increasing the number of interventions progressively aligns dRNN outputs with the ground truth flip-flop states (Fig. 4**C**). Even though the optimal strategy (green) has no information on the task itself, *i.e.*, does not involve samples encountered during task-relevant operation of the dRNNs, dRNNs trained with these samples over-perform those trained with equally more samples collected from task-performing generators. Hence, a task-agnostic parameter disambiguation strategy may provide more information on task-relevant parameters than task-relevant information.

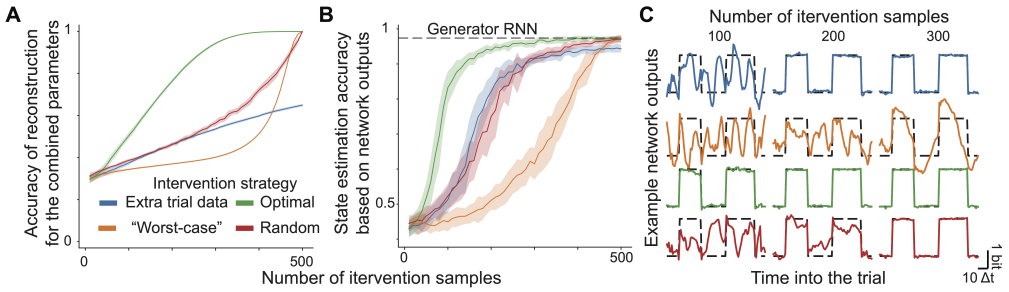

Figure 4: **Targeted interventions can recover non-identifiable parameters of RNNs trained to perform 3-bit flip-flop tasks.** **A** Reconstruction accuracy of parameters $\theta = [W^{\text{rec}}, W^{\text{in}}]$ as a function of the number of intervention samples in training. **B** Corresponding state estimation accuracy (agreement between predicted and target flip-flop states). In **A-B**, solid lines indicate the mean and shaded regions the s.d. across 20 RNNs. **C** Example outputs from trained RNNs for a single trial ($T = 100$); dashed black lines denote the ideal outputs. Parameters: Parameters: Refer to Appendix D.2.6.

5.4 Identifiable dynamics are constrained by observed trajectories

Our findings above lead to a crucial question: Do RNNs that share the same top spectral components also share the same dynamical structures within these components? To answer, we recall that the task-relevant parameters span only the first few top spectral components. We define "identifiable neural activity subspace" to be that which is spanned by the relevant set of spectral eigenvectors via $S_{\text{id}} = \{\sum_{r=1}^{R} a_i v_i | v_i : \text{top } i\text{th spectral eigenvector}\}$, where R is some cutoff chosen empirically. We rephrase the question: Can parameter differences confined to the non-identifiable directions induce distinct dynamical behaviors in the identifiable neural activity subspaces?

Theorem 3 (Preserved dynamics in identifiable neural activity subspaces)**.** *Let $S_{\text{id}} = \text{span}\{v_1, \ldots, v_R\}$ be the identifiable subspace spanned by the top R spectral eigenvectors of the Gram matrix, and assume that for a noiseless, task-performing RNN with dynamics in Eq. 1, the activities satisfy $r[t] \in S_{\text{id}}$ for all t. Let $\tilde{\theta}$ be identifiable with $\tilde{\theta} P_{\text{id}} = \tilde{\theta}$, where P_{id} projects onto S_{id}. Then, any parameterization $\theta = \tilde{\theta} + \Delta\theta$ with $\Delta\theta P_{\text{id}} = 0$ but $\Delta\theta \neq 0$ yields identical dynamics $\dot{r}[t]$ for all $r[t] \in S_{\text{id}}$, but not necessarily when $r[t] \notin S_{\text{id}}$.*

Theorem 3 ties our results to the central motivation for training dRNNs. It shows that identifiability is not only a property of parameter recovery but also of dynamical fidelity. It states that within the identifiable subspace, all RNNs consistent with the training data yield the same dynamics, even for regions of state space not sampled during training, thus enabling principled generalization beyond the observed data. By contrast, outside this subspace, unconstrained parameter variations can induce divergent dynamics, with no empirical support from the observed neural dataset.

Overall, our main theoretical results (Theorems 1, 2, 3) set up a theory of RNN identifiability for experimental and computational neuroscientists seeking to use them as models of dynamical systems. Appendix B provides empirical analyses and ablation studies to test its robustness and scope. There, we examine the effects of different noise models, estimators, and partial observability; extend the framework to low-rank RNNs; and investigate how temporal scales influence parameter estimation. We also show how to use the Gram matrix to reveal solution degeneracy in task-trained RNNs.

6 Conclusion

Our results delineate boundaries of state space where dynamical models can and cannot be trusted, offering a practical guide for the responsible use of data-constrained RNNs as digital twins (Perich & Rajan, 2020). While dRNNs are often framed as generalizing beyond training samples and recovering neural circuit dynamics, no theoretical guarantee had established that the recovered dynamics are consistent, unique, or generalizable. We prove that identifiability governs both consistency and generalization, introducing identifiable and non-identifiable parameters with their associated subspaces. Finally, we propose an empirical intervention framework for probing non-identifiable subspaces, showing that although such components cannot be resolved by computation alone with limited data, they *can* be uncovered through perturbation, opening the way for experimental designs that expand prediction-safe regions.

LIMITATIONS

While our work establishes a general theoretical framework for identifiability in dynamical recurrent neural networks, several limitations remain that should be acknowledged and that point to concrete directions for future research.

Firstly, there is likely a simple but important connection to Takens' theorem in dynamical systems theory Takens (2006), which posits that the attractor of a dynamical system can be reconstructed from time-delay embeddings of a generic observable. We did not explore this direction here, but it is plausible that introducing delayed embeddings into our framework could further strengthen the identifiability results and provide a complementary perspective to our Gram-based analysis.

Second, while we studied both task-trained and low-rank RNNs in the Appendices, these analyses were intended primarily to support our central results on dRNNs. A more complete theory in these domains remains to be developed. For task-trained networks, this would involve clarifying how structured manifolds, multiple timescales, and mixed selectivity affect the identifiable subspace. For low-rank RNNs, this would mean integrating the latent dynamics implied by the low-rank factorization with the spectral constraints derived from the Gram matrix. Both directions represent natural and important extensions of our work.

Following established practice in the field Das & Fiete (2020); Qian et al. (2024) and for clarity of presentation, our paper is intentionally limited to theory and controlled synthetic experiments. While dRNNs have been applied to real neural recordings many times Perich et al. (2021); Valente et al. (2022), we chose not to pursue such applications here. Beyond the practical issue of dataset access and additional complications associated with (somewhat nonstandard (Rajan et al., 2016; Perich et al., 2021; Valente et al., 2022)) preprocessing of neural activities, we believe little is to be gained scientifically from training one more RNN on these datasets without causal perturbations that can only be performed in experimental settings. Notably, theorems 1 and 2 are best illustrated in simulated datasets where the ground truth is known.

Currently, two key empirical applications of our theory remain practically untested and will likely remain so until single-cell level interventions become mainstream and instant. Testing two of our central ideas, most notably Theorem 3 and the proposed interventions, requires not just observational data but direct empirical evaluations at the level of individual neurons, which may take years to develop Vinograd et al. (2024); Liu et al. (2024). We hope that future work will use our framework to rapidly discard inconsistent hypotheses (*e.g.*, perturbation predictions that result from non-identifiable components) and to design closed-loop intervention experiments that directly test Theorem 3. Such experiments would provide a stringent evaluation of our theory and clarify how identifiability constraints limit inference from real neural recordings.

Finally, we acknowledge the use of large language models for copyediting and grammar corrections, as well as simplification of jargon in several places of our writing.

REFERENCES

AA Al-Falou and D Trummer. Identifiability of recurrent neural networks. *Econometric Theory*, 19 (5):812–828, 2003.

Francesca Albertini and Eduardo D. Sontag. For neural networks, function determines form. *Neural Networks*, 6(7):975–990, 1993. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(09)80007-5. URL https://www.sciencedirect.com/science/article/pii/S0893608009800075.

Francesca Albertini, Eduardo D Sontag, et al. State observability in recurrent neural networks. *Systems & Control Letters*, 22:235–244, 1994.

Bruno B Averbeck, Peter E Latham, and Alexandre Pouget. Neural correlations, population coding and computation. *Nature reviews neuroscience*, 7(5):358–366, 2006.

Manuel Beiran, Alexis Dubreuil, Adrian Valente, Francesca Mastrogiuseppe, and Srdjan Ostojic. Shaping dynamics with multiple populations in low-rank recurrent networks. *Neural Computation*, 33(6):1572–1615, 2021.

Tirthabir Biswas and James E Fitzgerald. Geometric framework to predict structure from function in neural networks. *Physical Review Research*, 4(2):023255, 2022.

Joachim Bona-Pellissier, François Bachoc, and François Malgouyres. Parameter identifiability of a deep feedforward relu neural network. *Machine Learning*, 112(11):4431–4493, 2023.

Hayley A Bounds and Hillel Adesnik. Network influence determines the impact of cortical ensembles on stimulus detection. *bioRxiv*, pp. 2024–08, 2024.

Braden AW Brinkman, Fred Rieke, Eric Shea-Brown, and Michael A Buice. Predicting how and when hidden neurons skew measured synaptic interactions. *PLoS computational biology*, 14(10): e1006490, 2018.

Roger W Brockett. Nonlinear systems and differential geometry. *Proceedings of the IEEE*, 64(1): 61–72, 2005.

Phuong Bui Thi Mai and Christoph Lampert. Functional vs. parametric equivalence of relu networks. In *8th International Conference on Learning Representations*, 2020.

Rosa Cao and Daniel Yamins. Explanatory models in neuroscience, part 1: Taking mechanistic abstraction seriously. *Cognitive Systems Research*, 87:101244, 2024.

Mark M Churchland, John P Cunningham, Matthew T Kaufman, Justin D Foster, Paul Nuyujukian, Stephen I Ryu, and Krishna V Shenoy. Neural population dynamics during reaching. *Nature*, 487 (7405):51–56, 2012.

Zach Cohen, Brian DePasquale, Mikio C Aoi, and Jonathan W Pillow. Recurrent dynamics of prefrontal cortex during context-dependent decision-making. *bioRxiv*, pp. 2020–11, 2020.

PE Crouch. Realisation theory for dynamical systems. In *Proceedings of the Institution of Electrical Engineers*, volume 126, pp. 605–615. IET, 1979.

Kayvon Daie, Karel Svoboda, and Shaul Druckmann. Targeted photostimulation uncovers circuit motifs supporting short-term memory. *Nature Neuroscience*, 24(2):259–265, 2021.

Abhranil Das and Ila R Fiete. Systematic errors in connectivity inferred from activity in strongly recurrent networks. *Nature Neuroscience*, 23(10):1286–1296, 2020.

Fatih Dinc, Adam Shai, Mark Schnitzer, and Hidenori Tanaka. CORNN: Convex optimization of recurrent neural networks for rapid inference of neural dynamics. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=GGIA1p9fDT.

Fatih Dinc, Marta Blanco-Pozo, David Klindt, Francisco Acosta, Yiqi Jiang, Sadegh Ebrahimi, Adam Shai, Hidenori Tanaka, Peng Yuan, Mark J Schnitzer, et al. Latent computing by biological neural networks: A dynamical systems framework. *arXiv preprint arXiv:2502.14337*, 2025.

Laura N Driscoll, Noah L Pettit, Matthias Minderer, Selmaan N Chettih, and Christopher D Harvey. Dynamic reorganization of neuronal activity patterns in parietal cortex. *Cell*, 170(5):986–999, 2017.

Alexis Dubreuil, Adrian Valente, Manuel Beiran, Francesca Mastrogiuseppe, and Srdjan Ostojic. Complementary roles of dimensionality and population structure in neural computations. *biorxiv*, 2020.

Alexis Dubreuil, Adrian Valente, Manuel Beiran, Francesca Mastrogiuseppe, and Srdjan Ostojic. The role of population structure in computations through neural dynamics. *Nature Neuroscience*, pp. 1–12, 2022.

Lea Duncker and Maneesh Sahani. Dynamics on the manifold: Identifying computational dynamical activity from neural population recordings. *Current opinion in neurobiology*, 70:163–170, 2021.

Daniel Durstewitz, Georgia Koppe, and Max Ingo Thurm. Reconstructing computational system dynamics from neural data with recurrent neural networks. *Nature Reviews Neuroscience*, pp. 1–18, 2023.

Sadegh Ebrahimi, Jérôme Lecoq, Oleg Rumyantsev, Tugce Tasci, Yanping Zhang, Cristina Irimia, Jane Li, Surya Ganguli, and Mark J Schnitzer. Emergent reliability in sensory cortical coding and inter-area communication. *Nature*, 605(7911):713–721, 2022.

Arseny Finkelstein, Lorenzo Fontolan, Michael N Economo, Nuo Li, Sandro Romani, and Karel Svoboda. Attractor dynamics gate cortical information flow during decision-making. *Nature Neuroscience*, 24(6):843–850, 2021.

Richard J Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A Baas, Benjamin A Dunn, May-Britt Moser, and Edvard I Moser. Toroidal topology of population activity in grid cells. *Nature*, 602(7895):123–128, 2022.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Niclas Alexander Göring, Florian Hess, Manuel Brenner, Zahra Monfared, and Daniel Durstewitz. Out-of-domain generalization in dynamical systems reconstruction. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=xTYIAD2NND.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Ann Huang, Satpreet H Singh, Flavio Martinelli, and Kanaka Rajan. Measuring and controlling solution degeneracy across task-trained recurrent neural networks. *ArXiv*, pp. arXiv–2410, 2025.

D Kepple, Rainer Engelken, and Kanaka Rajan. Curriculum learning as a tool to uncover learning principles in the brain. In *International Conference on Learning Representations*, 2022.

Mikail Khona and Ila R Fiete. Attractor and integrator networks in the brain. *Nature Reviews Neuroscience*, 23(12):744–766, 2022.

Timothy Doyeon Kim, Thomas Zhihao Luo, Tankut Can, Kamesh Krishnamurthy, Jonathan W Pillow, and Carlos D Brody. Flow-field inference from neural data using deep recurrent networks. *bioRxiv*, 2023.

Tony Hyun Kim and Mark J Schnitzer. Fluorescence imaging of large-scale neural ensemble dynamics. *Cell*, 185(1):9–41, 2022.

Shinichiro Kira, Houman Safaai, Ari S Morcos, Stefano Panzeri, and Christopher D Harvey. A distributed and efficient population code of mixed selectivity neurons for flexible navigation decisions. *Nature communications*, 14(1):2121, 2023.

Bariscan Kurtkaya, Fatih Dinc, Mert Yuksekgonul, Marta Blanco-Pozo, Ege Cirakman, Mark Schnitzer, Yucel Yemez, Hidenori Tanaka, Peng Yuan, and Nina Miolane. Dynamical phases of short-term memory mechanisms in RNNs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ybBuwgOPOd.

Christopher Langdon, Mikhail Genkin, and Tatiana A Engel. A unifying perspective on neural manifolds and circuits for cognition. *Nature Reviews Neuroscience*, pp. 1–15, 2023.

Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian learning and inference in recurrent switching linear dynamical systems. In *Artificial intelligence and statistics*, pp. 914–922. PMLR, 2017.

Mengyu Liu, Aditya Nair, Nestor Coria, Scott W Linderman, and David J Anderson. Encoding of female mating dynamics by a hypothalamic line attractor. *Nature*, pp. 1–3, 2024.

Jason Manley, Sihao Lu, Kevin Barber, Jeffrey Demas, Hyewon Kim, David Meyer, Francisca Martínez Traub, and Alipasha Vaziri. Simultaneous, cortex-wide dynamics of up to 1 million neurons reveal unbounded scaling of dimensionality with neuron number. *Neuron*, 2024.

Valerio Mante, David Sussillo, Krishna V Shenoy, and William T Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *nature*, 503(7474):78–84, 2013.

Francesca Mastrogiuseppe and Srdjan Ostojic. Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron*, 99(3):609–623, 2018.

Aditya Nair, Tomomi Karigo, Bin Yang, Surya Ganguli, Mark J Schnitzer, Scott W Linderman, David J Anderson, and Ann Kennedy. An approximate line attractor in the hypothalamus encodes an aggressive state. *Cell*, 186(1):178–193, 2023.

Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

Matthew G Perich and Kanaka Rajan. Rethinking brain-wide interactions through multi-region 'network of networks' models. *Current opinion in neurobiology*, 65:146–151, 2020.

Matthew G Perich, Charlotte Arlt, Sofia Soares, Megan E Young, Clayton P Mosher, Juri Minxha, Eugene Carter, Ueli Rutishauser, Peter H Rudebeck, Christopher D Harvey, et al. Inferring brain-wide interactions using data-constrained recurrent neural network models. *bioRxiv*, pp. 2020–12, 2021.

Matthew G Perich, Devika Narain, and Juan A Gallego. A neural manifold view of the brain. *Nature Neuroscience*, pp. 1–16, 2025.

Alexandre Pouget, Peter Dayan, and Richard Zemel. Information processing with population codes. *Nature Reviews Neuroscience*, 1(2):125–132, 2000.

Alexander S Poznyak, Edgar N Sanchez, and Wen Yu. *Differential neural networks for robust nonlinear control: identification, state estimation and trajectory tracking*. World Scientific, 2001.

William Qian, Jacob Zavatone-Veth, Ben Ruben, and Cengiz Pehlevan. Partial observation can induce mechanistic mismatches in data-constrained models of neural dynamics. *Advances in Neural Information Processing Systems*, 37:67467–67510, 2024.

Kanaka Rajan, LF Abbott, and Haim Sompolinsky. Stimulus-dependent suppression of chaos in recurrent neural networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 82(1):011903, 2010.

Kanaka Rajan, Christopher D Harvey, and David W Tank. Recurrent network models of sequence generation and memory. *Neuron*, 90(1):128–142, 2016.

Oleg I Rumyantsev, Jérôme A Lecoq, Oscar Hernandez, Yanping Zhang, Joan Savall, Radosław Chrapkiewicz, Jane Li, Hongkui Zeng, Surya Ganguli, and Mark J Schnitzer. Fundamental bounds on the fidelity of sensory cortical coding. *Nature*, 580(7801):100–105, 2020.

Shreya Saxena and John P Cunningham. Towards the neural population doctrine. *Current opinion in neurobiology*, 55:103–111, 2019.

Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, pp. 1–9, 2023.

Friedrich Schuessler, Alexis Dubreuil, Francesca Mastrogiuseppe, Srdjan Ostojic, and Omri Barak. Dynamics of random recurrent networks with correlated low-rank structure. *Physical Review Research*, 2(1):013111, 2020.

Eduardo D Sontag. *Mathematical control theory: deterministic finite dimensional systems*, volume 6. Springer Science & Business Media, 2013.

Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Matteo Carandini, and Kenneth D Harris. High-dimensional geometry of population responses in visual cortex. *Nature*, 571(7765):361–365, 2019.

David Sussillo and Larry F Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009.

David Sussillo and Omri Barak. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural computation*, 25(3):626–649, 2013.

Hector J Sussmann. Existence and uniqueness of minimal realizations of nonlinear systems. *Mathematical systems theory*, 10(1):263–284, 1976.

Héctor J. Sussmann. Uniqueness of the weights for minimal feedforward nets with a given input-output map. *Neural Networks*, 5(4):589–593, 1992. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(05)80037-1. URL https://www.sciencedirect.com/science/article/pii/S0893608005800371.

Floris Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*, pp. 366–381. Springer, 2006.

Adrian Valente, Jonathan W Pillow, and Srdjan Ostojic. Extracting computational mechanisms from neural data using low-rank rnns. *Advances in Neural Information Processing Systems*, 35:24072–24086, 2022.

Amit Vinograd, Aditya Nair, Joseph Kim, Scott W Linderman, and David J Anderson. Causal evidence of a line attractor encoding an affective state. *Nature*, pp. 1–3, 2024.

Saurabh Vyas, Matthew D Golub, David Sussillo, and Krishna V Shenoy. Computation through neural population dynamics. *Annual review of neuroscience*, 43(1):249–275, 2020.

Edgar Y Walker, Fabian H Sinz, Erick Cobos, Taliah Muhammad, Emmanouil Froudarakis, Paul G Fahey, Alexander S Ecker, Jacob Reimer, Xaq Pitkow, and Andreas S Tolias. Inception loops discover what excites neurons most using deep predictive models. *Nature neuroscience*, 22(12):2060–2065, 2019.

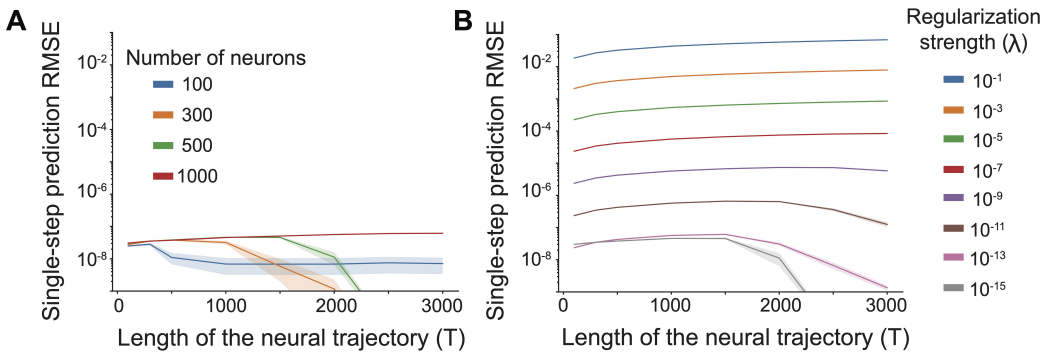

Figure S1: **Single-step prediction root-mean-squared errors corresponding to Figure 2. A** Effect of the number of observed neurons on single-step prediction RMSE across trajectory lengths T with negligible regularization ($\lambda = 10^{-15}$). **B** Effect of regularization strength λ on single-step prediction RMSE across trajectory lengths T. These results complement Figure 2 by showing the direct single-step prediction errors for RNNs trained to reproduce neural trajectories sampled from chaotic RNNs. A reasonable tolerance level for the single-step prediction RMSE is $O(10^{-8})$, since squared error is minimized during training and machine precision is $\sim 10^{-16}$. Parameters: For **A**, same as in Fig. 1**A-B** but with varying N. For **B**, same as in Fig. 1**C** but with additional λ values.

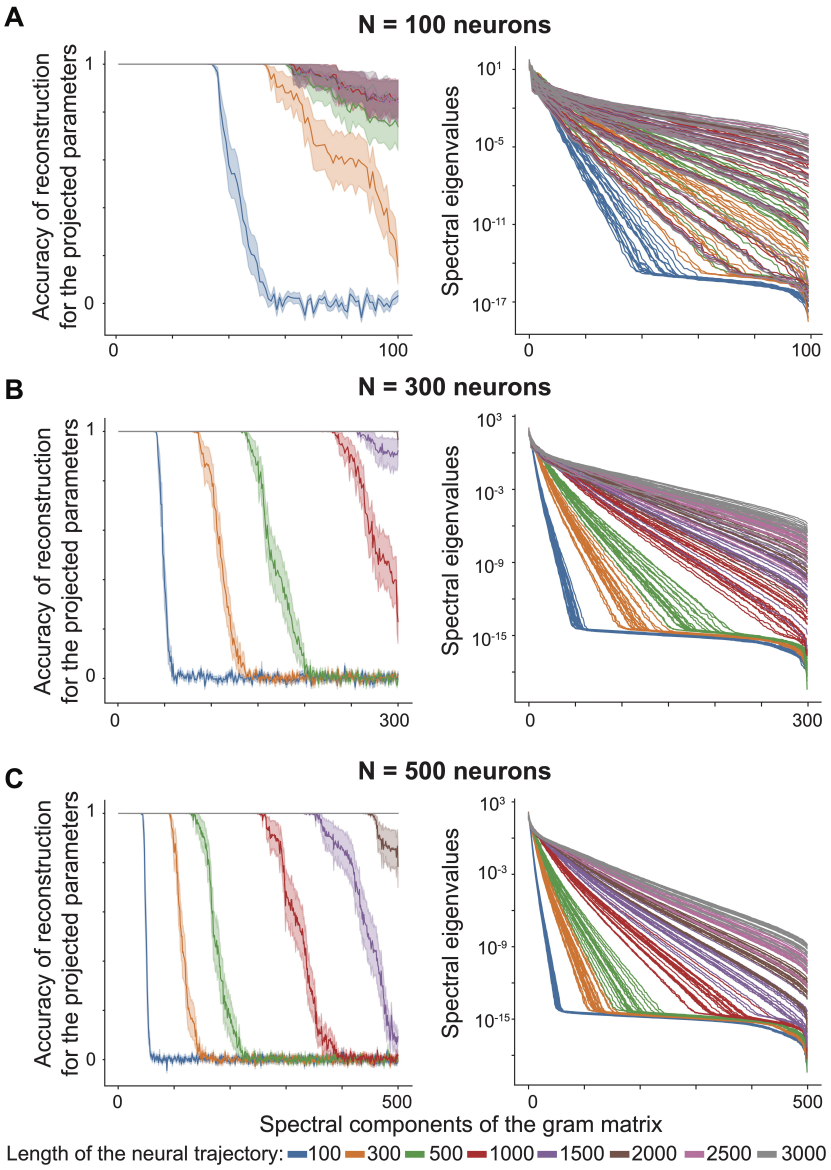

Figure S2: **Empirical verification of the neural uncertainty principle with networks of varying sizes.** We performed the same analysis as in Fig. 2**A-B**, but for RNNs that had $N = 100$ (**A**), $N = 300$ (**B**), and $N = 500$ (**C**) neurons.

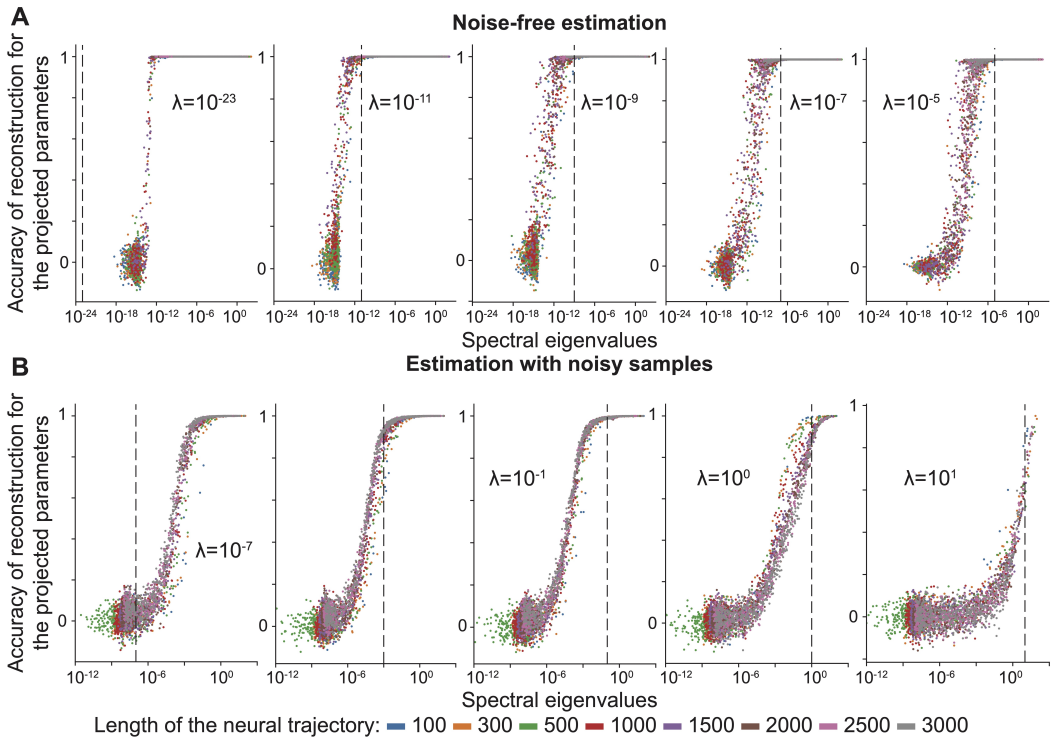

Figure S3: **Regularization levels control which spectral components are used for parameter estimation.** We performed the same analysis as in Fig. 2**C** for varying λ values for noiseless (**A**) and noisy (**B**) evolutions of the RNN. For the noisy case, we picked $\epsilon_{\text{in}} \sim \mathcal{N}(0, 10^{-6})$ and $\epsilon_{\text{conv}} \sim \text{Laplace}(10^{-3})$, in which x in Laplace(x) refers to the scale parameter.

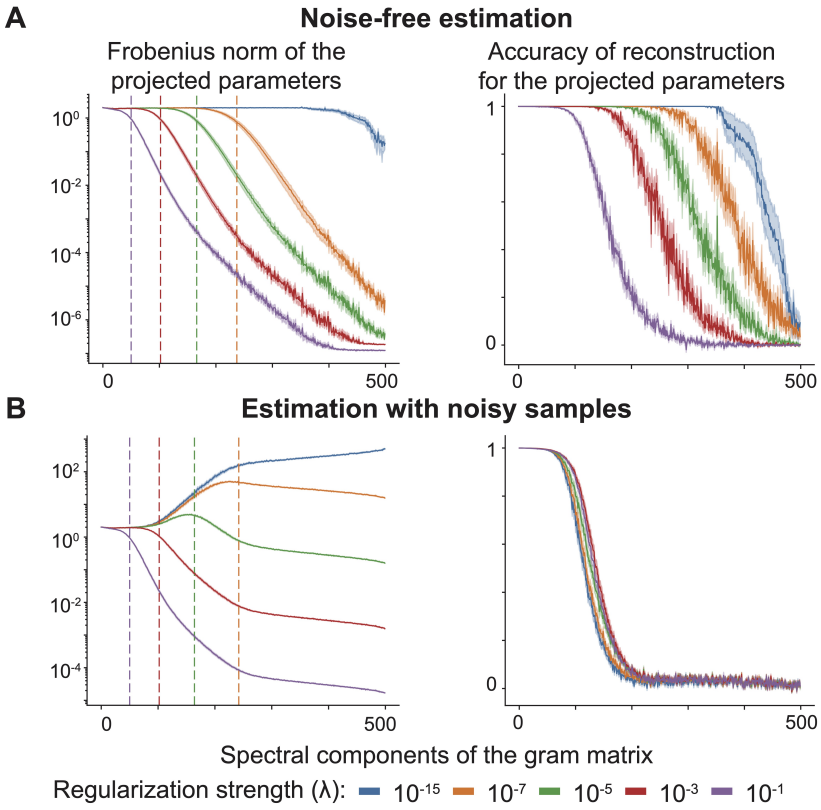

Figure S4: **When estimation is not regularized, noise inflates the non-identifiable components.** Similar to Fig. S3 (using the same parameters), we examined the Frobenius norm of parameter components projected onto the spectral components of the Gram matrix (*left*) and the reconstruction accuracy measured as the correlations between ground truth and predicted projections (*right*) across varying λ values for noiseless (**A**) and noisy (**B**) RNN evolutions. Without regularization, noise caused systematic overestimation of magnitude in non-identifiable components, which should ideally have been chosen with zero norm.

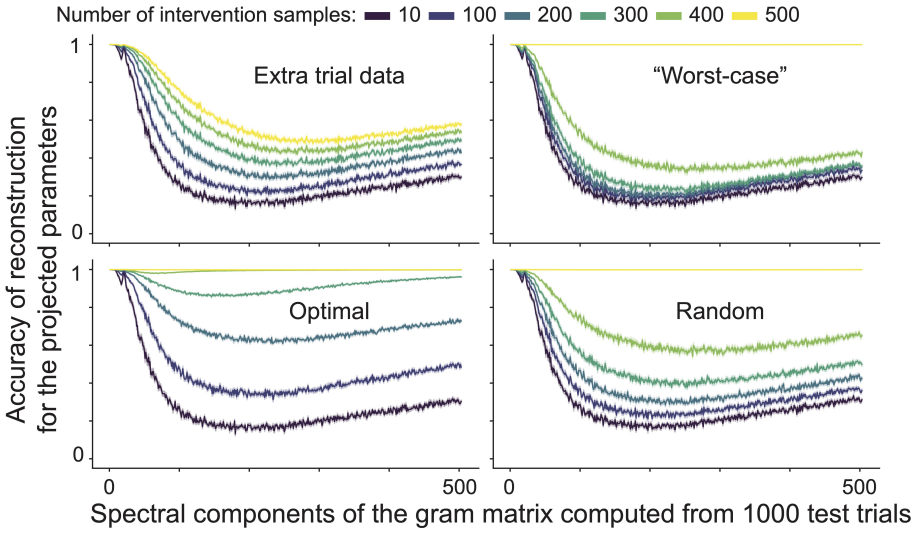

Figure S5: **Revealing all RNN parameters is distinct from revealing those that encode task-relevant computations, the latter predominantly reside in the top spectral components.** We reanalyzed the RNNs trained in Fig. 4 by computing a generalized Gram matrix from 1000 trials of each network performing the 3-bit flip-flop task. Using its spectral components, we evaluated reconstruction accuracies of the projected parameter dimensions as a function of the number of intervention samples used during reconstruction. Compared with Fig. 4**C** (where ~ 100 intervention samples led to high output accuracies for the optimal strategy; and $\sim$ 200 for the random and "extra-trial-data" strategies), RNNs that reliably solved the 3-bit flip-flop task exhibited substantial variability in their higher spectral components relative to the generator RNNs. As expected from Theorem 1, all three Gram-matrix–based strategies (*i.e.*, all except "extra-trial-data") eventually achieved uniform perfect reconstruction.

A Extended Related Works

In the main text, we summarized prior work on RNNs as models of neural activity, as well as the general identifiability of RNNs and nonlinear systems.

Dynamical models of neural activity. A central premise of computational neuroscience is that computational models that reproduce neural activity will provide biological insight. However, recovering synaptic connectivity or precise mechanisms from dynamics alone is generally ill-posed (Das & Fiete, 2020; Brinkman et al., 2018), and even functional properties such as attractors can be unreliable when inferred from observational data alone (Qian et al., 2024; Göring et al., 2024).

Despite these limitations, predictive models have generated potentially meaningful results. RNNs trained on neural trajectories have been shown to capture features such as population-level gating (Finkelstein et al., 2021), inter-area communication motifs (Perich & Rajan, 2020), and low-dimensional attractor dynamics (Valente et al., 2022). Several of these predictions have been refined and confirmed through causal perturbations (Daie et al., 2021; Liu et al., 2024; Vinograd et al., 2024; Walker et al., 2019), demonstrating that data-driven models can sometimes generate testable mechanistic hypotheses. In an effort to preserve biological interpretability, some studies have trained "data-constrained" RNNs with a one-to-one mapping between model units and recorded neurons (Perich & Rajan, 2020; Perich et al., 2021; Dinc et al., 2023). This approach aims to avoid confounds introduced by hidden units and to estimate functional connectivity directly. However, even in these restricted settings, little has been studied about the identifiability of parameters, leaving open the question of whether different underlying models can equally explain the same data.

Identifiability in nonlinear systems. The broader control and systems literature provides a foundation for understanding when models can be uniquely determined from observed behavior. Classical realization theory shows that any finite-dimensional system's external behavior can be represented by a minimal, unique system if it is both controllable and observable (Sussmann, 1976). In this framework, two systems are indistinguishable if they generate the same outputs for all inputs, and minimal models are live in the quotienting the parameter space of the original model with this equivalence relation.

Complementary results come from dynamical systems theory. Takens' embedding theorem (Takens, 2006) guarantees that, given a sufficiently large embedding dimension, the dynamics of a system can be reconstructed from time-delayed measurements of even a single observable (Schmid, 2010). This provides theoretical justification for reconstructing dynamics from partial observations, as is common in neuroscience. Yet in practice, neural data often violate these assumptions. Activations are highly redundant and typically lie in a low-dimensional subspace (Dubreuil et al., 2020; Perich et al., 2025), undermining identifiability.

Identifiability in neural networks. Identifiability in neural networks has been studied for decades, though usually under restrictive assumptions. For recurrent architectures with linear or smooth nonlinear activations (such as tanh), input–output mappings can constrain parameters up to permutation symmetries, except in degenerate situations caused by dependencies, nonobservability, or noncontrollability (Sussmann, 1992; Poznyak et al., 2001; Albertini & Sontag, 1993; Albertini et al., 1994; Sontag, 2013). Real neural data, however, are precisely such degenerate cases: redundancy and low dimensionality leave entire parameter directions unconstrained.

Recent work has formalized these issues in both recurrent and feedforward networks. For example, distinct connectivity matrices in piecewise-linear RNNs can produce identical steady states (Biswas & Fitzgerald, 2022), and equivalence classes of minimal, identifiable systems have been defined for restricted classes of RNNs (Al-Falou & Trummer, 2003). Parallel efforts have analyzed parameter symmetries in feedforward networks, especially with ReLU nonlinearities (Bui Thi Mai & Lampert, 2020; Bona-Pellissier et al., 2023).

Solution degeneracy in task-trained RNNs. Within neuroscience and machine learning, non-identifiability is often discussed under the broader notion of solution degeneracy. Input-driven constraints shape RNN dynamics, but leave ambiguity (Rajan et al., 2010), and partial observability creates challenges for learning and inference (Kepple et al., 2022). More recently, Huang et al. (2025) introduced a framework to quantify and control solution degeneracy in task-trained RNNs,

showing that variability across solutions depends on model capacity and task complexity. Their results highlight the need for interventions to disambiguate latent mechanisms, as multiple parameterizations can fit the same task, using different mechanisms.

B Additional results and ablation studies

B.1 Estimation with a broad class of estimators

Theorem 2 and Corollary 1 jointly propose a clear path for designing estimators, *i.e.*, those that respect an update rule in Eq. 7, and/or a class of loss functions, *i.e.*, those that follow Eq. 8. In this section, we focus on one group of estimators inspired by Eq. 8, for which parameters at a local minimum is guaranteed to be reconstructed in a manner consistent with Theorem 1. Specifically, for a given set of auxiliary ($r[s]$) and target ($r[s+1]$) activities, we first define an equivalent target:

$$d_i[s] = \frac{r_i[s+1] - (1-\alpha) r_i[s]}{\alpha}. \tag{S1}$$

Notably, this is also an observable once $r[s]$ and $r[s+1]$ are known and can be estimated via:

$$\hat{d}[s] = \tanh(\theta^T x[s]), \tag{S2}$$

where $x[s] = [r[s], u[s]] \in \mathbb{R}^{N+N_{\text{in}}}$ and the parameter matrix $\theta = [W^{\text{rec}}, W^{\text{in}}] \in \mathbb{R}^{N \times (N+N_{\text{in}})}$ defined as before. Then, the loss function in Eq. 8 becomes:

$$\mathcal{L}(\theta) = \sum_{s=1}^{TM} \mathcal{L}(d_i[s], \hat{d}_i[s]), \tag{S3}$$

in which the loss can be chosen as a standard ℓ_2 loss, or cross entropy, or a weighted cross entropy as in Dinc et al. (2023).

Building on this construction, we now test how different estimators behave in practice when trained on finite, noisy datasets. Figure S6 compares four approaches: CORNN, second-order cross-entropy minimization, first-order cross-entropy (Adam), and a standard ℓ_2 loss. Each method minimizes

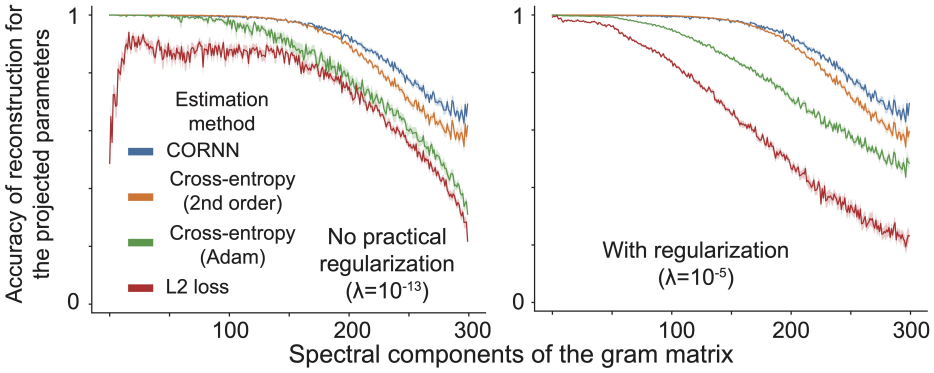

Figure S6: **Estimators consistent with Theorem 2 can employ distinct loss functions and optimization strategies, yet still agree with Theorem 1 near their minima.** We reconstructed parameters of RNNs ($N = 300$) trained on a delayed match-to-sample (DMTS) task using four distinct optimization strategies, each minimizing single-step prediction error. *Left:* Estimators with negligible regularization ($\lambda = 10^{-13}$). All converged except the first-order minimization of an ℓ_2 loss, which is non-convex and likely had not converged near a local minimum. *Right:* With proper regularization ($\lambda = 10^{-5}$), all estimators produced reconstruction accuracies that decreased systematically along the spectral components of the Gram matrix G computed from the training samples. Parameters: $T_{\text{in}} = 30$ ms, $T_{\text{delay}} = 80$ ms, $T_{\text{resp}} = 50$ ms, $\Delta t = 5$ ms, and $\alpha = 0.5$. RNNs were injected with random noise at every time step with $\epsilon_{\text{in}} \sim \mathcal{N}(0, 10^{-4})$ and $\epsilon_{\text{conv}} \sim 0.1\,\text{Poisson}(10^{-3})$. Training samples included $B = 40$ trials, each with 38 data points, totaling $T = 1520$.

single-step prediction error, hence guaranteed by Corollary 1 to follow the reconstruction pattern as stated by Theorem 1, but differs in optimization and loss formulation. To do so, we focus on generator RNNs that are trained to perform a canonical working memory task, *i.e*, delayed match-to-sample (DMTS) (Fig. S6). In this task, the network receives a brief input cue, maintains it over a variable delay, and generates the corresponding output upon the go signal.

When dRNNs were trained with negligible regularization (Fig. S6, *left*; $\lambda \approx 10^{-13}$), all estimators but the one minimizing the ℓ_2 loss produced reconstructions aligned with the theoretical predictions of Theorem 1. CORNN and both cross-entropy variants, all of which are convex loss functions, track the leading spectral components of the Gram matrix with near-perfect accuracy. By contrast, the first-order ℓ_2 loss, being non-convex, fails to converge near a local minimum described by Corollary 1. This is expected, as proper regularization is often needed for a first-order estimation to settle into a local minimum with a non-convex loss function. This was indeed the case with regularization (Fig. S6, *right*; $\lambda = 10^{-5}$), for which all estimators collapse onto the same characteristic behavior: reconstruction accuracy decays systematically with spectral index as regularization suppresses the low spectral components following Corollary 1. Importantly, while the specific loss function and optimizer differ, their asymptotic solutions agree at the top spectral components of the parameters.

Taken together, these results demonstrate that dRNNs can employ distinct loss functions and optimization strategies yet still yield reconstructions that agree within the identifiable subspace. In other words, the precise choice of estimator matters less than whether its updates conform to the structure prescribed by Theorem 2, or its loss function follows the form given in Corollary 1 (albeit, optimization with non-convex loss functions may not need to converge).

B.2 Practical estimation of the Gram matrix with nontrivial noise models

The practical applicability of Theorem 1, as well as the premise of Theorem 2, both rely on the assumption that Eq. 6 holds, *i.e.*, noise is zero-mean and independently sampled across neurons and time points. What if either was not the case, which is expected to occur in practice, what hope do we have in reconstructing identifiable parameters in dRNNs?

Independent noise with finite mean. We already considered this case empirically in Fig. S6, in which Poisson distribution introduced finite mean to the conversion noise. Yet, the parameter reconstruction roughly follows Theorem 1. Here, we theoretically argue why finite bias introduced by independent noise is expected to have minor, if any, perturbation to Eq. 6. First, define $a = E[\epsilon]$, then we can write:

$$\begin{aligned}\mathbb{E}[G] = \mathbb{E}[X^T X] &= \mathbb{E}\left[(\tilde{X}+\epsilon)^T(\tilde{X}+\epsilon)\right] = \mathbb{E}\left[\tilde{X}^T\tilde{X} + \epsilon^T\tilde{X} + \tilde{X}^T\epsilon + \epsilon^2\right], \\ &= \mathbb{E}[\tilde{G}] + TM(\sigma_\epsilon^2 + aa^T) + \tilde{X}^T\mathbf{1}a^T + a\mathbf{1}^T\tilde{X},\end{aligned} \tag{S4}$$

where $\mathbf{1}$ is a vector of all 1s. Notably, this is at most a rank-3 correction to the original Eq. 6, and a diagonal plus low-rank correction to $\tilde{G}$. Hence, while noise with finite mean can hurt the estimation of the Gram matrix, this happens at most with a low-rank correction.

Low-rank spatiotemporally correlated noise. In the most general case, we can write the empirical Gram matrix as

$$\mathbb{E}[G] = \mathbb{E}[X^T X] = \mathbb{E}\left[(\tilde{X}+\epsilon)^T(\tilde{X}+\epsilon)\right] = \mathbb{E}[\tilde{G}] + \Sigma_\epsilon + \mathbb{E}\left[\epsilon^T\tilde{X} + \tilde{X}^T\epsilon\right]. \tag{S5}$$

Here, the first term $\tilde{G} = \tilde{X}^T\tilde{X}$ is the noiseless Gram matrix, the second term accounts for the intrinsic spatiotemporal covariance of the noise, while the remaining terms capture possible correlations between $\tilde{X}$ and ϵ. Unlike the case with a random noise, here, there is no guarantee that any of the corrections are diagonal and/or low-rank. On the other hand, earlier work has hypothesized (supported with empirical evidence Ebrahimi et al. (2022); Rumyantsev et al. (2020)) that noise correlations may actually be low-rank Dinc et al. (2025), which is what we consider next.

Specifically, suppose the noise $\epsilon \in \mathbb{R}^{T\times N}$ is confined to an r-dimensional subspace of $\mathbb{R}^N$, so that we can write $\epsilon = ZU^T$ with $U \in \mathbb{R}^{N\times r}$ and $Z \in \mathbb{R}^{T\times r}$. Then the correction takes the form

$$\mathbb{E}[G] = \mathbb{E}[\tilde{G}] + \mathbb{E}[U[Z^TZ]U^T] + \mathbb{E}[UZ^T\tilde{X} + \tilde{X}^TZU^T]. \tag{S6}$$

The first correction term has rank at most r, while the cross-terms add at most another $2r$ directions. Hence, the corrections are bounded by the total number of dimensions that noise is confined to.

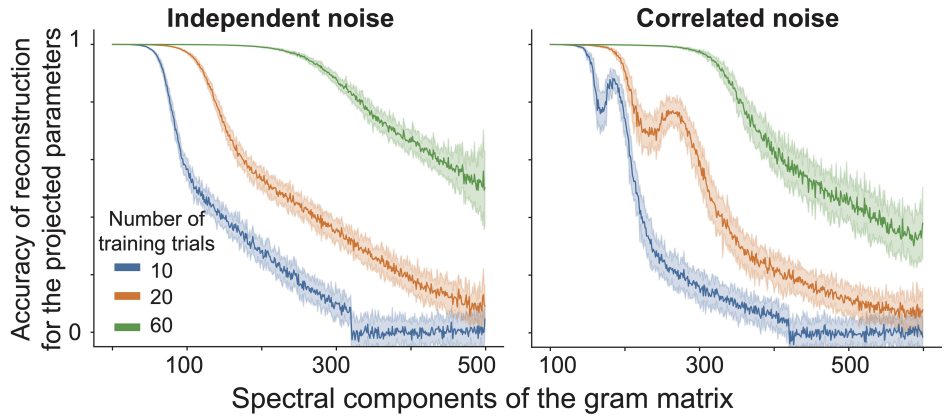

Figure S7: **Spatiotemporally correlated noise induces structure that complicates estimation, but can be mitigated with increased number of trials.** We trained generator RNNs ($N = 500$) on the delayed cue discrimination task with injected random, spatiotemporally independent (*left*) vs correlated (*right*) noise. Estimation accuracies (Pearson's r) of the projected parameters between dRNNs and generator RNNs were plotted as a function of the spectral components of the Gram matrix used during dRNN training. While training with spatiotemporally correlated noise required more trials to be accurate, it converged to the structure predicted by Theorem 1 in the end. Parameters: $T_{\text{in}} = 30$ ms, $T_{\text{delay}} = 80$ ms, $T_{\text{resp}} = 50$ ms, $\Delta t = 10$ ms, and $\alpha = 0.5$. RNNs were injected with (random or correlated, see Appendix D.2.3 for the noise generation process) noise at every time step with $\epsilon_{\text{in}} \sim \mathcal{N}(0, 10^{-4})$ and $\epsilon_{\text{conv}} \sim 0.1\,\text{Poisson}(10^{-3})$. Each training trial contained 32 data points. dRNNs were trained ($\lambda = 10^{-7}$) with varying numbers of observed neurons and trials, as indicated in the figure legends.

Notably, while confinement of noise to a subspace may impede with the signal, there is empirical evidence that in biological brains, noise is largely orthogonal to the signaling directions Rumyantsev et al. (2020).

Empirical tests with spatiotemporally correlated noise. We next asked how realistic (full-rank) spatiotemporal correlations empirically affect estimation (Fig. S7). We focused on generator RNNs trained to perform a delayed cue discrimination task, in which the network receives a brief input cue out of two options, maintains it over a variable delay, and generates the corresponding output in the appropriate response channel. To introduce correlated noise, we first sampled independent Gaussian noise at every neuron and time step, then applied a two-dimensional low-pass filter (convolution with a Gaussian kernel) across neurons and time, and finally rescaled the variance to a fixed target (see Appendix D.2.3 for details). Because the noise was injected directly into the dynamics rather than into an observation process, the resulting neural activities were effectively correlated with the injected noise.

When noise was independent, estimation accuracies decayed systematically with the spectral components of the Gram matrix, and increasing the number of training trials improved recovery as expected (Fig. S7, *left*). By contrast, spatiotemporally correlated noise introduced additional low-dimensional structure that complicated estimation and reduced accuracy at small sample sizes (Fig. S7, *right*). Nevertheless, increasing the number of trials progressively mitigated these effects, and the reconstructions eventually converged to the structure predicted by Theorem 1. This shows that while correlated noise complicates estimation, its impact can be overcome with sufficient data. This finding is consistent with earlier literature, *i.e.*, effective estimation under correlated noise requires larger sample sizes (Dinc et al., 2023), but generalizes it by stating that increased data size and richness enables empirical parameter identification in line with Theorem 1.

Effects of mismatches in the estimated time-scales. Another important, empirically relevant aspect of dRNN training is the potentially incorrect estimation of the timescale parameter τ. Notably, in many applications τ is set by the kernel of the smoothing performed on the neural datasets Perich et al. (2021), yet it is instructive to study the effects of such mismatches. Since the Gram matrix is computed directly from the observed neural activities, a potential mismatch in the assumed time

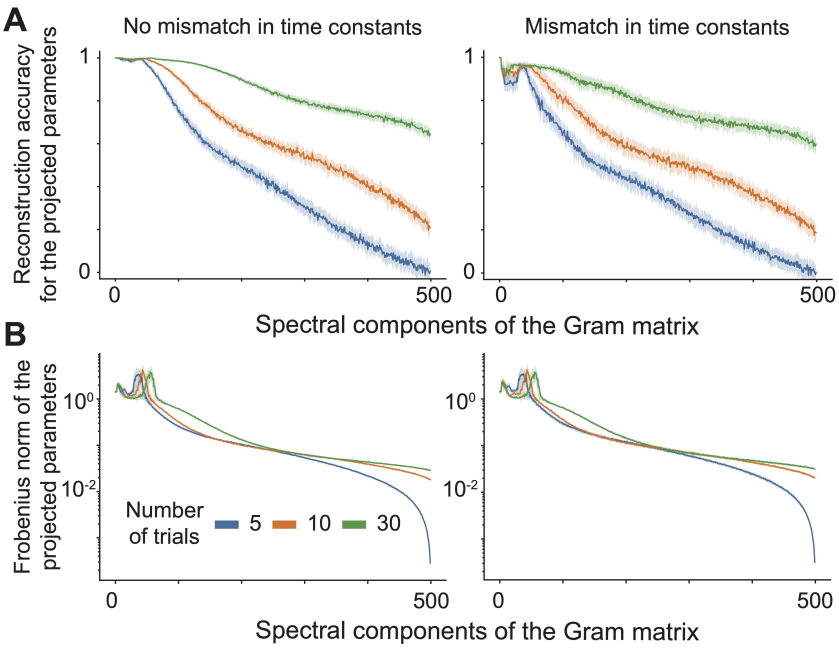

Figure S8: **Even with mismatched time constants, dRNN training tracks the spectrum of the Gram matrix.** We trained dRNNs to reproduce RNNs performing the 3-bit flip-flop task from Fig. 4. In the mismatch condition, time constants were sampled as $\alpha \sim \mathcal{N}(0.5, 0.05^2)$ instead of being fixed at $\alpha = 0.5$. **A** Reconstruction accuracy vs. spectral components of the Gram matrix. **B** Parameter norms vs. spectral components. Parameters: $N = 500$, $\lambda = 10^{-2}$, $\epsilon_{\text{in}} \sim \mathcal{N}(0, 10^{-6})$, $\epsilon_{\text{conv}} \sim 0.1\text{Poisson}(10^{-1})$. Each trial contained 100 data points.

constants of dRNNs does not affect our ability to estimate the spectral components, although it can introduce systematic biases into the parameter estimates Dinc et al. (2023).

To test this, we trained dRNNs to reproduce RNNs performing the 3-bit flip-flop task (Fig. 4), either with matched time constants (fixed at $\alpha = 0.5$) or with mismatched ones, where α was drawn from a distribution $\mathcal{N}(0.5, 0.05^2)$ at each step. Figure S8**A** shows reconstruction accuracies as a function of the spectral components of the Gram matrix. In both conditions, estimation accuracies decayed systematically with spectral index, and increasing the number of training trials improved recovery as expected. Even with mismatched time constants, the dRNNs continued roughly to track the Gram spectrum in line with Theorem 1, though some perturbations similar to Fig. S7 did occur. Notably, the Frobenius norm of the projected parameters did not change substantially between two cases (Figure S8**B**), revealing that mismatches did not particularly bias the norms of the estimated parameters the way that random noise did, *e.g.*, in Fig. S4.

Together, these results demonstrate that mismatched time constants do not compromise identifiability of the spectral components themselves, but they do introduce systematic biases in parameter recovery.

Effects of unobserved influences on the identifiable estimation of dRNN parameters. A major challenge in training dRNNs is the presence of unobserved neurons, which is either omitted in practice Das & Fiete (2020) or used to argue caution against their use Qian et al. (2024). For a given RNN, the dynamics of the observed neurons can be written as

$$\tau \dot{r}(t) = -r(t) + \tanh(W^{\text{rec}} r(t) + W^{\text{in}} u(t) + i(t) + \epsilon_{\text{in}}) + \epsilon_{\text{conv}}, \tag{S7}$$

where $i(t)$ denotes the influence of unobserved neural activities. This influence can be viewed as spatiotemporally correlated noise, but unlike random fluctuations, its structure is often highly aligned with the signal itself. This raises the question of whether parameter estimation remains possible at all, even under the guarantees of Theorem 1.

To formalize this, we extend Theorem 1 to partially observed populations:

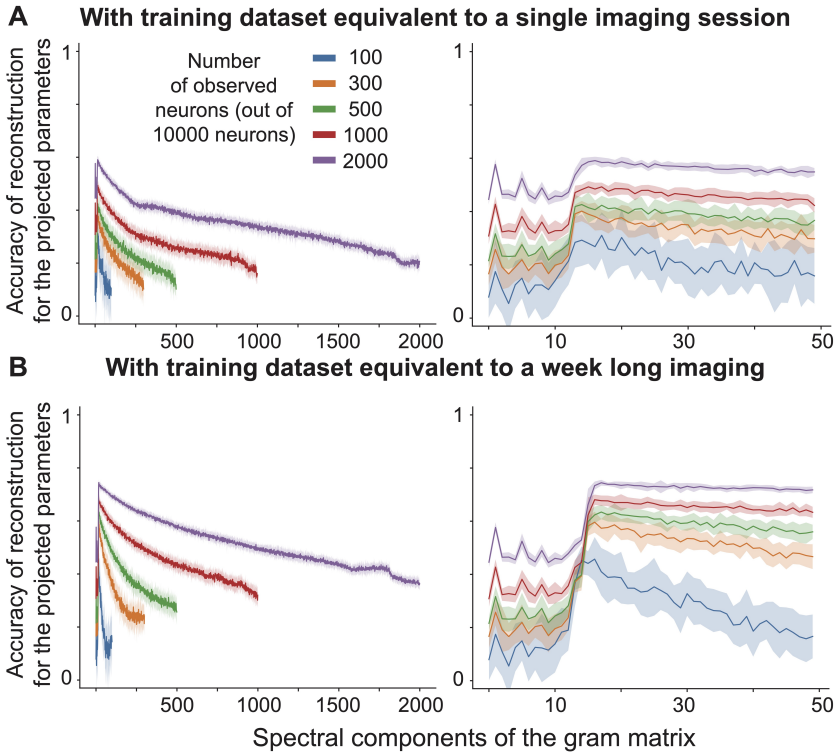

Figure S9: **Partial observations induce unobserved influences that bias the top spectral components of the estimated Gram matrix.** We trained large-scale generator RNNs with $N = 10,000$ neurons on the delayed cue discrimination task from Fig. S7 and performed inference using only partially observed neural populations. Estimation accuracies (Pearson's r) of the projected parameters between dRNNs and generator RNNs were plotted as a function of the spectral components of the Gram matrix used during dRNN training. The right panel provides a close-up of the left. As the number of observed neurons increased, the top $\sim 10 - 15$ spectral components, initially non-identifiable, became identifiable again. Parameters: Same as in Fig. S7, but with $N = 10000$ and only independent noise injections. For **A**, we used 300 trials, comparable to a single imaging session, whereas for **B**, we used 2000 trials, comparable to a week long dataset Ebrahimi et al. (2022).

Proposition S1 (Non-identifiability under partial observation)**.** *Let $x[i] = [x_{\text{obs}}[i], x_{\text{unobs}}[i]]$, where $x_{\text{obs}}[i]$ denotes the observed neural activities and inputs for $i = 1, \ldots, T$, and $x_{\text{unobs}}[i]$ the corresponding unobserved variables. Let θ^*_{obs} denote the parameters among observed neurons, and θ^*_{unobs} those involving unobserved neurons. Let $\phi(\cdot)$ be any nonlinearity, not necessarily continuous. Then, when $\theta_{\text{unobs}} = \theta^*_{\text{unobs}}$, all RNNs with parameters θ_{obs} produce the same observed neural trajectory provided that*

$$\theta_{\text{obs}} = \tilde{\theta}_{\text{obs}} + \Delta\theta_{\text{obs}}, \quad \textit{where} \quad \tilde{\theta}_{\text{obs}} = \theta^*_{\text{obs}} P, \quad \textit{and} \quad \Delta\theta_{\text{obs}} P = 0, \tag{S8}$$

with P denoting the projection operator onto the observed subspace of the neural dataset.

This result highlights that when there are unobserved variables, the converse of Theorem 1 fails to hold. In particular, even if $P = I$, RNN parameters may remain non-identifiable due to the hidden influence of unobserved neurons or the redundancy introduced by non-monotonic activation functions.

Empirical effects of partial observations on identifiability under common influence factors. A realistic assumption, supported by recent theories of neural computation Dinc et al. (2025) and empirical work in task-trained low-rank RNNs Valente et al. (2022); Beiran et al. (2021); Mastrogiuseppe & Ostojic (2018); Schuessler et al. (2020); Dubreuil et al. (2022), is that the same latent variables underlie the dynamics of both observed and unobserved neurons. If this is the case, then the top spectral components of the Gram matrix are presumably dominated by the projections of

latent computations and may become secluded and biased by the missing activity. In contrast, the lower spectral components may still be reconstructed. Figure S9 illustrates this effect by training large-scale generator RNNs ($N = 10{,}000$) on the delayed cue discrimination task. From these, we observed only a subset of the neurons, whose activities were reconstructed with dRNNs. When the training dataset was limited to the scale of a single imaging session (Fig. S9A), the top spectral components could not be recovered reliably under $\sim 1\%$ subsampling, but recovery improved rapidly once $\sim 10\%$ of neurons were observed. When more data were collected, equivalent to a week-long dataset (Fig. S9B), estimation accuracies improved considerably in the lower spectral components, similar to the case of spatiotemporally correlated noise in Fig. S7, but we observed little to no improvement in the top components.

Taken together, these results show that partial observations introduce biases that compromise identifiability of the leading components at finite sample sizes, but increasing the number of samples and recording from more neurons progressively mitigates these effects and restores accurate reconstruction of the broader structure of the dynamics. Importantly, this is not a trivial restatement of known limitations, as the lower spectral components are most important to capture to predict nontrivial dynamics with dRNNs (Theorem 3). Moreover, with the latest imaging technologies Kim & Schnitzer (2022); Manley et al. (2024), such sampling fractions are now feasible, and the number of trials used here are well within experimental reach, with month-long recordings already demonstrated Driscoll et al. (2017). Notably, these tools will only get improved over time Kim & Schnitzer (2022). Overall, our finding that only a fraction of the full neuronal population may be sufficient to counteract the effects of non-identifiable components helps explain the latest successes of dynamical system models in predicting causal perturbations Vinograd et al. (2024); Liu et al. (2024), and should motivate rather than discourage experimentalists to use dRNN models as tools for extracting new insights into neural algorithms from large-scale brain recordings.

B.3 Extensions of the identifiability theory to low-rank RNNs

Non-identifiability in low-rank RNN parameters. A first intuition might be that the intrinsically low-rank nature of the identifiable subspace, as predicted by Theorem 1, could be enforced directly by training low-rank RNNs constrained on neural trajectories Valente et al. (2022). In this approach, the recurrent weight matrix is factorized as $W^{\mathrm{rec}} = CD$ with $C \in \mathbb{R}^{N\times K}$, $D \in \mathbb{R}^{K\times N}$, and $K \ll N$, so that only $O(KN)$ parameters are learned. This parametrization appears to align with the expectation that only a low-rank subset of parameters is identifiable. However, recent theoretical results reveal a crucial complication: even rank-one RNNs can generate neural trajectories that span the full N-dimensional space of activities Dinc et al. (2025). In that sense, enforcing low-rank structure on W^{rec} does not guarantee that the observed activity itself is low-dimensional, nor should it reduce the identifiability requirements of the system. To fully constrain the parameters, the observation conditions remain just as strict as in the full-rank case. To resolve the apparent contradiction between these two intuitions, we state and prove the following theorem:

Proposition S2 (Non-identifiability in low-rank RNNs)**.** *Consider the noiseless RNN in Eq. 2 with $\epsilon_{\mathrm{in/conv}}(t) = 0$. Suppose the recurrent weights are parameterized as $W^{\mathrm{rec}} = CD$ with $C \in \mathbb{R}^{N\times K}$, $D \in \mathbb{R}^{K\times N}$, and $K \ll N$. Let X be the observation matrix with the projection operator P onto its column space. Then any parameterization of the form $W^{\mathrm{rec}} = C(D + \Delta D)$ with $\Delta DP = 0$ produces the same neural dataset $\mathcal{D}$ while preserving $\mathrm{rank}(W^{\mathrm{rec}}) \leq K$.*

In essence, Proposition S2 shows that low-rank parameterizations do not resolve the fundamental ambiguity: perturbations of the form $C(D + \Delta D)$ with $\Delta DP = 0$ leave the dataset unchanged while preserving the network rank.

Low-rank regularization does not necessarily mitigate incorrect estimation, weight regularization does. Combining Theorems 2 and 3 with our results in Figs. S7 and S9, it is reasonable to expect that introducing ℓ_2 penalty on the weights (though, only when the estimator is correctly structured, see Fig. 3 for a counterexample) improves the estimation of underlying dynamics in dRNNs. As a (commonly argued Valente et al. (2022)) alternative, since the number of free parameters significantly decreases in the low-rank setting, we next asked whether constraining dRNNs to a low-rank parameterization would improve recovery of the underlying dynamics under partial observation.

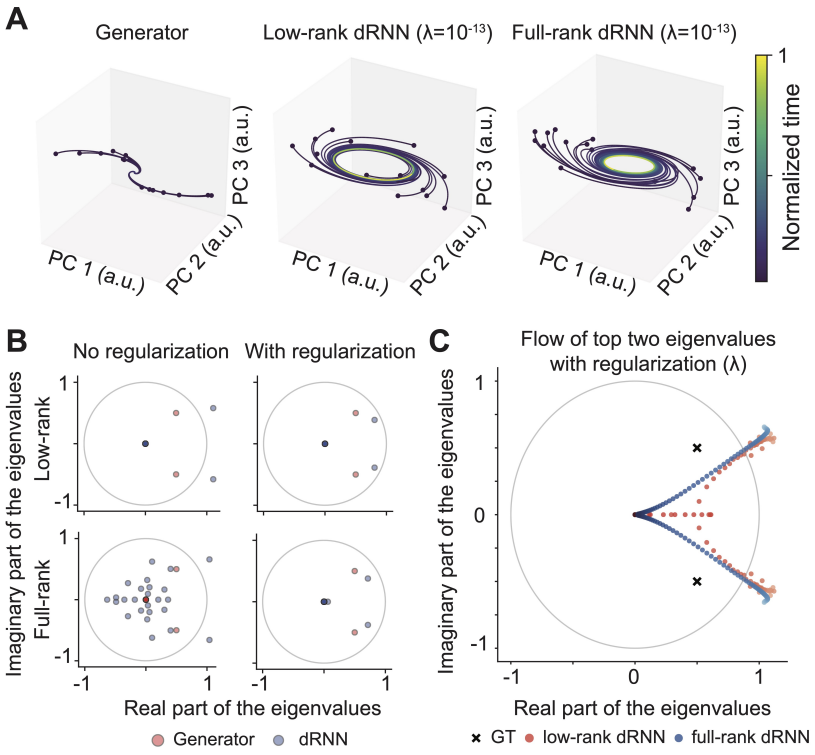

Figure S10: **Low-rank estimation is not sufficient to correctly recover generator dynamics.** We reanalyzed the experiment from (Qian et al., 2024, Figure 5b), in which a generator RNN with only two non-zero (oscillatory, decaying) eigenvalues was reconstructed with a dRNN under partial observation. **A** Principal component projections of neural activity for 15 distinct initializations: ground truth (*left*), rank-2 dRNN (*middle*), and full-rank dRNN (*right*), dRNNs trained with minimal regularization ($\lambda = 10^{-13}$). **B** Eigenvalue spectra of the generator RNN and the reconstructed dRNNs. Optimal ℓ_2 regularization enabled recovery of the correct structure ($\lambda_{\mathrm{low-rank}} = 10^{-3}$, $\lambda_{\mathrm{full-rank}} = 10^{-1}$). **C** Flow of the two largest eigenvalues (by magnitude) as a function of regularization strength. Low-rank dRNNs quickly suppress oscillatory dynamics, whereas full-rank dRNNs preserve the oscillatory structure until it collapses into the origin. Parameters: Generator RNN with $N = 500$ neurons, 25 observed for dRNNs. Dataset from (Qian et al., 2024, Figure 5b): $\alpha = 0.01$, $T = 80000$ time steps, with noise injected at each step ($\epsilon_{\mathrm{in}} \sim \mathcal{N}(0, 1)$), and an extra observation noise ($\mathcal{N}(0, 1)$) was added only to the measured activities and not fed back into the dynamics.

To test this, we reanalyzed the experiment of (Qian et al., 2024, Figure 5), in which a generator RNN with two oscillatory and decaying eigenvalues was reconstructed from partially observed trajectories (Figure S10). When dRNNs were trained with negligible regularization ($\lambda \approx 10^{-13}$), both low-rank and full-rank dRNNs incorrectly generated limit cycles instead of the expected decaying spirals (Fig. S10A). Examining the eigenvalue spectra confirmed this failure (Fig. S10B): neither model matched the ground truth in this regime and both overestimated the eigenvalues, consistent with the observations of Qian et al. (2024). Nevertheless, introducing an ℓ_2 penalty on the weights corrected this behavior, allowing both low-rank and full-rank models to recover the spiraling dynamics. Notably, however, tracking the two dominant eigenvalues under increasing regularization strength (Fig. S10C) further revealed that low-rank dRNNs rapidly suppressed the oscillatory modes by collapsing them into a non-oscillatory form, whereas full-rank dRNNs preserved the correct spiral structure until regularization became excessively strong.

These results demonstrate that low-rank constraints alone do not resolve incorrect estimation and can bias the learned dynamics toward oversimplified solutions, whereas proper weight regularization (as formalized by Theorem 2 and Corollary 1) may be necessary to stabilize recovery of the true dynamics. While a more detailed study of identifiability in low-rank RNNs remains an important

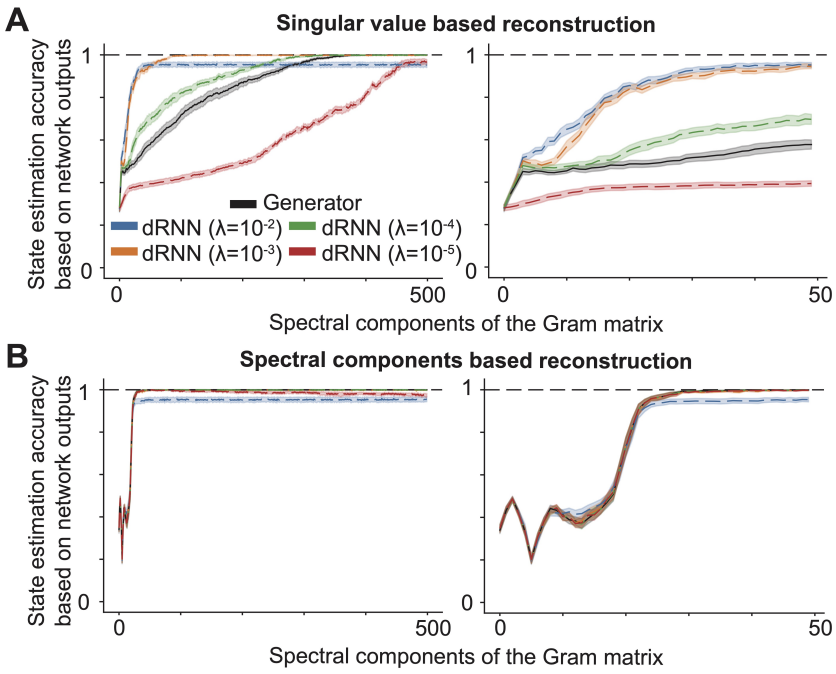

Figure S11: **Only top** $10-20$ **spectral, but not singular, components store task-relevant parameters.** We re-analyzed the RNNs from Fig. 4, but for dRNNs trained with varying numbers of regularization parameters. **A** We used the projected parameters to the top K component of the singular value decomposotion and computed the accuracies of the reconstructed dRNNs to perform the task. *Left.* The full spectrum. *Right.* Close-up into top 50 components. **B** Same as in **A**, but for projections onto top K spectral components of the Gram matrix, computed from the neural activities of the generator RNNs across 1000 trials. Parameters: Same as in Fig. 4.

direction for future work, the present findings highlight that identifiability limitations imposed by the (lack of) richness of the dataset cannot be circumvented simply by enforcing low-rank structure.

B.4 Top spectral components of the Gram matrix reveal task-relevant RNN parameters

Another important and widely discussed aspect of task-trained RNNs is solution degeneracy, referring to the existence of many different parameter configurations that achieve similar task performance (Huang et al., 2025; Cao & Yamins, 2024). Such degeneracy can arise from distinct computational strategies that solve the same task in qualitatively different ways (Kurtkaya et al., 2025), which is fundamentally different from a potential redundancy created by non-identifiable components of RNNs that effectively use the same solution, shared by the same identifiable parameters. The possibility of the existence for the latter case highlights the need to distinguish which components of the parameter space are truly task-relevant and identifiable, and which instead reflect redundant degrees of freedom. Our theoretical results suggest that the identifiable subspace is captured by the top spectral components of the Gram matrix, which encode the directions most strongly constrained by the observed neural activity. If task performance relied equally on all degrees of freedom, no clear cutoff in reconstruction accuracy would be expected.

Strikingly, this is not what we observe. As shown in the main text (Fig. 4), only the top $\sim$ 10–20 spectral components become well constrained, even in networks that reliably perform the task (see also Fig. S5). This suggests that task-relevant information is embedded in a restricted subset of spectral modes rather than being distributed across all parameters, providing a principled resolution to the apparent degeneracy of solutions. To test this further, we compared two types of reconstructions. In the first test, we projected the estimated parameters onto the top K singular vectors of the recurrent weight matrix obtained through singular value decomposition (SVD), a standard method for

analyzing network dimensionality. In the second, we projected the combined parameters θ onto the top K spectral components of the Gram matrix, which by construction capture the data-constrained directions of the parameter space via Theorem 1. Figure S11 shows the outcome of this comparison. Reconstructions based on SVD (Fig. S11**A**) produced performance that depended strongly on the choice of regularization and increased only gradually with K, with no sharp cutoff. By contrast, reconstructions based on the spectral components of the Gram matrix (Fig. S11**B**) achieved near-perfect task performance once the top $\sim$ 10-20 spectral components were included, independent of the regularization strength.

Together, these results demonstrate that task-relevant parameters are concentrated in the top spectral components of the Gram matrix obtained from the trial-relevant activations of the generator RNN (here computed with 1000 trials), not in the top singular vectors of the weight matrix. This distinction shows that many RNNs can solve the same task using the same strategies as long as they share a similar structure in their dominant spectral components, providing a principled framework for distinguishing task-relevant dynamics from a trivially redundant solution degeneracy.

C Proofs of theorems, propositions, and corollaries

In this section, we state (again) and prove the theorems, propositions, and corollaries used in the main text and appendices.

C.1 Proof of Theorem 1

Theorem (Restatement of Theorem 1)**.** *Consider the noiseless RNN defined by Eq. 2 with parameters θ^*. Consider X an observation matrix defining the conditioning space $\mathcal{X}$ and denote $P \in \mathbb{R}^{N_X \times N_X}$ the projection matrix onto its column space. Then, any RNN parametrized by θ such that:*

$$\theta = \theta^* P + \Delta\theta \quad \text{for some } \Delta\theta \text{ that verifies} \quad \Delta\theta P = 0, \tag{S9}$$

gives the same conditional probability distribution as the ground-truth RNN. θ^* is conditionally identifiable if one of the following holds: (i) P is full rank, in which case $P = I$, or (ii) the parameter space is restricted to $\{\theta \in \Theta : \theta(I - P) = 0\}$ (identifiability condition).

Proof. We perform this proof in three steps. We first recast the conditional identifiability criterion for noiseless RNNs into an equivalent deterministic form. Then, we characterize the full set of parameters that yield identical next-step predictions. Finally, we read out the identifiability conditions explicitly.

Step 1: Equivalence of noiseless RNNs. Consider the noiseless discretized RNN in Eq. 2 with parameter θ. Define its flowmap as $F_\theta(\cdot)$ for notational simplicity. Given $x[s] \in \mathcal{X}$, the next state is deterministically

$$r[s+1] = F_\theta(x[s]). \tag{S10}$$

Thus, the conditional probability distribution of $r[s+1]$ given $x[s]$ is

$$P(r[s+1] \mid x[s]; \theta) = \delta\big(r[s+1] - F_\theta(x[s])\big), \tag{S11}$$

where $\delta(\cdot)$ is the Dirac delta distribution centered at the deterministic prediction. Now consider two parameterizations θ_1 and θ_2. Their induced conditional distributions are

$$P(r[s+1] \mid x[s]; \theta_1) = \delta\big(r[s+1] - F_{\theta_1}(x[s])\big), \tag{S12}$$

$$P(r[s+1] \mid x[s]; \theta_2) = \delta\big(r[s+1] - F_{\theta_2}(x[s])\big). \tag{S13}$$

For these two distributions to be equal for all $x[s] \in \mathcal{X}$, their supports must coincide:

$$F_{\theta_1}(x[s]) = F_{\theta_2}(x[s]), \qquad \forall x[s] \in \mathcal{X}. \tag{S14}$$

Therefore, two noiseless RNNs are equivalent in the sense that they produce the same conditional probability distributions if and only if they make the same single-step predictions.

Step 2: Find the set of parameters that can reproduce the same next step predictions deterministically. Let $x \in \mathcal{X}$ be an observation and denote P the projection matrix onto the column space of X. The ground-truth RNN with parameters θ^* produces predictions determined by

$$r[s+1] = (1-\alpha)r[s] + \alpha\,\phi(\theta^* x[s]), \tag{S15}$$

where ϕ is the fixed monotone nonlinearity. Suppose there exists an alternative parameterization θ that yields identical predictions for all $x \in \mathcal{X}$. Then, for all $x \in \mathcal{X}$,

$$\phi(\theta x) = \phi(\theta^* x). \tag{S16}$$

Because ϕ is monotone and applied elementwise, equality of outputs implies

$$\theta x = \theta^* x, \qquad \forall x \in \mathcal{X}. \tag{S17}$$

Equivalently, θ and θ^* must act identically on the subspace spanned by $\mathcal{X}$. This condition can be expressed using the projection P as

$$\theta P = \theta^* P. \tag{S18}$$

Thus, every parameterization θ that reproduces the same predictions can be written as

$$\theta = \theta^* P + \Delta\theta, \qquad \Delta\theta P = 0. \tag{S19}$$

Conversely, if θ takes this form, then for any $x \in \mathcal{X}$ we can write $x = Px$, which gives

$$\theta x = \theta^* P x + \Delta\theta P x = \theta^* P x = \theta^* x, \tag{S20}$$

showing that θ and θ^* produce identical single-step predictions. Hence, the set of parameters that can reproduce the same next-step predictions as θ^* is exactly

$$\{\theta = \theta^* P + \Delta\theta : \Delta\theta P = 0\}. \tag{S21}$$

Step 3: Read out the identifiability conditions explicitly. Finally, conditional identifiability of θ^* requires uniqueness within the admissible parameter set. This holds in either of the following cases: (i) if P is full rank, then $P = I$ and the parameter is uniquely determined; or (ii) if we restrict the parameter space to $\{\theta \in \Theta : \theta(I - P) = 0\}$, ensuring that no additional unconstrained components remain. □

C.2 Proof of Proposition S1

Proposition (Restatement of Proposition S1)**.** *Let $x[i] = [x_{\text{obs}}[i], x_{\text{unobs}}[i]]$, where $x_{\text{obs}}[i]$ denotes the observed neural activities and inputs for $i = 1, \ldots, T$, and $x_{\text{unobs}}[i]$ the corresponding unobserved variables. Let θ^*_{obs} denote the parameters among observed neurons, and θ^*_{unobs} those involving unobserved neurons. Let $\phi(\cdot)$ be a function. Then, when $\theta_{\text{unobs}} = \theta^*_{\text{unobs}}$, all RNNs with parameters θ_{obs} produce the same observed neural trajectory provided that*

$$\theta_{\text{obs}} = \tilde{\theta}_{\text{obs}} + \Delta\theta_{\text{obs}}, \quad \textit{where} \quad \tilde{\theta}_{\text{obs}} = \theta^*_{\text{obs}} P, \quad \textit{and} \quad \Delta\theta_{\text{obs}} P = 0, \tag{S22}$$

with P denoting the projection operator onto the observed subspace of the neural dataset.

Proof. By assumption, the parameters involving unobserved neurons are fixed at their ground-truth values, $\theta_{\text{unobs}} = \theta^*_{\text{unobs}}$. Hence, the observed trajectory depends only on the action of θ_{obs} on the observed inputs $x_{\text{obs}}[i]$. Applying Theorem 1 to the observed subspace spanned by $\{x_{\text{obs}}[i]\}_{i=1}^T$, we obtain that all parameterizations θ_{obs} yielding the same observed trajectories must satisfy

$$\theta_{\text{obs}} = \theta^*_{\text{obs}} P + \Delta\theta_{\text{obs}}, \qquad \Delta\theta_{\text{obs}} P = 0, \tag{S23}$$

where P denotes the projection operator onto the column space of the observed dataset. Thus, any such θ_{obs} produces the same observed neural trajectory when paired with $\theta_{\text{unobs}} = \theta^*_{\text{unobs}}$. □

C.3 Proof of Proposition S2

Proposition (Restatement of Proposition S2)**.** *Consider the noiseless RNN in Eq. 2 with $\epsilon_{\text{in/conv}}(t) = 0$. Suppose the recurrent weights are parameterized as $W^{\text{rec}} = CD$ with $C \in \mathbb{R}^{N \times K}$, $D \in \mathbb{R}^{K \times N}$, and $K \ll N$. Let X be the observation matrix with the projection operator P onto its column space. Then any parameterization of the form $W^{\text{rec}} = C(D + \Delta D)$ with $\Delta D P = 0$ produces the same neural dataset $\mathcal{D}$ while preserving $\text{rank}(W^{\text{rec}}) \leq K$.*

Proof. We prove the claim in three steps. We first express how the low-rank recurrent parameterization determines the observed trajectories. We then characterize the family of equivalent parameterizations that leave the dataset unchanged. Finally, we verify that these parameterizations preserve the low-rank constraint.

Step 1: Express the recurrent contribution. The recurrent weights are parameterized as $W^{\mathrm{rec}} = CD$ with $C \in \mathbb{R}^{N\times K}$ and $D \in \mathbb{R}^{K\times N}$. For an input $x \in \mathcal{X}$, the recurrent contribution to the update is

$$W^{\mathrm{rec}}x = CDx. \tag{S24}$$

Thus, the predictions of the RNN depend directly on how D acts on the projection of x onto the column space of X, i.e. onto Px. In other words, Dx should remain invariant for the single-step predictions to remain invariant.

Step 2: Characterize equivalent parameterizations. Consider a perturbation $\tilde{D} = D + \Delta D$. Then for any $x \in \mathcal{X}$,

$$C\tilde{D}x = C(D + \Delta D)x = CDx + C\Delta Dx. \tag{S25}$$

Since $x = Px$ for $x \in \mathcal{X}$, we have

$$C\tilde{D}x = CDPx + C\Delta DPx. \tag{S26}$$

If $\Delta DP = 0$, the second term vanishes, yielding

$$C\tilde{D}x = CDPx = CDx. \tag{S27}$$

Therefore, $C(D + \Delta D)$ produces the same outputs as CD for all $x \in \mathcal{X}$.

Step 3: Verify preservation of rank constraint and conclude the argument. Since $\tilde{D} = D + \Delta D$ is still a $K \times N$ matrix, the rank of the recurrent weights satisfies

$$\mathrm{rank}(W^{\mathrm{rec}}) = \mathrm{rank}(C\tilde{D}) \leq K, \tag{S28}$$

i.e., the low-rank structure is preserved. Therefore, any recurrent weight matrix of the form $W^{\mathrm{rec}} = C(D+\Delta D)$ with $\Delta DP = 0$ yields the same neural dataset $\mathcal{D}$ while ensuring $\mathrm{rank}(W^{\mathrm{rec}}) \leq K$. □

C.4 Proof of Theorem 2

Theorem (Restatement of Theorem 2)**.** *Under the assumptions and notation of this section, consider an RNN whose parameters $\theta \in \mathbb{R}^{N_{\mathrm{rec}}\times N_X}$ is estimated by gradient descent of a differentiable loss $L(\theta)$. Assume that the gradient satisfies $v^{(r)T}\nabla L(\theta) = O(\sigma_r^n)$ for every $\theta \in \Theta$ and some n. If $\theta^{(k)}P = \theta^{(k)}$ at iteration k of the gradient descent, then for any λ with $TM\sigma_\epsilon^2 \ll \lambda \ll \tilde{\sigma}_R^2$, and for any step $\alpha > 0$, the update*

$$\theta^{(k+1)} = \theta^{(k)} - \alpha\,\nabla L(\theta)\left(X^TX + \lambda I\right)^{-1}, \tag{S29}$$

is a descent direction that satisfies $\theta^{(k+1)}P = \theta^{(k+1)} + O(\sigma_\epsilon^n/\lambda)$.

Proof. We prove the claim in three steps. We first expand the preconditioner using the SVD of X. We then show that the update approximately preserves the identifiability condition $\theta^{(k)}P = \theta^{(k)}$. Finally, we verify that the update direction is a descent direction under the spectral assumptions.

Step 1: Expand the preconditioner via SVD. Let $X = U\Sigma V^T$ be the singular value decomposition, with $V = [v^{(1)}, \ldots, v^{(N_X)}]$ and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_{N_X})$. Then

$$X^TX + \lambda I = V(\Sigma^2 + \lambda I)V^T, \qquad (X^TX + \lambda I)^{-1} = V(\Sigma^2 + \lambda I)^{-1}V^T. \tag{S30}$$

In this basis, directions $v^{(r)}$ with large singular values $\sigma_r^2 \gg \lambda$ are nearly preserved, while directions with $\sigma_r^2 \ll \lambda$ are suppressed.

Step 2: Show approximate preservation of identifiability. Suppose $\theta^{(k)}P = \theta^{(k)}$. The update rule is

$$\theta^{(k+1)} = \theta^{(k)} - \alpha\,\nabla L(\theta)\,(X^TX + \lambda I)^{-1}. \tag{S31}$$

Right–multiplying by P gives

$$\theta^{(k+1)}P = \theta^{(k)}P - \alpha\,\nabla L(\theta)\,(X^TX + \lambda I)^{-1}P. \tag{S32}$$

Since $\theta^{(k)}P = \theta^{(k)}$, it remains to compare the last term with the unprojected update. Expanding in the eigenbasis $\{v^{(r)}\}$,

$$(X^TX + \lambda I)^{-1}v^{(r)} = \frac{1}{\sigma_r^2 + \lambda}v^{(r)}. \tag{S33}$$

By assumption, the gradient component satisfies $v^{(r)T}\nabla L(\theta) = O(\sigma_r^n)$ by assumption. Thus the contribution introduced by projecting is at most of order $C_r = \frac{\sigma_r^n}{\sigma_r^2+\lambda}$. Noting that $\sigma_r^2 = \tilde{\sigma}_r^2 + TM\sigma_\epsilon^2$ and $\tilde{\sigma}_r = 0$ for $r > R$, we have the following contributions

$$C_r \begin{cases} \sim O(1) & \text{for} \quad r \le R \\ \sim O(\sigma_\epsilon^n/\lambda) & \text{for} \quad r > R \end{cases} \tag{S34}$$

which follows from $TM\sigma_\epsilon^2 \ll \lambda \ll \tilde{\sigma}_R^2$. Here, $O(1)$ refers to independence from the regularization. Hence

$$\theta^{(k+1)}P = \theta^{(k+1)} + O(\sigma_\epsilon^n/\lambda). \tag{S35}$$

Step 3: Verify descent property. This follows from the fact that $X^TX + \lambda I$ is positive semi-definite such that

$$-Tr[\nabla\mathcal{L}(\theta)(X^TX + \lambda I)^{-1}\nabla\mathcal{L}(\theta)^T] \le 0. \tag{S36}$$

Thus, the preconditioned update in Eq. equation 7 both preserves the identifiability condition up to $O(\sigma_\epsilon^n/\lambda)$ and guarantees descent, as claimed. □

C.5 Proof of Corollary 1

Corollary (A longer version of Corollary 1)**.** *Consider the scenario in Theorem 2, and suppose the loss is*

$$\mathcal{L}(\theta) = \sum_{i=1}^{TM} L\big(Y_i, g(\theta^T X_i)\big) + \lambda\|\theta\|_2^2, \tag{S37}$$

where L is twice differentiable in its second argument, g is smooth, and $\lambda \ge 0$ is a regularization parameter. Define $X = \sum_{r=1}^{N}\sqrt{TM}\sigma_r v^{(r)}u^{(r)T}$ Then, the following statements hold:

1. *Around a minimizer $\bar{\theta}$, $\mathcal{L}(\theta)$ is locally equivalent to the weighted ridge regression objective*

$$\bar{\mathcal{L}}(\theta) = \|W^{1/2}(Y - X\theta)\|_2^2 + \lambda\|\theta\|_2^2, \tag{S38}$$

   where W depends on the local minimum $\bar{\theta}$.

2. *Assuming $W = I$ without loss of generality, the stationary point of $\bar{\mathcal{L}}$ is*

$$\bar{\theta} = (X^TX + \lambda I)^{-1}X^TY = \sum_{r=1}^{N}\frac{\sigma_r}{\sigma_r^2 + \lambda}\,[u^{(r)T}Y]\,v^{(r)}. \tag{S39}$$

3. *In the noiseless case ($\sigma_\epsilon = 0$), if $\lambda = 0$ and $\tilde{\sigma}_r = 0$ for some $r > R$, the solution is not unique: any component of $\bar{\theta}$ parallel to the kernel $\ker(X)$ is arbitrary. In the noisy case ($\sigma_\epsilon > 0$), these null directions acquire empirical singular values of order σ_ϵ, yielding coefficients*

$$\frac{1}{1 + \left(\frac{\lambda}{\sigma_\epsilon^2}\right)}\frac{[u^{(r)T}Y]}{\sigma_\epsilon}, \quad \text{for} \quad r > R. \tag{S40}$$

   When $\lambda \gg \sigma_\epsilon^2$ and assuming a correlation between $Y_i \sim X_i$ such that $v^{(r)T}Y \sim O(\sigma_r)$, such spurious contributions are suppressed to order $O(\sigma_\epsilon^2/\lambda)$.

4. If $\sigma_\epsilon^2 \ll \lambda \ll \tilde{\sigma}_R^2$, the solution approximates

$$\bar{\theta} \approx \sum_{r=1}^{R} \frac{1}{\tilde{\sigma}_r} [\tilde{u}^{(r)T} Y] \, \tilde{v}^{(r)}, \tag{S41}$$

i.e., the uniquely identifiable part of the estimator.

Proof. We prove each statement in turn.

Step 1: Reduce to a weighted ridge regression objective. Consider the loss

$$\mathcal{L}(\theta) = \sum_{i=1}^{T} L\big(Y_i, g(\theta^T X_i)\big) + \lambda \|\theta\|_2^2. \tag{S42}$$

Since L is twice differentiable in its second argument and g is smooth, a second-order Taylor expansion of $L(Y_i, g(\theta^T X_i))$ around a minimizer $\bar{\theta}$ yields a quadratic approximation of $\mathcal{L}(\theta)$ near $\bar{\theta}$. Collecting the quadratic terms gives a weighted least-squares objective

$$\bar{\mathcal{L}}(\theta) = \|W^{1/2}(Y - X\theta)\|_2^2 + \lambda \|\theta\|_2^2, \tag{S43}$$

where W is a positive definite weight matrix depending on $\bar{\theta}$. This proves (1).

Step 2: Solve the stationary point of the quadratic problem. Without loss of generality, assume $W = I$ (this can be absorbed into a change of variables by $X \to W^{1/2}X$ and $Y \to W^{1/2}Y$ and redefining σ_r^2). The stationary point satisfies the normal equations

$$(X^T X + \lambda I)\bar{\theta} = X^T Y. \tag{S44}$$

Thus

$$\bar{\theta} = (X^T X + \lambda I)^{-1} X^T Y. \tag{S45}$$

Expanding using the SVD $X = U\Sigma V^T$, with singular values σ_r and singular vectors $u^{(r)}, v^{(r)}$, yields

$$\bar{\theta} = \sum_{r=1}^{N} \frac{\sigma_r}{\sigma_r^2 + \lambda} [u^{(r)T} Y] \, v^{(r)}. \tag{S46}$$

This proves (2).

Step 3: Analyze uniqueness in noiseless vs noisy cases. In the noiseless case ($\sigma_\epsilon = 0$), if $\lambda = 0$ and $\tilde{\sigma}_r = 0$ for some $r > R$, then any component of $\bar{\theta}$ in $\ker(X)$ is arbitrary, because $(X^T X)^{-1}$ is undefined along these directions. Hence the solution is not unique.

In the noisy case ($\sigma_\epsilon > 0$), the empirical Gram matrix $X^T X$ acquires perturbations of order σ_ϵ^2, so directions $r > R$ in the kernel now appear with effective singular values of order σ_ϵ. In these directions, the coefficients take the form

$$\frac{1}{\sigma_r^2 + \lambda} \sigma_r [u^{(r)T} Y] v^{(r)} \sim \frac{1}{\sigma_\epsilon^2 + \lambda} \sigma_\epsilon [u^{(r)T} Y] v^{(r)}. \tag{S47}$$

Equivalently, one may write

$$\frac{1}{1 + \left(\frac{\lambda}{\sigma_\epsilon^2}\right)} \frac{[u^{(r)T} Y]}{\sigma_\epsilon}, \quad r > R. \tag{S48}$$

When $\lambda \gg \sigma_\epsilon^2$ and assuming $v^{(r)T} Y \sim O(\sigma_r)$, these contributions are suppressed to order $O(\sigma_\epsilon^2/\lambda)$. This proves (3).

Step 4: Approximate the identifiable estimator. If $\sigma_\epsilon^2 \ll \lambda \ll \tilde{\sigma}_R^2$, then directions with $\sigma_r^2 \gg \lambda$ (for $r \le R$) are nearly unaffected by the ridge term (*i.e.*, remain $O(1)$), while directions with $\sigma_r^2 \ll \lambda$ (for $r > R$) are heavily damped (*i.e.*, are $O(\lambda^{-1})$). Thus the solution approximates

$$\bar{\theta} \approx \sum_{r=1}^{R} \frac{1}{\tilde{\sigma}_r} [\tilde{u}^{(r)T} Y] \, \tilde{v}^{(r)}, \tag{S49}$$

which is precisely the identifiable part of the estimator. This proves (4).

Conclusion. Each of the four claims follows under the assumptions of Theorem 2 (but with a redefinition of σ_r to remove TM dependencies), establishing the corollary. □

C.6 Proof of Theorem 3

Theorem (Restatement of Theorem 3)**.** *Let $S_{\text{id}} = \text{span}\{v_1, \ldots, v_R\}$ be the identifiable subspace spanned by the top R spectral eigenvectors of the Gram matrix, and assume that for a noiseless, task-performing RNN with dynamics in Eq. 1, the activities satisfy $r[t] \in S_{\text{id}}$ for all t. Let $\tilde{\theta}$ be identifiable with $\tilde{\theta}P_{\text{id}} = \tilde{\theta}$, where P_{id} projects onto S_{id}. Then, any parameterization $\theta = \tilde{\theta} + \Delta\theta$ with $\Delta\theta P_{\text{id}} = 0$ but $\Delta\theta \neq 0$ yields identical dynamics $\dot{r}[t]$ for all $r[t] \in S_{\text{id}}$, but not necessarily when $r[t] \notin S_{\text{id}}$.*

Proof. The RNN dynamics are given by

$$\dot{r}[t] = -r[t] + \phi(\theta r[t]), \tag{S50}$$

with ϕ applied elementwise. Let $\tilde{\theta}$ be an identifiable parameterization such that $\tilde{\theta}P_{\text{id}} = \tilde{\theta}$, where P_{id} projects onto S_{id}. Consider now $\theta = \tilde{\theta} + \Delta\theta$ with $\Delta\theta P_{\text{id}} = 0$ and $\Delta\theta \neq 0$. If $r[t] \in S_{\text{id}}$, then $r[t] = P_{\text{id}} r[t]$, and hence

$$\theta r[t] = \tilde{\theta}P_{\text{id}} r[t] + \Delta\theta P_{\text{id}} r[t] = \tilde{\theta} r[t]. \tag{S51}$$

It follows that

$$\dot{r}[t] = -r[t] + \phi(\theta r[t]) = -r[t] + \phi(\tilde{\theta} r[t]), \tag{S52}$$

so the dynamics under θ and $\tilde{\theta}$ coincide for all $r[t] \in S_{\text{id}}$. If $r[t] \notin S_{\text{id}}$, then $(I - P_{\text{id}})r[t] \neq 0$, and in general

$$\theta r[t] = \tilde{\theta}P_{\text{id}} r[t] + \Delta\theta(I - P_{\text{id}})r[t] \neq \tilde{\theta} r[t], \tag{S53}$$

so the dynamics need not coincide. This proves the claim. □

D Methods

D.1 Identifiability in dynamical systems

In dynamical system models, the prediction depends not only on the parameters θ^*, but also on the current state of the system and any external inputs. Let $X_i \in \mathbb{R}^{N+N_{\text{in}}}$ denote the combined state and input at time i, where N is the number of state variables and N_{in} is the number of input dimensions. Then, we formally write the data generation process as:

$$Y_i \sim P(Y_i|X_i;\theta^*), \tag{S54}$$

where the observed data consist of pairs (X_i, Y_i) for $i = 1, \ldots, T$. Here, $P(Y_i|X_i;\theta^*)$ is the conditional distribution of Y_i given X_i, parameterized by the deterministic parameter values θ^*. For our purposes, one can assume $Y_i \in \mathbb{R}^N$ refers to the state variables in the next time step. With this data generation model, we can now define the concept of conditional identifiability:

Definition S1 (Conditional Identifiability)**.** *Let $\mathcal{P} = \{P(\cdot|\cdot;\theta) : \theta \in \Theta\}$ be a statistical model with parameter space Θ. Let the ground truth data generation process follow the distribution $Y_i \sim P(Y_i|X_i;\theta^*)$ for some unknown θ^*, where $Y_i \in \mathbb{R}^N$ refers to the observed sample i and $X_i \in \mathbb{R}^{N+N_{\text{in}}}$ the observed sample i of a set of auxiliary random variables. We say that $\mathcal{P}$ is conditionally identifiable if the mapping $\theta \mapsto P(\cdot|\cdot;\theta)$ is one-to-one for all possible values of X:*

$$\forall X \in \mathcal{X} \quad P(Y_i|X_i;\theta_1) \hat{=} P(Y_i|X_i;\theta_2) \implies \theta_1 = \theta_2, \tag{S55}$$

where $\mathcal{X}$ refers to the domain of X.

We note that in the statistics literature, conditional identifiability sometimes refers to identifiability under additional identification conditions. In contrast, here we use the term to denote identifiability with respect to conditional probability distributions.

As we show below, whether the distribution $P(X)$ has support over the full domain $\mathcal{X} = \mathbb{R}^{N+N_{\text{in}}}$ is particularly relevant for the identifiability of RNNs. This definition can easily generalize to RNNs, but first, we need to specify the observables. For simplicity of notation, we assume that a discretized neural trajectory is observed along with the corresponding inputs. Specifically, we define the observed neural trajectory as $\mathcal{O} := \{r[1], r[2], \ldots, r[T]\}$, and the inputs are denoted similarly

as $u[1], \ldots, u[T]$. Since $r[s] \in \mathbb{R}^N$ are observed and α known by the definition of RNNs in Eq. 1, we can define an observed sample as:

$$d_i[s] = \frac{r_i[s+1] - (1-\alpha) r_i[s]}{\alpha}. \tag{S56}$$

Earlier work has shown that $d_i[s]$ are numerically convenient to work with for empirical estimation Dinc et al. (2023). Theoretically, following Eq. 2, $d[s]$ values can be considered samples of a random vector defined as $Y := \phi(\theta^* X + \epsilon_{\text{in}}) + \epsilon_{\text{conv}}$, where θ^* is the concatenated weight matrix and X contains corresponding neural activities and outside inputs, as defined above. Then, with the samples $X_i := x[i]$ and $Y_i := d[i]$, we recover the form of the data generation model introduced in Eq. S54. This allows us to formally define conditional identifiability for RNNs constrained on a trajectory $\mathcal{O}$:

Definition S2 (Conditional Identifiability in RNNs)**.** *Let $\theta \in \Theta$ denote the parameters of an RNN model and let $\mathcal{O}$ denote the set of observed quantities as defined above. We say that the RNN is* conditionally identifiable *on $\mathcal{O}$ if the following holds:*

$$\forall i = 1, \ldots, T; \quad P(Y_i | X_i; \theta_1) = P(Y_i | X_i; \theta_2) \quad \Longrightarrow \quad \theta_1 = \theta_2. \tag{S57}$$

Note that if $\theta_1 \neq \theta_2$ yield exactly the same next-step predictions within $\mathcal{O}$, then (noiseless) RNNs with either parameter set would yield the exact same neural trajectory $\mathcal{O}$ when initialized with $r[0]$. Therefore, even though defined on the next step predictions, this definition formalizes the condition under which the parameters of an RNN are uniquely determined by the sequence of (noiseless) neural activities and inputs observed along a trajectory. Notably, however, the converse is not true. For a simple example, noisy RNNs can produce distinct trajectories even if they have exact same given parameters. Thus, we find it instructive to define identifiability in a noiseless setting (which is often the case, *e.g.*, the limit of infinite samples Lehmann & Casella (2006)), and then test the implications for noisy RNNs empirically or by considering expectation values (*e.g.*, see Eq. 6).

D.2 Experiment Details for Reproducibility

Here, we provide details of our experiments to ensure reproducibility. Additional details can be found in the code shared in the supplementary materials.

D.2.1 Recurrent Neural Networks

As described in the Background section (Section 2), we use a biologically motivated and interpretable class of RNNs Perich et al. (2021); Dinc et al. (2025). Since we focus on the discrete version of the RNNs, we utilize the Euler discretization described in Equation 2. In this section, we specify our implementation choices: how we initialize the weight matrices W^{rec}, W^{in}, and W^{out}, the distributions we sample for noise terms ϵ_{in} and ϵ_{conv}, and other implementation details.

For reference, we construct RNN dynamics as follows:

$$\tau \dot{r}(t) = -r(t) + \phi(W^{\text{rec}} r(t) + W^{\text{in}} u(t) + \epsilon_{\text{in}}(t)) + \epsilon_{\text{conv}}(t) \tag{S58}$$

$$\hat{o}(t) = \psi(W^{\text{out}} r(t)) \tag{S59}$$

where $\tau \in \mathbb{R}$ represents the neuronal time constant, $r(t) \in \mathbb{R}^{N_{\text{rec}}}$ the neural activities, $\dot{r}(t) \in \mathbb{R}^{N_{\text{rec}}}$ their temporal derivatives, $u(t) \in \mathbb{R}^{N_{\text{in}}}$ the input signals, and $\hat{o}(t) \in \mathbb{R}^{N_{\text{out}}}$ the network outputs. In our experiments, we set $\phi(\cdot) = \tanh(\cdot)$ and $\psi(\cdot)$ as identity, $\tanh$, or sigmoid depending on the task, and use discretization parameter α, which is calculated as the ratio of sampling interval Δt to time constant τ. Note that while the output weights $W^{\text{out}} \in \mathbb{R}^{N_{\text{rec}} \times N_{\text{out}}}$ are used when training task-performing RNNs to generate ground-truth neural trajectories (as described in the following section), they are not involved in the parameter recovery process.

In our RNN implementation, we use Kaiming and Xavier initializations He et al. (2015); Glorot & Bengio (2010) with uniform and normal distributions for the weight parameters W^{in}, W^{rec}, and W^{out}. For the input noise ϵ_{in} and conversion noise ϵ_{conv}, we implement Gaussian, Laplace, and Poisson distributions. However, we use the Poisson distribution predominantly in our experiments. During firing rate updates, since conversion noise ϵ_{conv} can cause values to exceed the bounds $[-1, 1]$, we clip the firing rates after each update: $r(t) = 1 - 10^{-6}$ when $r(t) \geq 1$ and $r(t) =$

$-1 + 10^{-6}$ when $r(t) \leq -1$. Initial firing rates $r(t = 0)$ are sampled from Gaussian or uniform distributions depending on the experiment.

To ensure reproducibility, we set fixed random seeds for Python's random, NumPy, and PyTorch random number generators. All detailed information about distribution selections and hyperparameters for each experiment can be found in Section D.2.6.

D.2.2 Obtaining ground truth neural trajectories

In our parameter recovery experiments, we use two different methodologies for generating ground truth neural trajectories. First, we use chaotic networks where we initialize parameters randomly and iterate without any supervision. Second, we train RNNs on one of three tasks (described in the following section: 3-bit flip-flop, delayed cue discrimination, delayed match-to-sample) and then examine parameter recovery in the presence of task-induced structure.

Chaotic networks: We use randomly connected recurrent neural networks to generate chaotic dynamics without any task-specific constraints. These networks consist of N_{rec} recurrently connected units with weights sampled from a Gaussian distribution $\mathcal{N}(\mu = 0, \sigma = 2/N_{\text{rec}})$, ensuring the network operates in a chaotic regime. The networks receive no external input during trajectory generation ($u(t) = 0$) and evolve according to their internal dynamics alone. Initial firing rates are sampled uniformly from $[-1, 1]$, and the system is iterated using the standard RNN update equation with $\tanh$ nonlinearity and step size $\alpha = 0.1$. These chaotic networks produce rich, complex temporal patterns that exhibit sensitive dependence on initial conditions while remaining bounded within the activation function's range. By studying parameter recovery from such unconstrained dynamics, we can assess identifiability in its most general form—without the structural biases imposed by task optimization.

Trained networks: In all training tasks, we train neural networks using input-output supervision, allowing networks to learn internal dynamics specific to each task. During initialization, we use Xavier initialization with uniform distribution as implemented in PyTorch.

Task-specific configurations vary as follows: for bias terms, we include learnable biases in the input and output linear layers of DCD and DMTS tasks but exclude biases in 3-bit flip-flop experiments. For output nonlinearities, we set $\theta(\cdot)$ as identity in 3-bit flip-flop, sigmoid in DMTS, and tanh in DCD tasks. Initial firing rates are sampled from $\mathcal{N}(0, \sqrt[4]{N_{\text{rec}}})$ in 3-bit flip-flop, from $\tanh$ applied to $\mathcal{N}(0, 1)$ in DMTS, and from $\tanh$ applied to $\mathcal{N}(0, 0.1)$ in DCD. Since each task has different input-output requirements, the input dimension N_{in} equals 3 in 3-bit flip-flop, 1 in DMTS, and 1 in DCD.

For all task training, we use Mean Squared Error (MSE) loss. The optimizers vary by task: we use Adam optimizer for 3-bit flip-flop and DMTS, while employing SGD for DCD. Additionally, we employ the ReduceLROnPlateau learning rate scheduler (with factor 0.5 and patience equal to the number of epochs) specifically in 3-bit flip-flop experiments. Table S1 summarizes the key training hyperparameters for each task. Section D.2.6 covers additional hyperparameter configurations for figures.

D.2.3 Noise generation processes

In most experiments, we sample noise independently at each timestep. However, for specific experiments examining the effects of realistic noise correlations, we implement spatially and temporally correlated noise.

Standard (uncorrelated) noise: By default, both input noise ϵ_{in} and conversion noise ϵ_{conv} are sampled independently at each timestep from the specified distributions (Gaussian, Laplace, or Poisson) with no spatial or temporal correlations.

Correlated noise (experiment-specific): In selected experiments, we generate spatially and temporally correlated input noise ϵ_{in} to model realistic neural recordings where nearby neurons and adjacent timepoints exhibit correlated fluctuations. First, we sample uncorrelated noise from $\mathcal{N}(0, \sigma)$ with dimensions $T \times N_{\text{rec}}$, where T is the number of timesteps and N_{rec} is the number of neurons. To introduce spatial and temporal correlations, we construct a 2D Gaussian kernel:

Table S1: Training hyperparameters to obtain generator networks for each task

Hyperparameter	**3-bit flip-flop**	**DCD**	**DMTS**
Network size (N_{rec})	$100, 500, 1000$	500	1000
Input dimension (N_{in})	3	1	1
Output dimension (N_{out})	3	1	1
Number of epochs	20000	5000	5000
Batch size	50	10	10
Learning rate	10^{-4}	10^{-3}	10^{-4}
Optimizer	Adam	SGD	Adam
LR scheduler	ReduceLROnPlateau	None	None
α (discretization)	0.5	0.5	0.5
Δt (ms)	5×10^{-3}	5×10^{-3}	5×10^{-3}
τ (ms)	10×10^{-3}	10×10^{-3}	10×10^{-3}
Input noise (ϵ_{in})	0	10^{-3}	0
Output nonlinearity (θ)	identity	tanh	sigmoid
Number of seeds	20	20	20

$$K(x, y) = \exp\left(-\frac{x^2}{2\sigma_T^2} - \frac{y^2}{2\sigma_N^2}\right) \tag{S60}$$

where σ_T controls temporal correlation strength and σ_N controls spatial (across-neuron) correlation strength. The kernel is normalized such that $\sum K(x, y) = 1$. We then convolve the uncorrelated noise with this kernel:

$$\epsilon_{\text{in}}^{\text{corr}} = K * \epsilon_{\text{in}}^{\text{uncorr}} \tag{S61}$$

where $*$ denotes 2D convolution with 'nearest' boundary conditions. Finally, we rescale the correlated noise to maintain the desired standard deviation σ:

$$\epsilon_{\text{in}} = \epsilon_{\text{in}}^{\text{corr}} \cdot \frac{\sigma}{\text{std}(\epsilon_{\text{in}}^{\text{corr}})} \tag{S62}$$

In these experiments, we use $\sigma_T = 3$ timesteps for temporal correlation and $\sigma_N = 50$ neurons for spatial correlation, with kernel size 30×30. Conversion noise ϵ_{conv} remains uncorrelated even in these experiments.

D.2.4 Description of the tasks

Here, we clarify the implementation details and structure of three neuroscience-inspired tasks. First, we explain the 3-bit flip-flop task, where the network must maintain and selectively update multiple internal memory states. Second, we describe the delayed cue discrimination (DCD) task, where the network must classify an input signal and give an output after a delay period. Third, we explain our final task, delayed match-to-sample (DMTS), where the network must compare two sequential inputs and determine whether they match.

3-bit flip-flop: This task consists of three independent input channels where the values are $u_i(t) \in \{+1, 0, -1\}$ for $i \in \{1, 2, 3\}$. When a channel receives a positive or negative input signal, the network must output the corresponding value in that channel until a new non-zero signal arrives in the same channel. Importantly, inputs are presented randomly across trials, and after each presentation, the input signal returns to zero until the next random signal arrives. Therefore, the RNN must simultaneously maintain information from all three channels while producing the correct output signals.

Formally, the input dynamics are defined as:

$$u_i(t) = \begin{cases} \pm 1 & \text{if } B_i(t) \sim \text{Bernoulli}(0.05) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{S63}$$

where $B_i(t)$ is a Bernoulli trial for channel i at time t, and when $B_i(t) = 1$, the sign is chosen uniformly at random.

The network output follows a flip-flop dynamic where each channel starts at zero and latches to the most recent non-zero input:

$$o_i(t+1) = \begin{cases} u_i(t) & \text{if } u_i(t) \neq 0 \\ o_i(t) & \text{otherwise} \end{cases} \quad \text{with } o_i(0) = 0 \tag{S64}$$

Delayed cue discrimination (DCD): The delayed cue discrimination task is more complex than 3-bit flip-flop as it requires both classification and delayed response. This task consists of three main intervals: input interval T_{in}, delay interval T_{delay}, and response interval T_{resp}. During the input interval, a cue of ± 1 is presented in a single input channel. Throughout this period, the RNN must latch the information but should not produce any output, opposite to 3-bit flip-flop. After the input interval ends, the input becomes 0 and the RNN must continue to maintain the output at 0 during the delay interval. During the response interval, the RNN must produce a classification output based on the cue: if the cue was $+1$, the output should be $+1$; if the cue was -1, the output should be -1.

Formally, the input signal can be formalized as follows:

$$u(t) = \begin{cases} \pm 1 & \text{if } t \in T_{\text{in}} \\ 0 & \text{otherwise} \end{cases} \tag{S65}$$

The expected output is described as:

$$\hat{o}(t) = \begin{cases} +1 & \text{if } u_{\text{in}} = +1 \text{ and } t \in T_{\text{resp}} \\ -1 & \text{if } u_{\text{in}} = -1 \text{ and } t \in T_{\text{resp}} \\ 0 & \text{otherwise} \end{cases} \tag{S66}$$

where u_{in} denotes the input value during T_{in}.

Delayed match-to-sample (DMTS): The third task is delayed match-to-sample. While sharing similarities with the delayed cue-discrimination task (delayed response, single input channel, and input classification), DMTS requires the network to compare two sequential inputs and respond accordingly. This task includes four distinct intervals: input interval T_{in}, delay interval T_{delay}, match interval T_{match}, and response interval T_{resp}. Similar to delayed cue-discrimination, the RNN should only produce the corresponding output during the response interval. Throughout the input interval, an input of ± 1 is presented; afterward, during the delay period, the signal becomes 0. After the delay period ends, another input of ± 1 is presented during the matching interval. In the response interval, if the input and matching signals match, the RNN must give a positive response $(+1)$; otherwise, the RNN should give a negative response (-1).

More formally, we can describe the input signal as follows:

$$u(t) = \begin{cases} \pm 1 & \text{if } t \in T_{\text{in}} \cup T_{\text{match}} \\ 0 & \text{otherwise} \end{cases} \tag{S67}$$

The ground truth output is described as:

$$\hat{o}(t) = \begin{cases} +1 & \text{if } u_{\text{in}} = u_{\text{match}} \text{ and } t \in T_{\text{resp}} \\ -1 & \text{if } u_{\text{in}} \neq u_{\text{match}} \text{ and } t \in T_{\text{resp}} \\ 0 & \text{otherwise} \end{cases} \tag{S68}$$

where u_{in} denotes the input value during T_{in} and u_{match} denotes the input value during T_{match}.

D.2.5 Fitting RNN parameters to reproduce neural trajectories

After obtaining ground truth neural trajectories from chaotic or trained networks, we fit new RNN parameters to reproduce these observed dynamics. Rather than using computationally expensive backpropagation through time (BPTT), we employ a single-step prediction approach that frames parameter estimation as a feedforward regression problem.

Single-step prediction framework: Given a trajectory of firing rates $r[0], r[1], \ldots, r[T]$ and corresponding inputs $u[0], u[1], \ldots, u[T]$, we construct a regression problem by computing the discretized target:

$$d[s] = \frac{r[s+1] - (1-\alpha)r[s]}{\alpha} = \phi(\theta x[s]) + \epsilon_{\text{conv}} \tag{S69}$$

where $x[s] = [r[s], u[s]]$ concatenates firing rates and inputs, and $\theta = [W^{\text{rec}}, W^{\text{in}}]$ are the parameters to be estimated. This transforms the temporal dynamics problem into a standard supervised learning task: predict $d[s]$ from $x[s]$ for all timesteps.

Optimization methods: We employ three primary approaches for parameter estimation:

1. CORNN algorithm: Our primary method uses the CORNN algorithm Dinc et al. (2023), which employs an iterative update scheme with fixed point initialization computed via ridge regression on $z[s] = \text{arctanh}(d[s])$. We implement three loss variants: weighted loss (dividing prediction errors by $1 - d^2$ to account for tanh saturation), standard L2 loss, and derivative-weighted loss (multiplying by $1 - \hat{d}^2$). The algorithm includes outlier detection based on a threshold parameter (typically 0.2 for trained networks, 1.0 for chaotic networks). Convergence criteria: (1) $\sqrt{N_{\text{rec}}} \cdot \sqrt{\text{mean}((\theta^{k+1} - \theta^k)^2)} < 10^{-5}$ after at least 10 iterations, or (2) maximum iterations reached (100-2000 depending on experiment complexity).

2. FORCE learning: In selected experiments with chaotic networks, we implement recursive least squares (RLS) FORCE learning Sussillo & Abbott (2009). FORCE updates parameters online using rank-one updates to the inverse covariance matrix, minimizing either current errors (pre-nonlinearity) or firing rate errors (post-nonlinearity). We use regularization parameters $\lambda = 100$ for recurrent weights and run the algorithm for up to 1000 iterations.

3. Gradient-based optimization: For comparison in selected experiments, we use PyTorch-based gradient descent with Adam optimizer (learning rate 10^{-3}, up to 10^4 iterations). Parameters are optionally initialized from the fixed point solution. This approach uses either Binary Cross-Entropy (BCE) loss or Mean Squared Error (MSE) loss, with L2 regularization applied through weight decay.

Regularization: The L2 regularization parameter λ ranges from 10^{-23} to 10^{-1} depending on the experiment, with typical values around 10^{-15} to 10^{-13} for chaotic networks and 10^{-13} to 10^{-5} for trained networks. In CORNN, regularization is scaled by the number of data points T.

Experimental variations: We perform parameter recovery on both chaotic RNNs and trained networks performing the three tasks (3-bit flip-flop, DCD, and DMTS). For experiments with external inputs (trained task networks), we concatenate $u[s]$ with firing rates in $x[s]$; for chaotic networks without inputs, we set $u[s] = 0$. Detailed configurations are provided in Section D.2.6.

D.2.6 Experimental Parameters by Figure

Figures 2, S1, S2: We generate chaotic dynamics from randomly initialized RNNs without external inputs, where network parameters are sampled from $\mathcal{N}(0, g/\sqrt{N_{\text{rec}}})$ with chaos parameter $g = 2$ and initial firing rates are sampled uniformly from $[-1, 1]$. We set the discretization parameter $\alpha = 0.1$ and iterate the network dynamics with no input ($u(t) = 0$), no input noise ($\epsilon_{\text{in}} = 0$), and no conversion noise ($\epsilon_{\text{conv}} = 0$). We systematically vary network size across $N_{\text{rec}} \in \{100, 300, 500, 1000\}$ and trajectory length across $T \in \{100, 300, 500, 1000, 1500, 2000, 2500, 3000\}$ timesteps, repeating all experiments across 20 random seeds (seeds 0-19). We employ the CORNN algorithm with weighted loss to recover parameters from the observed trajectories, setting regularization as $\lambda = T \times 10^{-15}$ (scaling linearly with trajectory length), fixed point initialization, update step size $\gamma = 1$, outlier threshold 1.0 (appropriate for chaotic networks), discretization parameter $\alpha = 0.1$ (matching the ground truth), and maximum iterations of 100 with convergence checking enabled, where training terminates when $\sqrt{N_{\text{rec}}} \cdot \sqrt{\text{mean}((\theta^{k+1} - \theta^k)^2)} < 10^{-5}$ after at least 10 iterations.

Figures 2, S1, S3, S4: Following the same setup as Figures 2, S1, S2, we fix the network size at $N_{\text{rec}} = 500$ and systematically vary the regularization parameter across $\lambda \in \{10^{-23}, 10^{-21}, 10^{-19}, 10^{-17}, 10^{-15}, 10^{-13}, 10^{-11}, 10^{-9}, 10^{-7}, 10^{-5}, 10^{-3}, 10^{-1}\}$ to examine the bias-variance tradeoff in parameter recovery across different trajectory lengths.

Figures S3, S4: Following the same setup as Figures 2, S1, S3, S4, we introduce noise to the chaotic dynamics by setting input noise $\epsilon_{\text{in}} = 10^{-3}$ and conversion noise $\epsilon_{\text{conv}} = 10^{-3}$ with Laplace distribution. We vary the regularization parameter across $\lambda \in \{10^{-15}, 10^{-7}, 10^{-5}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ to examine parameter recovery robustness under noisy conditions.

Figure 3: Following the same setup as Figures 2, S1, S2 with fixed network size $N_{\text{rec}} = 500$, we compare CORNN and FORCE learning algorithms for parameter recovery. We vary trajectory length across $T \in \{1000, 2000\}$ timesteps and regularization parameter across $\lambda \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. For FORCE learning, we use recursive least squares with regularization $\lambda_{\text{FORCE}} = \lambda \times 10^5$, input scaling $g_{\text{in}} = 3$, and maximum iterations of 100. FORCE minimizes errors in the predicted firing rates (i.e., post-nonlinearity outputs) rather than pre-nonlinearity currents, directly matching the observable neural activity at each timestep.

Figures 4, S5: We use trained 3-bit flip-flop networks ($N_{\text{rec}} = 500$, trained as described in the Trained Networks section) and generate trajectories with the trained parameters. We set discretization parameter $\alpha = 0.5$, introduce Poisson-distributed input noise $\epsilon_{\text{in}} = 10^{-2}$ and conversion noise $\epsilon_{\text{conv}} = 10^{-2}$, and vary the number of trials $T_{\text{trial}} \in \{5\}$ (each trial contains 100 timesteps). We augment the training data with n_{int} intervention samples using spectral vectors (top, bottom, or random eigenvectors of the covariance matrix), where n_{int} ranges from 10 to 500. We employ the CORNN algorithm with weighted loss, fixed regularization $\lambda = 10^{-6}$, outlier threshold 0.2 (appropriate for trained networks), discretization parameter $\alpha = 0.5$, maximum iterations of 2000 with convergence checking enabled, and repeat experiments across 20 random seeds (seeds 0-19).

Figure S6: We use trained DMTS networks ($N_{\text{rec}} = 300$, trained as described in the Trained Networks section) and generate trajectories with the trained parameters. We set discretization parameter $\alpha = 0.5$, input dimension $N_{\text{in}} = 2$ (including bias), Poisson-distributed input noise $\epsilon_{\text{in}} = 10^{-2}$ and conversion noise $\epsilon_{\text{conv}} = 10^{-3}$, and use 10 trials (each trial contains 38 timesteps covering input, delay, match, and response intervals). We compare four parameter recovery methods: CORNN with standard L2 loss (with outlier detection), CORNN with weighted loss, PyTorch gradient descent with BCE loss, and PyTorch with MSE loss. For CORNN methods, we use regularization $\lambda \in \{10^{-13}, 10^{-5}\}$, outlier threshold 0.2, and maximum iterations of 2000. For PyTorch methods, we use learning rate 10^{-2}, maximum iterations of 10^5, and the same regularization values. All experiments are repeated across 20 random seeds (seeds 0-19).

Figure S7: We use trained DCD networks ($N_{\text{rec}} = 500$, trained as described in the Trained Networks section) and generate trajectories with the trained parameters. We set discretization parameter $\alpha = 0.5$, input dimension $N_{\text{in}} = 2$ (including bias), Poisson-distributed input noise $\epsilon_{\text{in}} = 10^{-2}$ and conversion noise $\epsilon_{\text{conv}} = 10^{-3}$. We vary the number of trials across $T_{\text{trial}} \in \{5, 10, 30\}$ (each trial contains 32 timesteps covering input, delay, and response intervals) and examine parameter recovery with both uncorrelated and spatially-temporally correlated noise patterns (using 2D Gaussian kernel convolution with $\sigma_T = 3$ for temporal correlation and $\sigma_N = 50$ for spatial correlation). We employ the CORNN algorithm with weighted loss and outlier detection, regularization $\lambda \in \{10^{-13}, 10^{-7}, 10^{-5}, 10^{-3}, 10^{-1}\}$, outlier threshold 0.2, and maximum iterations of 2000 with convergence checking enabled, repeating experiments across 20 random seeds (seeds 0-19).

Figure S9: We use trained 3-bit flip-flop networks ($N_{\text{rec}} = 500$, trained as described in the Trained Networks section) and examine parameter recovery through spectral analysis. We set discretization parameter $\alpha = 0.5$, Poisson-distributed input noise $\epsilon_{\text{in}} = 10^{-3}$ and conversion noise $\epsilon_{\text{conv}} = 10^{-1}$, and vary the number of trials across $T_{\text{trial}} \in \{5, 10, 30\}$ (each trial contains 100 timesteps). We employ the CORNN algorithm with weighted loss, regularization $\lambda \in \{10^{-13}, 10^{-7}, 10^{-5}, 10^{-3}, 10^{-2}, 10^{-1}\}$, outlier threshold 0.5, and maximum iterations of 2000 with convergence checking enabled, repeating experiments across 20 random seeds (seeds 0-19). We analyze parameter recovery quality by projecting onto individual eigenvectors of the input covariance matrix.

Figure S10: We use trained DCD networks with a large network size ($N_{\text{rec}} = 10000$, trained as described in the Trained Networks section) and examine parameter recovery under partial observability, where we only observe $N_{\text{obs}} \in \{100, 300, 500, 1000, 2000\}$ neurons from the full network. We set discretization parameter $\alpha = 0.5$, input dimension $N_{\text{in}} = 2$ (including bias), Poisson-distributed input noise $\epsilon_{\text{in}} = 10^{-2}$ and conversion noise $\epsilon_{\text{conv}} = 10^{-3}$, and use 150 trials (each trial contains

32 timesteps). We employ the CORNN algorithm with weighted loss and outlier detection, fixed regularization $\lambda = 10^{-13}$, outlier threshold 0.2, and maximum iterations of 2000 with convergence checking enabled, repeating experiments across 20 random seeds (seeds 0-19).

Figure S11: We use trained 3-bit flip-flop networks ($N_{\text{rec}} = 500$, trained as described in the Trained Networks section) and examine parameter recovery quality through performance metrics. We set discretization parameter $\alpha = 0.5$, Poisson-distributed input noise $\epsilon_{\text{in}} = 10^{-2}$ and conversion noise $\epsilon_{\text{conv}} = 10^{-2}$, and use 5 trials for training (each trial contains 100 timesteps). We employ the CORNN algorithm with weighted loss, varying regularization across $\lambda \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\}$, outlier threshold 0.5, and maximum iterations of 2000 with convergence checking enabled, repeating experiments across 20 random seeds (seeds 0-19). We test recovered parameters on a separate test set of 1000 trials and evaluate both reconstruction accuracy via spectral projections and task performance accuracy on the flip-flop task.