DISCOVERING ALTERNATIVE SOLUTIONS BEYOND THE SIMPLICITY BIAS IN RECURRENT NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Training recurrent neural networks (RNNs) to perform neuroscience-style tasks has become a popular way to generate hypotheses for how neural circuits in the brain might perform computations. Recent work has demonstrated that task-trained RNNs possess a strong simplicity bias. In particular, this inductive bias often causes RNNs trained on the same task to collapse on effectively the same solution, typically comprised of fixed-point attractors or other low-dimensional dynamical motifs. While such solutions are readily interpretable, this collapse proves counterproductive for the sake of generating a set of genuinely unique hypotheses for how neural computations might be performed. Here we propose Iterative Neural Similarity Deflation (INSD), a simple method to break this inductive bias. By penalizing linear predictivity of neural activity produced by standard task-trained RNNs, we find an alternative class of solutions to classic neuroscience-style RNN tasks. These solutions appear distinct across a battery of analysis techniques, including representational similarity metrics, dynamical systems analysis, and the linear decodability of task-relevant variables. Moreover, these alternative solutions can sometimes achieve superior performance in difficult or out-of-distribution task regimes. Our findings underscore the importance of moving beyond the simplicity bias to uncover richer and more varied models of neural computation.

1 INTRODUCTION

Developing recurrent models of neural computations has become an increasingly popular approach to generate hypotheses for neuroscience (Mante et al., 2013; Rajan et al., 2016; Maheswaranathan et al., 2019; Yang et al., 2019; Sylwestrak et al., 2022; Daie et al., 2023; Beiran et al., 2023; Nair et al., 2023; Driscoll et al., 2024; Javadzadeh et al., 2024; Genkin et al., 2025). In particular, recurrent neural networks (RNNs) trained on neuroscience-style tasks offer insight into possible solutions that may be implemented at an approximate level by biological neural circuits. Such RNNs are typically trained via backpropagation through time (Werbos, 1990) or FORCE (Sussillo & Abbott, 2009), methods that seem to bear little resemblance to the way learning proceeds in biological circuits (Crick, 1989; Lillicrap et al., 2020). Nonetheless, resemblances between solutions found by artificial and biological networks have the potential to shed light on shared principles of neural computation that emerge despite these differences (Mante et al., 2013; Yamins et al., 2014; Sussillo et al., 2015; Kell et al., 2018; Banino et al., 2018; Schrimpf et al., 2020; Feather et al., 2023; Jensen et al., 2024; Pagan et al., 2025).

Central to this research program is the ability to produce multiple competing hypotheses that can then be evaluated on equal footing via comparisons against experimental data (Barak et al., 2013; Sussillo et al., 2015; Soldado-Magraner et al., 2024; Pagan et al., 2025; Huang et al., 2025). Ideally, training multiple RNNs on a particular task would be sufficient to yield a diverse range of solutions for this purpose. Yet, this strategy faces major obstacles in scenarios where training procedures overwhelmingly bias RNNs towards particular kinds of solutions.

Recent work has shown that task-trained RNNs exhibit a bias towards simple solutions—solutions that use a minimal arrangement of low-dimensional dynamical structures such as fixed point attractors and limit cycles, and reuse dynamical motifs where possible (Turner & Barak, 2023; Driscoll et al., 2024; Hazelden et al., 2025). These types of solutions have desirable properties including

parsimony and flexibility, and often lend themselves to relatively straightforward interpretation via analysis techniques such as targeted dimensionality reduction and dynamical systems analysis (Sussillo & Barak, 2013; Mante et al., 2013; Khona & Fiete, 2022; Driscoll et al., 2024). However, for many neuroscience-style tasks, this simplicity bias can be strong enough to cause different networks trained on the same task to collapse to effectively the same, minimal solution, a phenomenon referred to as dynamic collapse (Hazelden et al., 2025). Despite the desirable properties of such solutions, it remains far from clear that this bias towards simplicity is always aligned with the inductive biases of biological circuits. For example, RNNs trained on simple memory tasks ubiquitously find solutions using persistent activity held in stable attractor states (Maheswaranathan et al., 2019; Turner & Barak, 2023; Driscoll et al., 2024; Hazelden et al., 2025), yet population-level recordings have shown that the neural representations underlying memory functions can be highly dynamic (Spaak et al., 2017; Lundqvist et al., 2018; Daie et al., 2023; Ritter & Chadwick, 2025). These observations raise an important question: how can RNNs be trained to generate unique hypotheses for recurrent computations that go beyond the simplicity bias?

The most natural toolkit for generating different task solutions includes varying hyperparameters such as the initialization scale, training seed, and model architecture. The initialization scale in particular has been shown to affect lazy versus rich learning in RNNs (Schuessler et al., 2020; Liu et al., 2023; Bordelon et al., 2025), as well as the emergence of "aligned" or "oblique" solutions (Schuessler et al., 2024). However, dynamic collapse can still be observed even when RNNs are initialized in the highly chaotic regime (Hazelden et al., 2025). While varying these basic knobs is sometimes sufficient to generate a multitude of qualitatively distinct solutions, (Turner et al., 2021; Huang et al., 2025; Murray, 2025; Kurtkaya et al., 2025), many classes of realistic solutions are likely still inaccessible through these means. For instance, Pagan et al. (2025) found that a large population of RNNs trained on the same context-dependent decision making task populated only one corner of the solution space compatible with neural data. Moreover, solutions obtained by varying architectural details can appear representationally distinct, but often implement the same underlying dynamical solution, as revealed by fixed-point topology (Maheswaranathan et al., 2019).

In this paper, we propose a simple method for generating unique solutions to RNN tasks, extending beyond solutions discoverable by standard means. This method, which we call Iterative Neural Similarity Deflation (INSD), is loosely analogous to the Gram-Schmidt procedure but in the space of RNN solutions. By iteratively penalizing the linear predictivity of neural activity produced by previously trained RNNs in an online fashion, we find solutions that diverge from the prototypical solutions to classic neuroscience-style tasks. We show that the alternative solutions generated in this manner not only use distinct representational geometry as expected, but also use different dynamical motifs and encode task variables more nonlinearly. Across all tasks, these solutions forgo the usage of fixed point attractors and slow manifolds for keeping track of task-relevant information, and instead tend to maintain task-relevant information in dynamically evolving subspaces of activity. Surprisingly, we find that these alternative solutions can sometimes achieve superior performance when tested in difficult out-of-distribution task conditions.

2 Methods

2.1 Setup and Training Procedures

We consider rate-based RNNs obeying the dynamics

$$\frac{d\boldsymbol{x}}{dt} = -\boldsymbol{x} + \boldsymbol{W}\boldsymbol{r} + \boldsymbol{J}^{\text{in}}\boldsymbol{u}(t) \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ represent neural activations over N units, $\boldsymbol{W} \in \mathbb{R}^{N\times N}$ is the recurrent weight matrix, $\boldsymbol{J}^{\text{in}} \in \mathbb{R}^{N\times N_{in}}$ and $\boldsymbol{u}(t) \in \mathbb{R}^{N_{in}}$ are the input weights and inputs, respectively, $\boldsymbol{r} = \phi(\boldsymbol{x})$ are the "firing rates", and ϕ is an elementwise nonlinearity which we take to be $\tanh$. The output is given by $\boldsymbol{y}(t) = \boldsymbol{J}^{\text{out}}\boldsymbol{r}(t)$, for readout weights $\boldsymbol{J}^{\text{out}} \in \mathbb{R}^{N_{out}\times N}$.

For each task, we first train a reference RNN to minimize the mean squared error

$$\mathcal{L} = \frac{1}{T}\int_0^T \|\boldsymbol{y}(t) - \boldsymbol{y}^\star(t)\|^2 dt, \tag{2}$$

averaged over different input conditions $\boldsymbol{u}(t)$, via batch gradient descent over the parameters $\Theta = \{\boldsymbol{W}, \boldsymbol{J}^{\text{in}}, \boldsymbol{J}^{\text{out}}\}$. We initialize the recurrent weights as $\boldsymbol{W}_{ij} \sim \mathcal{N}(0, g^2/N)$, where g is a gain parameter. The input and output weights are both initialized with entries drawn from $\mathcal{N}(0, 1/N)$.

We then apply a neural activity similarity penalty to subsequent RNNs trained on the same task. In particular, for each batch of input conditions, firing rates $\boldsymbol{R}_1 \in \mathbb{R}^{(BL_t)\times N}$ and $\boldsymbol{R}_2 \in \mathbb{R}^{(BL_t)\times N}$ are collected from the reference RNN and the second RNN, respectively, where the batch (B) and discrete timestep (L_t) dimensions have been flattened. These firing rates are then projected into their respective readout null spaces, yielding $\boldsymbol{R}_1^{\perp}$ and $\boldsymbol{R}_2^{\perp}$. The second RNN is then trained with the loss

$$\mathcal{L}' = \mathcal{L} + \lambda S(\boldsymbol{R}_2^{\perp}, \boldsymbol{R}_1^{\perp}), \tag{3}$$

where S is some neural similarity measure, and λ is a hyperparameter representing the strength of the penalty. We project firing rates to readout nullspaces prior to applying the similarity penalty because allowing it to operate on the output potent component of activity would be counterproductive to solving the task. In particular, if the reference RNN achieves near perfect outputs $\boldsymbol{y}(t) \approx \boldsymbol{y}^{\star}(t)$, then to achieve similar task performance, the second RNN's activity must necessarily be able to linearly predict the output potent component of the reference RNN's activity. This procedure can be continued iteratively, with a third RNN penalized with respect to both previous RNNs via a loss

$$\mathcal{L}'' = \mathcal{L} + \lambda \left[S(\boldsymbol{R}_3^{\perp}, \boldsymbol{R}_1^{\perp}) + S(\boldsymbol{R}_3^{\perp}, \boldsymbol{R}_2^{\perp}) \right], \tag{4}$$

and so on. We refer to this procedure as Iterative Neural Similarity Deflation (INSD), and label RNNs trained in this manner alt-1, alt-2, etc. This approach for explicitly encouraging different task solutions somewhat resembles the Barlow Twins method (Zbontar et al., 2021) in computer vision and the method of linear adversarial concept erasure (Ravfogel et al., 2022) in algorithmic fairness.

For comparison, we also train a population of "standard" RNNs on each task. For simplicity, we use the same architecture for all RNNs, training ten RNNs with different seeds for each initialization scale $g \in [0.01, 0.5, 1.0, 1.5]$. A more detailed sweep including architecture, hyperparameters, and nonlinearities can be found in Maheswaranathan et al. (2019). Training details are specified in A.1.

2.2 Neural Similarity measures

There exists a large variety of neural similarity measures that could be used for the similarity penalty, each with their own advantages and drawbacks (Raghu et al., 2017; Kornblith et al., 2019; Williams et al., 2021; Harvey et al., 2024a; Williams, 2024; Cloos et al., 2024; Harvey et al., 2024b). For our purposes, we seek a metric which is invariant to relabeling or rotation of neural axes, and for which forwards and backwards passes can be efficiently computed online.

For many neural similarity measures, solving a task while maintaining low neural similarity with respect to a reference network admits a trivial yet undesirable solution: a subspace of activity implements a version of the reference solution, while the remaining degrees of freedom simply inflate the dimensionality of the neural activity with task-irrelevant dynamics. In particular, centered kernel alignment, representational similarity analysis (RSA), and linear predictivity scores in the direction of [reference RNN $\rightarrow$ penalized RNN] can all be driven arbitrarily close to 0 in this manner (see A.2). To avoid this solution, we use linear predictivity in the opposite direction [penalized RNN $\rightarrow$ reference RNN] as the similarity penalty. This asymmetry between predictivity and predictability in their sensitivity to irrelevant dynamics has also been noted in recent work on latent variable models of neural activity (Versteeg et al., 2024; Dabholkar & Barak, 2025). We remark that canonical correlation analysis (Hotelling, 1936; Raghu et al., 2017) can also avoid this undesirable solution, although the extra whitening step incurs a slight additional computational cost.

We define linear predictivity as $r^2(\boldsymbol{X}, \boldsymbol{Y}) = 1 - \min_{\boldsymbol{M}\in\mathbb{R}^{N\times N}} \frac{\|\boldsymbol{X}\boldsymbol{M}-\boldsymbol{Y}\|^2}{\|\boldsymbol{Y}\|^2} = \frac{\|\boldsymbol{U}_X\boldsymbol{Y}\|^2}{\|\boldsymbol{Y}\|^2}$ where $\boldsymbol{U}_X = \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{+}\boldsymbol{X}^\top \in \mathbb{R}^{(BL_t)\times(BL_t)}$ projects to the column space of $\boldsymbol{X}$. As the input matrices are often rank-deficient in our usage, for numerical stability, we also add a small ridge regularizer when computing the similarity penalty: $S(\boldsymbol{X}, \boldsymbol{Y}) = \frac{\|\boldsymbol{U}_{X,\rho}\boldsymbol{Y}\|^2}{\|\boldsymbol{Y}\|^2}$, where $\boldsymbol{U}_{X,\rho} = \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X} + \rho\boldsymbol{I})^{-1}\boldsymbol{X}^\top$.

2.3 Dynamical systems analysis

We probe the dynamical properties of task solutions via numerically solving for fixed points, as in (Sussillo & Barak, 2013). In line with previous studies (Sussillo & Barak, 2013; Maheswaranathan

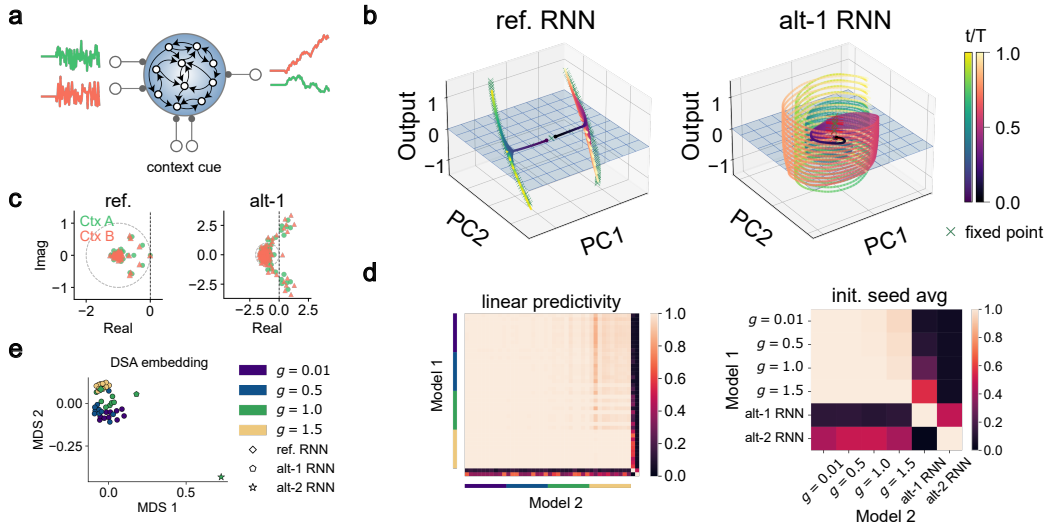

Figure 1: **Similarity-penalized RNNs yield distinct solutions to context-dependent integration.** **a.** Task schematic: two noisy stimuli are passed as input. In each trial, only the input stream selected by the context cue needs to be integrated, while the other is ignored. **b.** Example trajectories shown along the first two PCs and the output axis for the reference RNN (left) and alt-1 RNN (right), respectively. Trajectories are colored by time and relevant context (Ctx A: viridis, Ctx B: magma) during the corresponding trial. Fixed points (green x's) and unstable oscillatory leading eigenmodes (red bars) are shown. **c.** Representative examples of eigenvalue spectrums for Jacobians computed at fixed points found for the reference RNN (left) and alt-1 RNN (right). **d.** Left: Linear predictivity matrix across RNNs at different initialization scales and seeds trained on the task, along with the alt-1 and alt-2 RNNs. Right: same, but with scores for the standard RNNs averaged over initialization seed. **e.** MDS embedding of the DSA dissimilarity matrix computed across the same RNNs as in **d**.

et al., 2019; Driscoll et al., 2024; Kurtkaya et al., 2025), we include approximate fixed points, also referred to as slow points. Where relevant, we also report the stability, eigenvalue spectrum and leading eigenmode(s) that govern the linearized dynamics in the vicinity of each fixed point.

3 Results

We analyze and compare similarity-penalized solutions across three neuroscience-style tasks that have been well studied in the literature (Barak et al., 2013; Mante et al., 2013; Maheswaranathan et al., 2019; Schuessler et al., 2020; Smith et al., 2021; Krause et al., 2022; Valente et al., 2022; Costacurta et al., 2024; Driscoll et al., 2024; Huang et al., 2025; Pagan et al., 2025). These tasks span context-dependent processing, discrete and analog memory, and delayed output production. Each of these tasks is associated with a prototypical solution that has been reported across multiple studies, which we briefly describe for each task. Task parameters are specified in A.3.

Context-dependent integration. We begin by studying RNNs trained on context-dependent integration (Fig. 1a). For this task, the network receives two streams of noisy input stimuli and a fixed context cue. For a short duration T_{pre}, only the one-hot encoded context cue is shown. Thereafter, the context cue remains on, while the noisy input stimuli are sampled independently at each timestep from $\mathcal{N}(\mu_i, \sigma^2/dt)$ (following the convention in (Mante et al., 2013; Schuessler et al., 2024)). For each trial, the stimuli coherences μ_i are sampled from $\mathcal{U}[-\mu_{max}, \mu_{max}]$. At each timestep, the network must output the cumulative sum (scaled by dt) of all inputs received so far in the stimulus channel selected by the context cue. RNNs trained on this task and its binary decision making variant have consistently been found to learn two lines of fixed points (line attractors), one for integrating the relevant stimulus in each context (Mante et al. (2013); Maheswaranathan et al. (2019); Smith et al. (2021); Krause et al. (2022); Pagan et al. (2025)).

To assess the properties of solutions, as in (Maheswaranathan et al., 2019), we first probe all trained networks using task trials of varying stimuli coherences, turning off stimuli noise for visual clarity.

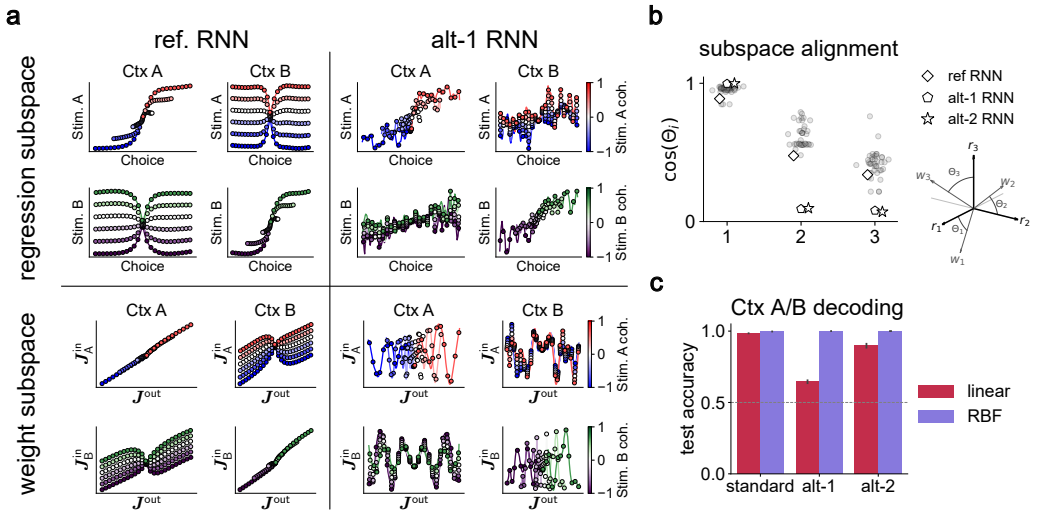

Figure 2: **Linear encoding of task-relevant information is degraded in similarity-penalized RNNs.** Task: context-dependent integration. **a.** Averaged trajectories plotted on different sets of axes, colored by the coherences of the input stimuli. Top row: axes directions estimated via predicting current target output (choice), stimulus A coherence, and stimulus B coherence via linear regression over neural activity aggregated over trials and time points. Bottom row: same averaged trajectories, but plotted on axes of the input and output weights. In each quadrant, left and right plots correspond to context A and B trials, respectively. Colorbars are normalized so that ± 1 corresponds to the minimum/maximum coherence value. **b.** Alignment between the regression and weight subspaces, as measured by the cosine of the principal angles. Grey dots represent alignments computed for the population of standard RNNs. **c.** Decodability of the relevant context from neural activity under linear or RBF kernel regression, as quantified by test accuracy on a heldout set. Error bars report the standard error of the mean. The grey dotted line represents the baseline accuracy.

In line with previous findings, we observed that all standard RNNs found the aforementioned prototypical solution, regardless of initialization scale and training seed. We illustrate this solution for a reference RNN in Fig. 1b (left), showing activity trajectories plotted on the axes of the first two principal components and the readout. During the context-only period, trajectories quickly segregate into separate regions of state space. Then, in each context, activity is driven along a line of approximate fixed points that densely tile the span of trajectories observed in that context. In contrast, similarity penalized RNNs yielded solutions characterized by oscillatory dynamics (Fig. 1, right). Activity in each context was readily distinguishable by the shape of trajectories, rather than the portion of state space they occupy. Moreover, activity was no longer driven along slow/fixed points. Instead, unstable fixed points with oscillatory eigenmodes were found, but were not used (at least directly) for remembering the cumulative input in either context. Comparing the eigenspectrums of the Jacobians at representative fixed points for both networks confirmed that marginally stable linearized dynamics were only present for fixed points of the reference RNN (Fig. 1c). For brevity, we defer the trajectory and eigenspectrum plots for the alt-2 RNN to the Appendix (Fig. A.1).

We compute linear predictivity scores in both directions between all pairs of models, including the population of standard RNNs and models produced by two iterations of INSD. We find that the representations used by standard RNNs are all highly linearly predictive of each other, with only slight deviations from perfect predictivity observed when predicting models of high initialization scale from models of lower initialization scale (Fig. 1d). Further, similarity penalized RNNs were markedly less predictive and less predictable with respect to standard solutions. To quantify relationships between the solutions beyond geometrical similarity, we also compute their Dynamical Similarity Analysis (DSA, Ostrow et al. (2023)) dissimilarity matrix, visualizing the scores via a multi-dimensional scaling embedding (Fig. 1e). This embedding reveals a degree of clustering by initialization scale. However, similarity-penalized solutions achieve a dynamical dissimilarity with respect to the standard population that far exceeds the scale of variability observed across clusters.

Next, we analyzed population responses via projecting activity trajectories onto task-relevant subspaces. For the reference and alt-1 RNNs, we first construct a regression-based subspace comprising

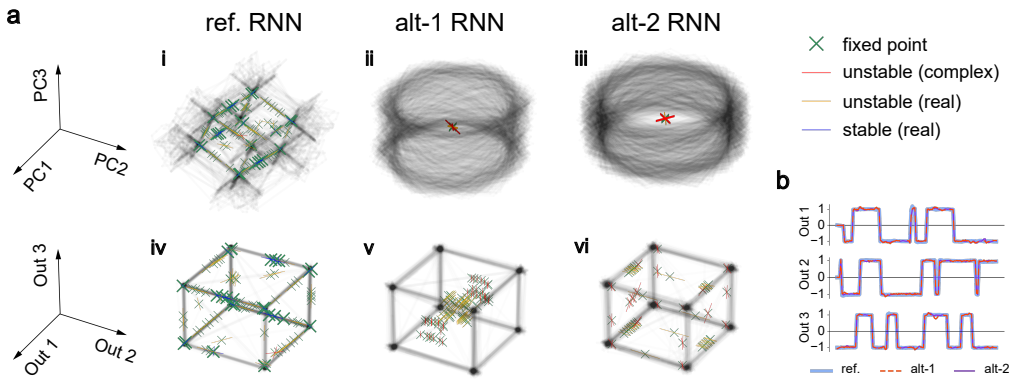

Figure 3: **Sustaining discrete memory states without fixed-point attractors.** Task: 3-bit flipflop. **a.** Example trajectories plotted on the principal component (**i, ii, iii**) and output (**iv, v, vi**) axes, shown for the reference (**i, iv**), alt-1 (**ii, v**), and alt-2 (**iii, vi**) RNNs . Fixed points (green x's) and their leading eigenmodes (colored bars) are shown. A larger marker size is used for stable fixed points. **b.** Example timeseries of network output for all three networks.

of the "stimulus A", "stimulus B", and "choice" axes. These directions were estimated via linearly regressing the coherences of stimulus A, stimulus B, and the task target, respectively, from neural activity aggregated across timesteps and 5000 trials. Consistent with prior studies (Mante et al., 2013; Smith et al., 2021; Pagan et al., 2025), projecting the averaged trajectories of standard RNNs onto this set of axes revealed a temporally stable and consistent encoding of the coherences of both input stimuli, regardless of the selected context (Fig. 2a, top left). In contrast, for the alt-1 RNN, the coherence of the relevant stimulus in each trial could still be linearly decoded somewhat consistently, but estimates of the irrelevant stimulus were often inconsistent with actual trial conditions (Fig. 2a, top right). We repeated these analyses, but for a weight-based subspace, projecting averaged trajectories onto the axes $[\boldsymbol{J}^{\text{in}}_A, \boldsymbol{J}^{\text{in}}_B, \boldsymbol{J}^{\text{out}}]$ defined by the input and output weights of each RNN. We again find that, for the standard solution, stimuli coherences for both relevant and irrelevant stimuli can be stably distinguished under these axes (Fig. 2a, bottom left). However, for the alt-1 RNN, the directions encoded by the input weights poorly captured the coherences of both stimuli, regardless of context (Fig. 2a, bottom right). To assess the relationship between the weight and regression subspaces, we quantified their alignment via computing the principal angles between them (Fig. 2b). Across all models, the leading overlap was near unity, likely due to the high alignment between the regression "choice" axis and $\boldsymbol{J}^{\text{out}}$ weight axis. Although the standard RNNs demonstrated varying degrees of moderate alignment between the remaining axes, these angles were near orthogonal for both the alt-1 and alt-2 RNNs. Finally, we assessed the extent to which task context—the most basic task variable—can be accurately decoded from activity. Consistent with the geometric picture of Fig. 1b, we find that context is linearly decodable at high accuracy for standard RNNs, whereas the alt-1 RNN (and to a lesser extent, alt-2 RNN) requires additional nonlinear featurization of representations for context to be decodable at similarly high accuracy (Fig. 2c).

3-bit flipflop. We next seek alternative solutions on 3-bit flipflop, a simple discrete memory task. For this task, three input channels are given. At each timestep, each channel independently has a probability p of having an upward or downward spike of magnitude $1/dt$, with both directions having equal probability. The target output for the network begins at 0 for all channels, and thereafter tracks the sign of the last spike in each channel. Trained RNNs consistently learn the most minimal and sensible solution: fixed point attractors arranged in a cube associated with each of the 8 main output states (aside from the starting outputs at 0), as well as saddle points whose unstable directions are aligned with edges of the cube to facilitate state transitions (Barak et al., 2013; Maheswaranathan et al., 2019; Ostrow et al., 2023). We plot trajectories of solutions as well as fixed points for a reference, alt-1 and alt-2 RNN trained on this task. We confirm that the reference RNN indeed learns the standard solution involving the cube of stable fixed points, and saddle points that transition between them (Fig. 3a,i). Moreover, the geometrical structure of activity in PCA space is minimal in the sense that it mirrors the cube-like geometry of the task output. For the similarity penalized RNNs, however, observed trajectories no longer show this geometry in PCA space, and instead follow oscillations generated by unstable fixed points with complex leading eigenmodes (Fig. 3a,ii,iii).

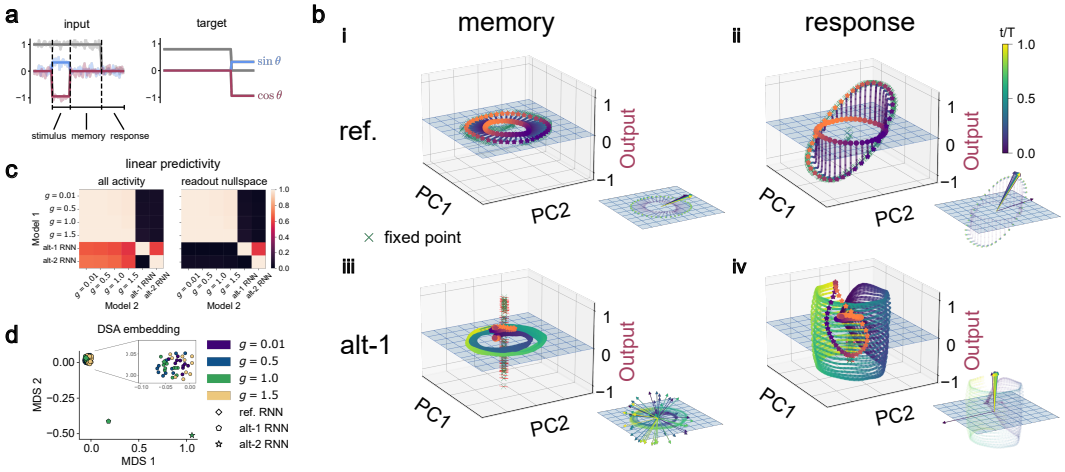

Figure 4: **Similarity-penalized RNNs find dynamic, rather than persistent, encoding of analog memories. a.** Task: MemoryPro. an angle encoded as a 2D vector is passed as input during a stimulus phase, followed by a memory phase where the angle input is absent. Only once the fixation cue (grey) is removed must the network output the angle that was observed. All inputs are noisy. **b.** Example trajectories divided by the memory (**i, iii**) and response (**ii, iv**) phases, shown for the reference RNN (**i, ii**) and alt-1 RNN (**iii, iv**). All plots are along the first two memory phase PCs and the cos output axis, with trajectories colored by time. The start and end of every trajectory is colored by the target output angle. Fixed points (green x's) and unstable oscillatory eigenmodes (red bars) are shown. The right corner of each subplot shows the direction encoding the target angle over time, as estimated by linear regression. **c.** Linear predictivity matrices comparing standard RNNs at different initialization scales to similarity penalized RNNs (alt-1, alt-2). Scores involving standard RNNs are averaged with respect to the initialization seed. Left: base linear predictivity scores. Right: linear predictivity scores when activity is first projected to the readout nullspace. **d.** MDS embedding of the DSA similarity matrix computed across the same RNNs as in **c**.

Despite these apparent differences in representational geometry, all three networks must ultimately produce cube-like geometry when trajectories are projected onto the output subspace; this is demanded by the structure of the target output of the task. Thus, to compare the solutions found more aptly, we also plot trajectories and fixed points on the output axes of each RNN. This reveals that, even in the output subspace, similarity-penalized RNNs exhibit distinct arrangements and stability properties of fixed points. In this example, the alt-1 RNN lacks fixed points that stabilize any of the output states, instead showing two groups of unstable fixed points with oscillatory eigenmodes, and saddle points that appear to transition between them (Fig. 3a,v). The alt-2 RNN recovers the presence of fixed points at each corner, but they are no longer stable/attractive (Fig. 3,vi). Moreover, the directions of saddle points that line the edges of the cube are often misaligned. These differences in dynamical motifs manifest as slight but noticeable imperfections in the output produced by the similarity-penalized RNNs (Fig. 3b). We also assess the similarity of representations across all RNNs using linear predictivity and DSA (Fig. A.2). Similar to the findings for context-dependent integration, all standard solutions are found to be perfectly linear predictive of each other, whereas the similarity-penalized RNNs occupy disparate areas of the DSA MDS embedding.

MemoryPro. Lastly, we turn our attention to the MemoryPro task (Fig. 4a). The RNN receives three piecewise constant inputs: a fixation cue and 2 stimuli channels encoding an angle. For each trial, the angle θ is sampled from $\mathcal{U}[-\pi, \pi]$. Following Driscoll et al. (2024), at train time, stimuli and response onsets and offsets are variable. Specifically, after a delay of length $T_{\text{del}} \sim \mathcal{U}[T_{\text{del}}^-, T_{\text{del}}^+]$, the angle stimuli $(\sin\theta \;\; \cos\theta)^\top$ are shown for a duration $T_{\text{stim}} \sim \mathcal{U}[T_{\text{stim}}^-, T_{\text{stim}}^+]$. Then, the stimuli are turned off for a duration $T_{\text{mem}} \sim \mathcal{U}[T_{\text{mem}}^-, T_{\text{mem}}^+]$, following which the fixation cue is removed and the response period begins. For a duration $T_{\text{resp}} \sim \mathcal{U}[T_{\text{resp}}^-, T_{\text{resp}}^+]$, the network must output the angle seen during the stimuli phase, also as a 2D vector. The network must also produce an output that tracks the fixation cue. All three inputs are also subjected to independent noise at each timestep, drawn from $\mathcal{N}(0, \sigma^2)$. Previous studies consistently report the following prototypical solution: during the memory phase, angles are encoded along a ring manifold of persistent states in the output nullspace,

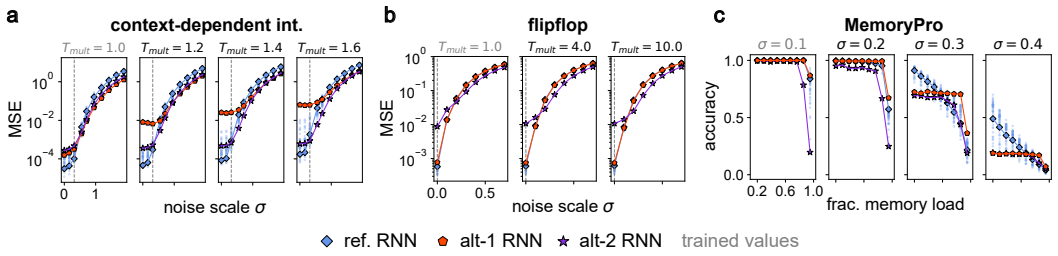

Figure 5: **Similarity penalized models can outperform standard models in difficult task regimes. a, b.** Mean squared error on the context-dependent integration (**a**) and flipflop (**b**) tasks, respectively, across different noise scales σ and trial length scaling T_{mult}. The grey dotted lines indicate the noise scale used during training. **c.** Accuracy on the MemoryPro task versus fractional memory load, at different noise scales σ. We assess accuracy using the same criteria as in (Driscoll et al., 2024). Small blue dots represent the scores achieved by the population of standard RNNs.

stabilized by a ring attractor. During the response phase, this ring of fixed points quickly rotates to become output potent (Driscoll et al., 2024; Costacurta et al., 2024; Hazelden et al., 2025).

To probe the properties of solutions, we plot trajectories collected over trials with various target angles and in the absence of input noise (Fig. 4b). As in Driscoll et al. (2024), we use the axes of the first two memory phase PCs and the $\cos\theta$ output channel, separating trajectories by the memory and response phases. Our results confirm that standard RNNs ubiquitously find the prototypical solution involving a ring attractor that rotates outwards, shown for the reference network (Fig. 4b,i,ii). We also plot the direction in activity space that best predicts the target angle via linear regression at each timestep, confirming that memorized angles are statically encoded (Fig. 4b,i,ii, bottom right). In contrast, the alt-1 RNN exhibits rotational dynamics during the memory phase that nonetheless maintains the relative ordering of trajectories by their corresponding target output (Fig. 4b, iii). The ring of fixed points is no longer present, and is instead replaced by a line of unstable fixed points with oscillatory leading eigenmodes. Linear decoding analysis reveals that the direction encoding the target angle is indeed rotating with the activity (Fig. 4b, iii, bottom right). Moreover, this direction even acquires output potency at times, despite the fact that the output potent component is, by task necessity, a low-variance fraction of the activity during the memory phase. During the response phase, these trajectories continue to oscillate, but rotate to become output potent (Fig. 4b, iv). We defer the corresponding plots for the alt-2 RNN to the Appendix (Fig. A.3).

As done for previous tasks, we compute linear predictivity and DSA dissimilarity scores between all pairs of models across the standard and similarity-penalized RNNs. While the linear predictivity of standard solutions from similarity-penalized solutions is degraded, we find that it is still significantly above zero (Fig. 4c, left). However, this partial predictivity is ablated once activities are projected into their respective readout nullspaces. This indicates that the only component of activity that the similarity penalized models can predict from standard solutions is that which is necessary to solve the task, namely, the output potent component. An MDS embedding of the DSA dissimilarity matrix confirms that the similarity-penalized RNNs achieve dynamically dissimilar solutions (Fig. 4d).

Assessing solutions by their performance under atypical task conditions. Across all three tasks, we found solutions that appear distinct from standard solutions by a variety of measures. However, are these solutions actually functionally distinct, or are they merely approximating the standard solution in ways that are difficult to discern? To answer this question, we tested all models under task conditions seldom or never seen during training. For the context-dependent integration task, we measured task performance across different noise scales σ of the input stimuli. We also introduce and sweep over the parameter T_{mult}, a factor that uniformly scales the duration of trials. We conduct a similar performance sweep for the flipflop task. For the MemoryPro task, we again sweep the input noise scale σ, but also sweep the fractional memory load, which we define as $\frac{T_{\text{mem}}}{T_{\text{stim}}+T_{\text{mem}}}$. To tune this parameter, we fix the duration of the pre-stimulus and response phases, as well as the total duration of the stimulus and memory phases combined. We then adjust the timing of the transition from the stimulus to the memory phase to produce test trials of varying fractional memory loads.

We report model performance across these sweeps in Fig. 5, as well as the corresponding effective dimensionality of activity as measured by the participation ratio in Fig. A.4. Across all tasks, we find

that standard RNNs typically outperform similarity-penalized RNNs when tested under conditions seen during training (Fig. 5). However, we also observe many cases where similarity-penalized RNNs outperform standard models. For instance, for the context-dependent integration task, the alt-2 RNN moderately outperforms the population of standard RNNs in highly noisy conditions, all the while remaining robust to lengthened trial durations. For the flipflop task, although we observe near-identical performance across most models, the alt-2 RNN achieves a moderate but significant gain in relative performance when noise is high. Lastly, for MemoryPro, we observe that the alt-1 RNN significantly outperforms standard RNNs on the most difficult trials, where both noise and memory load are high, but significantly underperforms the population under low memory loads. The alt-2 RNN only matched or underperformed the population under all conditions, suggesting that it simply failed to learn the task as well. Altogether, these performance deviations confirm that similarity-penalized models indeed produce solutions that are functionally distinct.

4 Discussion

Generating a rich set of diverse hypotheses that can be tested against experimental data is foundational for progressing our understanding of the brain. Motivated by recent observations of dynamic collapse in task-trained RNNs (Maheswaranathan et al., 2019; Driscoll et al., 2024; Hazelden et al., 2025), we propose a method called Iterative Neural Similarity Deflation (INSD) for expanding the space of accessible solutions. Across three neuroscience-style tasks, we extensively study and compare the solutions generated by iteratively penalizing the linear predictivity of past solutions. These analyses revealed alternative solutions that did not directly use simple dynamical motifs such as fixed point attractors or continuous slow manifolds to store information. Instead, similarity-penalized RNNs tended to produce activity characterized by quasi-periodic oscillatory modes. Further analysis revealed that these oscillations were not simply nuisance dynamics that emerged as a peculiarity of the similarity penalty, but rather actively supported the dynamic encoding of task-relevant information. These solutions are reminiscent of a theory proposed by Park et al. (2023) on how memories can be stably maintained in the phase difference between two oscillations, rather than through persistent attractor states. In the same vein, recent work by Ritter & Chadwick (2025) argues that optimally efficient and noise-robust working memory requires high-dimensional rotational dynamics, and further finds signatures of such dynamics in monkey prefrontal cortex. These observations are consistent with our finding of improved robustness for some similarity-penalized solutions.

For context-dependent integration, unlike similarity-penalized RNNs, standard RNNs produced solutions where task-relevant information was stably represented in linear subspaces, consistent with neural data recorded during analogous tasks (Mante et al., 2013; Pagan et al., 2025). Thus, a natural concern is that similarity-penalized RNNs may produce solutions whose population coding properties are not realistic. However, we argue that being able to also find unrealistic solutions is crucial for probing when and why simple solutions align with biology. Moreover, in principle, one could construct networks that interpolate between standard and similarity-penalized solutions. Most simply, this could be achieved by an RNN with two populations of neurons, one dedicated to implementing each solution. Much as how ensembling is used in machine learning to reduce variance and improve generalization, such mixed models may possibly enjoy greater robustness, all the while maintaining more realistic linear encoding properties at the population level. We leave a more detailed investigation of this idea to future work.

Finally, we acknowledge that linear predictivity is an imperfect measure of both dynamical similarity and functional equivalence (Ostrow et al., 2023; Qian et al., 2024; Braun et al., 2025). The recently proposed Dynamical Similarity Analysis (DSA, Ostrow et al. (2023)) has been shown to effectively identify RNN solutions whose dynamical properties are only superficially distinct, while other metrics often fall short. However, computing this metric as a similarity penalty in an online fashion would be prohibitively computationally expensive. Despite the limitations of linear predictivity, we found that penalizing the predictivity of representations used by standard RNNs was sufficient to generate solutions with distinct dynamical features and unique task performance profiles. While we did not perform an extensive sweep over the strength of the predictivity penalty here, solutions that differ from standard solutions in more fine-grained ways may potentially also be discoverable if the penalty is set to be small. For instance, for context-dependent decision making, a more granular form of solution degeneracy can be characterized in terms of the relative arrangements of input vectors and eigenvectors of the linearized dynamics across contexts (Pagan et al., 2025). Since the penalty

we use relates to neural activity, our approach likely cannot fully capture degeneracies in the relative arrangements between left eigenvectors and input vectors, but may be able to explore degeneracies in the relative arrangements of the right eigenvectors (integration directions) across contexts, which do directly manifest in activity.

A limitation of our study is that we focus on simple single-task settings where standard solutions invoke attractor dynamics. Future work should investigate tasks that require transient dynamics, such as timing tasks, where standard RNN solutions are already somewhat varied (Turner et al., 2021; Beiran et al., 2023; Huang et al., 2025). Experiments in multitask settings would also be insightful for understanding whether greater task demands make it more difficult to find solutions that are not linearly predictive of reference solutions (Cao & Yamins, 2024; Huang et al., 2025). More generally, this "capacity" for distinct solutions is expected to depend on task difficulty and network size, and could be systematically explored using INSD. Such experiments could serve as a way to empirically assess ideas such as the Contravariance Principle and the Platonic Representation Hypothesis (Cao & Yamins, 2024; Huh et al., 2024), both of which assert that sufficiently difficult or constraining task demands necessitate representational convergence across systems.

REFERENCES

Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J. Chadwick, Thomas Degris, Joseph Modayil, Greg Wayne, Hubert Soyer, Fabio Viola, Brian Zhang, Ross Goroshin, Neil Rabinowitz, Razvan Pascanu, Charlie Beattie, Stig Petersen, Amir Sadik, Stephen Gaffney, Helen King, Koray Kavukcuoglu, Demis Hassabis, Raia Hadsell, and Dharshan Kumaran. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433, May 2018. ISSN 1476-4687. doi: 10.1038/s41586-018-0102-6.

Omri Barak, David Sussillo, Ranulfo Romo, Misha Tsodyks, and L. F. Abbott. From fixed points to chaos: Three models of delayed discrimination. *Progress in Neurobiology*, 103:214–222, April 2013. ISSN 0301-0082. doi: 10.1016/j.pneurobio.2013.02.002.

Manuel Beiran, Nicolas Meirhaeghe, Hansem Sohn, Mehrdad Jazayeri, and Srdjan Ostojic. Parametric control of flexible timing through low-dimensional neural manifolds. *Neuron*, 111(5): 739–753.e8, March 2023. ISSN 0896-6273. doi: 10.1016/j.neuron.2022.12.016.

Blake Bordelon, Jordan Cotler, Cengiz Pehlevan, and Jacob A. Zavatone-Veth. Dynamically Learning to Integrate in Recurrent Neural Networks, March 2025.

Lukas Braun, Erin Grant, and Andrew M. Saxe. Not all solutions are created equal: An analytical dissociation of functional and representational similarity in deep linear neural networks. In *Forty-Second International Conference on Machine Learning*, June 2025.

Rosa Cao and Daniel Yamins. Explanatory models in neuroscience, Part 2: Functional intelligibility and the contravariance principle. *Cognitive Systems Research*, 85:101200, June 2024. ISSN 1389-0417. doi: 10.1016/j.cogsys.2023.101200.

Nathan Cloos, Moufan Li, Markus Siegel, Scott L. Brincat, Earl K. Miller, Guangyu Robert Yang, and Christopher J. Cueva. Differentiable Optimization of Similarity Scores Between Models and Brains. In *The Thirteenth International Conference on Learning Representations*, October 2024.

Julia C. Costacurta, Shaunak Bhandarkar, David M. Zoltowski, and Scott Linderman. Structured flexibility in recurrent neural networks via neuromodulation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024.

Francis Crick. The recent excitement about neural networks. *Nature*, 337(6203):129–132, January 1989. ISSN 1476-4687. doi: 10.1038/337129a0.

Kabir Dabholkar and Omri Barak. When predict can also explain: Few-shot prediction to select better neural latents, August 2025.

Kayvon Daie, Lorenzo Fontolan, Shaul Druckmann, and Karel Svoboda. Feedforward amplification in recurrent networks underlies paradoxical neural coding. *bioRxiv*, pp. 2023.08.04.552026, August 2023. doi: 10.1101/2023.08.04.552026.

Laura N. Driscoll, Krishna Shenoy, and David Sussillo. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *Nature Neuroscience*, 27(7):1349–1363, July 2024. ISSN 1546-1726. doi: 10.1038/s41593-024-01668-6.

Jenelle Feather, Guillaume Leclerc, Aleksander Madry, and Josh H. McDermott. Model metamers reveal divergent invariances between biological and artificial neural networks. *Nature Neuroscience*, 26(11):2017–2034, November 2023. ISSN 1546-1726. doi: 10.1038/s41593-023-01442-0.

Mikhail Genkin, Krishna V. Shenoy, Chandramouli Chandrasekaran, and Tatiana A. Engel. The dynamics and geometry of choice in the premotor cortex. *Nature*, 645(8079):168–176, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09199-1.

Matthew D. Golub and David Sussillo. FixedPointFinder: A Tensorflow toolbox for identifying and characterizing fixed points in recurrent neural networks. *Journal of Open Source Software*, 3(31): 1003, November 2018. ISSN 2475-9066. doi: 10.21105/joss.01003.

Sarah E. Harvey, Brett W. Larsen, and Alex H. Williams. Duality of Bures and Shape Distances with Implications for Comparing Neural Representations. In *Proceedings of UniReps: The First Workshop on Unifying Representations in Neural Models*, pp. 11–26, May 2024a.

Sarah E. Harvey, David Lipshutz, and Alex H. Williams. What Representational Similarity Measures Imply about Decodable Information. In *UniReps: 2nd Edition of the Workshop on Unifying Representations in Neural Models*, October 2024b.

James Hazelden, Laura Driscoll, Eli Shlizerman, and Eric Shea-Brown. KPFlow: An Operator Perspective on Dynamic Collapse Under Gradient Descent Training of Recurrent Networks, July 2025.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3-4):321–377, December 1936. ISSN 0006-3444. doi: 10.1093/biomet/28.3-4.321.

Ann Huang, Satpreet H. Singh, Flavio Martinelli, and Kanaka Rajan. Measuring and Controlling Solution Degeneracy across Task-Trained Recurrent Neural Networks, May 2025.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The Platonic Representation Hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 20617–20642, July 2024.

Mitra Javadzadeh, Marine Schimel, Sonja B. Hofer, Yashar Ahmadian, and Guillaume Hennequin. Dynamic consensus-building between neocortical areas via long-range connections, December 2024.

Kristopher T. Jensen, Guillaume Hennequin, and Marcelo G. Mattar. A recurrent network model of planning explains hippocampal replay and human behavior. *Nature Neuroscience*, 27(7):1340–1348, July 2024. ISSN 1546-1726. doi: 10.1038/s41593-024-01675-7.

Alexander J. E. Kell, Daniel L. K. Yamins, Erica N. Shook, Sam V. Norman-Haignere, and Josh H. McDermott. A Task-Optimized Neural Network Replicates Human Auditory Behavior, Predicts Brain Responses, and Reveals a Cortical Processing Hierarchy. *Neuron*, 98(3):630–644.e16, May 2018. ISSN 0896-6273. doi: 10.1016/j.neuron.2018.03.044.

Mikail Khona and Ila R. Fiete. Attractor and integrator networks in the brain. *Nature Reviews Neuroscience*, 23(12):744–766, December 2022. ISSN 1471-0048. doi: 10.1038/s41583-022-00642-0.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3519–3529, May 2019.

Renate Krause, Matthew Cook, Sepp Kollmorgen, Valerio Mante, and Giacomo Indiveri. Operative dimensions in unconstrained connectivity of recurrent neural networks. *Advances in Neural Information Processing Systems*, 35:17073–17085, December 2022.

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*, June 2025.

Timothy P. Lillicrap, Adam Santoro, Luke Marris, Colin J. Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, June 2020. ISSN 1471-0048. doi: 10.1038/s41583-020-0277-3.

Yuhan Helena Liu, Aristide Baratin, Jonathan Cornford, Stefan Mihalas, Eric Todd SheaBrown, and Guillaume Lajoie. How connectivity structure shapes rich and lazy learning in neural circuits. In *The Twelfth International Conference on Learning Representations*, October 2023.

Mikael Lundqvist, Pawel Herman, and Earl K. Miller. Working Memory: Delay Activity, Yes! Persistent Activity? Maybe Not. *Journal of Neuroscience*, 38(32):7013–7019, August 2018. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.2485-17.2018.

Niru Maheswaranathan, Alex Williams, Matthew Golub, Surya Ganguli, and David Sussillo. Universality and individuality in neural dynamics across large populations of recurrent networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

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, November 2013. ISSN 1476-4687. doi: 10.1038/nature12742.

Keith T. Murray. Phase codes emerge in recurrent neural networks optimized for modular arithmetic, July 2025.

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.e15, January 2023. ISSN 0092-8674, 1097-4172. doi: 10.1016/j.cell.2022.11.027.

Mitchell Ostrow, Adam Eisen, Leo Kozachkov, and Ila Fiete. Beyond Geometry: Comparing the Temporal Structure of Computation in Neural Circuits with Dynamical Similarity Analysis. *Advances in Neural Information Processing Systems*, 36:33824–33837, December 2023.

Marino Pagan, Vincent D. Tang, Mikio C. Aoi, Jonathan W. Pillow, Valerio Mante, David Sussillo, and Carlos D. Brody. Individual variability of neural computations underlying flexible decisions. *Nature*, 639(8054):421–429, March 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08433-6.

Il Memming Park, Ábel Ságodi, and Piotr Aleksander Sokół. Persistent learning signals and working memory without continuous attractors, August 2023.

William Qian, Jacob A. Zavatone-Veth, Benjamin S. 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, December 2024.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Kanaka Rajan, Christopher D. Harvey, and David W. Tank. Recurrent Network Models of Sequence Generation and Memory. *Neuron*, 90(1):128–142, April 2016. ISSN 0896-6273. doi: 10.1016/j.neuron.2016.02.009.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan D. Cotterell. Linear Adversarial Concept Erasure. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 18400–18421, June 2022.

Laura Ritter and Angus Chadwick. Efficient Working Memory Maintenance via High-Dimensional Rotational Dynamics. *bioRxiv*, pp. 2025.09.08.674838, September 2025. ISSN 2692-8205. doi: 10.1101/2025.09.08.674838.

Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J. Majaj, Rishi Rajalingham, Elias B. Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, Kailyn Schmidt, Daniel L. K. Yamins, and James J. DiCarlo. Brain-Score: Which Artificial Neural Network for Object Recognition is most Brain-Like? *bioRxiv*, pp. 407007, January 2020. doi: 10.1101/407007.

Friedrich Schuessler, Francesca Mastrogiuseppe, Alexis Dubreuil, Srdjan Ostojic, and Omri Barak. The interplay between randomness and structure during learning in RNNs. In *Advances in Neural Information Processing Systems*, volume 33, pp. 13352–13362, 2020.

Friedrich Schuessler, Francesca Mastrogiuseppe, Srdjan Ostojic, and Omri Barak. Aligned and oblique dynamics in recurrent neural networks. *eLife*, 13, October 2024. doi: 10.7554/eLife.93060.2.

Jimmy Smith, Scott Linderman, and David Sussillo. Reverse engineering recurrent neural networks with Jacobian switching linear dynamical systems. In *Advances in Neural Information Processing Systems*, volume 34, pp. 16700–16713, 2021.

Joana Soldado-Magraner, Valerio Mante, and Maneesh Sahani. Inferring context-dependent computations through linear approximations of prefrontal cortex dynamics. *Science Advances*, 10(51): eadl4743, December 2024. doi: 10.1126/sciadv.adl4743.

Eelke Spaak, Kei Watanabe, Shintaro Funahashi, and Mark G. Stokes. Stable and Dynamic Coding for Working Memory in Primate Prefrontal Cortex. *Journal of Neuroscience*, 37(27):6503–6516, July 2017. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.3364-16.2017.

David Sussillo and L. F. Abbott. Generating Coherent Patterns of Activity from Chaotic Neural Networks. *Neuron*, 63(4):544–557, August 2009. ISSN 0896-6273. doi: 10.1016/j.neuron.2009.07.018.

David Sussillo and Omri Barak. Opening the Black Box: Low-Dimensional Dynamics in High-Dimensional Recurrent Neural Networks. *Neural Computation*, 25(3):626–649, March 2013. ISSN 0899-7667. doi: 10.1162/NECO_a_00409.

David Sussillo, Mark M. Churchland, Matthew T. Kaufman, and Krishna V. Shenoy. A neural network that finds a naturalistic solution for the production of muscle activity. *Nature Neuroscience*, 18(7):1025–1033, July 2015. ISSN 1546-1726. doi: 10.1038/nn.4042.

Emily L. Sylwestrak, YoungJu Jo, Sam Vesuna, Xiao Wang, Blake Holcomb, Rebecca H. Tien, Doo Kyung Kim, Lief Fenno, Charu Ramakrishnan, William E. Allen, Ritchie Chen, Krishna V. Shenoy, David Sussillo, and Karl Deisseroth. Cell-type-specific population dynamics of diverse reward computations. *Cell*, 185(19):3568–3587.e27, September 2022. ISSN 1097-4172. doi: 10.1016/j.cell.2022.08.019.

Elia Turner and Omri Barak. The Simplicity Bias in Multi-Task RNNs: Shared Attractors, Reuse of Dynamics, and Geometric Representation. *Advances in Neural Information Processing Systems*, 36:25495–25507, December 2023.

Elia Turner, Kabir V Dabholkar, and Omri Barak. Charting and Navigating the Space of Solutions for Recurrent Neural Networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 25320–25333, 2021.

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, December 2022.

Christopher Versteeg, Andrew R. Sedler, Jonathan D. McCart, and Chethan Pandarinath. Expressive dynamics models with nonlinear injective readouts enable reliable recovery of latent features from neural activity. In *Proceedings of the 2nd NeurIPS Workshop on Symmetry and Geometry in Neural Representations*, pp. 255–278, August 2024.

P.J. Werbos. Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, October 1990. ISSN 1558-2256. doi: 10.1109/5.58337.

Alex H. Williams. Equivalence between representational similarity analysis, centered kernel alignment, and canonical correlations analysis. In *UniReps: 2nd Edition of the Workshop on Unifying Representations in Neural Models*, October 2024.

Alex H. Williams, Erin Kunz, Simon Kornblith, and Scott Linderman. Generalized Shape Metrics on Neural Representations. In *Advances in Neural Information Processing Systems*, November 2021.

Daniel L. K. Yamins, Ha Hong, Charles F. Cadieu, Ethan A. Solomon, Darren Seibert, and James J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, June 2014. doi: 10.1073/pnas.1403112111.

Guangyu Robert Yang, Madhura R. Joglekar, H. Francis Song, William T. Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22(2):297–306, February 2019. ISSN 1546-1726. doi: 10.1038/s41593-018-0310-2.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 12310–12320, July 2021.

A Appendix

A.1 Training and other miscellaneous details

For all experiments, we use RNNs with $N = 128$ neurons. All RNNs are trained in PyTorch. We use the Adam optimizer with a learning rate of 10^{-3}, a weight decay of 10^{-5}, and a batch size of 32. For the strength of the similarity penalty, we use $\lambda = 0.05$ throughout. When computing linear predictivity, we use $\rho = 10^{-3}$ as the ridge regularizer. RNNs trained as part of the INSD procedure are initialized at the scale $g = 1$. All networks are trained for a minimum of 10^6 iterations, with training terminating when the loss stops improving. Training runs were primarily done using 4th Generation Intel Xeon CPUs; GPU acceleration was not necessary.

For computing DSA dissimilarity matrices, we use the open source package from (Ostrow et al., 2023). Across all tasks, we used a rank of 100, 8 delays, and a delay interval of 10 timesteps. The delay parameters were selected to be compatible with trials of duration 100 timesteps, as used for context-dependent integration and 3-bit flipflop.

For finding fixed points, we use the open source package FixedPointFinder (Golub & Sussillo, 2018). We report approximate fixed points with velocities q spanning $q = 5 \times 10^{-4}$ to $q = 10^{-9}$, and subsample redundant fixed points by adjusting the uniqueness tolerance parameter. As in (Driscoll et al., 2024), we report fixed points over a wide range of velocity tolerances to best account for variations in relevant timescales across the different tasks.

All training and analysis code will be made public on GitHub upon acceptance.

A.2 A brief note on neural similarity penalty loopholes

We model the scenario described in the main text as follows: we are given two sets of neural representations $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{P \times N}$. Suppose that the representations in $\boldsymbol{X}$ are contained in a low dimensional subspace of dimension $k \ll N, P$. We represent this by factorizing $\boldsymbol{X} = \boldsymbol{L}\boldsymbol{W}$, where $\boldsymbol{L} \in \mathbb{R}^{P \times k}$ are the latent representations and $\boldsymbol{W} \in \mathbb{R}^{k \times N}$. Suppose further that $\boldsymbol{Y}$ is composed of identical latents, along with some irrelevant noise in other dimensions. We write this as $\boldsymbol{Y} = [\boldsymbol{L}\boldsymbol{Q} \quad \sigma \boldsymbol{Z}]$, where $\boldsymbol{Q} \in \mathbb{R}^{k \times k}$ is an orthogonal matrix, $\boldsymbol{Z} \in \mathbb{R}^{P \times d}$ represents the irrelevant noise, and $d = N - k$. For simplicity, we model the entries of $\boldsymbol{Z}$ as drawn i.i.d from $\mathcal{N}(0, 1)$. Below, we compute and describe the behavior of various similarity metrics on these inputs at large N, P, and σ.

A.2.1 Centered kernel alignment (CKA)

We focus on linear CKA:

$$\text{CKA}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{\|\boldsymbol{X}^\top \boldsymbol{Y}\|^2}{\|\boldsymbol{X}^\top \boldsymbol{X}\| \|\boldsymbol{Y}^\top \boldsymbol{Y}\|} \tag{5}$$

We expand the numerator as $\|\boldsymbol{X}^\top \boldsymbol{Y}\|^2 = \|\boldsymbol{X}^\top \boldsymbol{L}\boldsymbol{Q}\|^2 + \sigma^2 \|\boldsymbol{X}^\top \boldsymbol{Z}\|^2$.

We also expand $\|\boldsymbol{Y}^\top \boldsymbol{Y}\|^2 = \|\boldsymbol{L}^\top \boldsymbol{L}\|^2 + 2\sigma^2 \|\boldsymbol{L}^\top \boldsymbol{Z}\|^2 + \sigma^4 \|\boldsymbol{Z}^\top \boldsymbol{Z}\|^2$.

At large N, we can approximate $\boldsymbol{Z}\boldsymbol{Z}^\top / d \to \boldsymbol{I}_P$. This allows the simplification $\|\boldsymbol{X}^\top \boldsymbol{Z}\|^2 = \text{Tr}(\boldsymbol{Z}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{Z}) = \text{Tr}(\boldsymbol{Z}\boldsymbol{Z}^\top \boldsymbol{X}\boldsymbol{X}^\top) = d\|\boldsymbol{X}\|^2$, and $\|\boldsymbol{Z}^\top \boldsymbol{Z}\|^2 = d^2 P$. At large σ, we can drop subleading terms in σ, giving

$$\text{CKA}(\boldsymbol{X}, \boldsymbol{Y}) \approx \frac{\sigma^2 d \|\boldsymbol{X}\|^2}{\sigma^2 d \sqrt{P} \|\boldsymbol{X}^\top \boldsymbol{X}\|} \leq \mathcal{O}\left(\sqrt{\frac{k}{P}}\right), \tag{6}$$

where the final inequality follows from the bound $\|\boldsymbol{X}\|^2 \leq \sqrt{k} \|\boldsymbol{X}^\top \boldsymbol{X}\|$.

Thus, CKA between otherwise identical representations can be suppressed through irrelevant noise.

A.2.2 Representational Similarity Analysis (RSA)

We take RSA to refer to the cosine similarity between the squared Euclidean distance representational dissimilarity matrices (RDMs), as in Williams (2024).

Let $\boldsymbol{D}^X_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$ and $\boldsymbol{D}^Y_{ij} = \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2$ represent the $P \times P$ RDMs. We have:

$$\text{RSA}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{\langle \boldsymbol{D}^X, \boldsymbol{D}^Y \rangle}{\|\boldsymbol{D}^X\| \|\boldsymbol{D}^Y\|} \tag{7}$$

We can write $\boldsymbol{D}^Y_{ij} = \boldsymbol{D}^S_{ij} + \sigma^2 \boldsymbol{D}^Z_{ij}$, where $\boldsymbol{S} = \boldsymbol{L}\boldsymbol{Q}$. Dropping terms subleading in σ, we have

$$\text{RSA}(\boldsymbol{X}, \boldsymbol{Y}) \approx \frac{\langle \boldsymbol{D}^X, \boldsymbol{D}^Z \rangle}{\|\boldsymbol{D}^X\| \|\boldsymbol{D}^Z\|} \tag{8}$$

Note that $\mathbb{E}[\boldsymbol{D}^Z_{ij}] = 2d$ for $i \neq j$. At large N (and therefore large d), we can expect concentration, yielding $\boldsymbol{D}^Z/d \to 2(\boldsymbol{J} - \boldsymbol{I})$, where $\boldsymbol{J}$ is a $P \times P$ matrix of ones. Thus, we have

$$\text{RSA}(\boldsymbol{X}, \boldsymbol{Y}) \approx \frac{\sum_{i \neq j} \boldsymbol{D}^X_{ij}}{\sqrt{\sum_{i \neq j} (\boldsymbol{D}^X_{ij})^2} \sqrt{P(P-1)}} = \mathcal{O}\left(\frac{1}{P}\right). \tag{9}$$

Thus, RSA is also suppressed by irrelevant noise.

A.2.3 Linear predictivity [ref. → penalized]

As in the main text, define the projection operator $\boldsymbol{U}_X = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^+ \boldsymbol{X}^\top$. We have that

$$r^2(\boldsymbol{X}, \boldsymbol{Y}) = \frac{\|\boldsymbol{U}_X \boldsymbol{Y}\|^2}{\|\boldsymbol{Y}\|^2}. \tag{10}$$

We can write $\|\boldsymbol{U}_X \boldsymbol{Y}\|^2 = \|\boldsymbol{S}\|^2 + \sigma^2 \|\boldsymbol{U}_X \boldsymbol{Z}\|^2$, where we have used that $\boldsymbol{U}_X \boldsymbol{S} = \boldsymbol{S}$, as by construction, $\boldsymbol{S} = \boldsymbol{L}\boldsymbol{Q}$ is contained in the column space of $\boldsymbol{X} = \boldsymbol{L}\boldsymbol{W}$. Similarly, we have $\|\boldsymbol{Y}\|^2 = \|\boldsymbol{S}\|^2 + \sigma^2 \|\boldsymbol{Z}\|^2$, yielding $r^2(\boldsymbol{X}, \boldsymbol{Y}) \approx \frac{\|\boldsymbol{U}_X \boldsymbol{Z}\|^2}{\|\boldsymbol{Z}\|^2}$ at large σ. Finally, at large N, we have that

$$r^2(\boldsymbol{X}, \boldsymbol{Y}) \approx \frac{\|\boldsymbol{U}_X \boldsymbol{Z}\|^2}{\|\boldsymbol{Z}\|^2} \to \frac{\mathbb{E}\left[\|\boldsymbol{U}_X \boldsymbol{Z}\|^2\right]}{\mathbb{E}\left[\|\boldsymbol{Z}\|^2\right]} = \frac{kd}{Pd} = \frac{k}{P}, \tag{11}$$

demonstrating that linear predictivity in this direction is also suppressed by irrelevant noise.

A.2.4 Linear predictivity [penalized → ref.]

Consider the opposite direction:

$$r^2(\boldsymbol{Y}, \boldsymbol{X}) = \frac{\|\boldsymbol{U}_Y \boldsymbol{X}\|^2}{\|\boldsymbol{X}\|^2}. \tag{12}$$

Since the column space of $\boldsymbol{Y}$ contains that of $\boldsymbol{X}$, we have $\|\boldsymbol{U}_Y \boldsymbol{X}\|^2 = \|\boldsymbol{X}\|^2$, yielding $r^2(\boldsymbol{Y}, \boldsymbol{X}) = 1$. Thus, perfect linear predictivity is maintained.

A.3 Task parameters

Context-dependent integration: We use a timestep of $dt = 0.1$, a context-only duration $T_{\text{pre}} = 2.5$ (25 timesteps), and a total trial duration of $T = 10$ (100 timesteps). We set the noise scale to $\sigma = \sqrt{0.1}$.

3-bit flipflop: We use a timestep of $dt = 0.2$, and a total trial duration of $T = 20$ (100 timesteps). We set $p = 0.1$ as the spike probability per timestep.

MemoryPro: We use a timestep of $dt = 0.2$. Mirroring timing parameters selected in (Driscoll et al., 2024), we set $T^-_{\text{del}} = T^-_{\text{resp}} = 3/dt$, $T^+_{\text{del}} = T^+_{\text{resp}} = 7/dt$, $T^-_{\text{stim}} = T^-_{\text{mem}} = 2/dt$, and $T^+_{\text{stim}} = T^+_{\text{mem}} = 16/dt$. We use a noise scale of $\sigma = 0.1$. As in (Costacurta et al., 2024), we scale down the output channel corresponding to the fixation target by a factor of 0.8.

A.4 Additional Figures

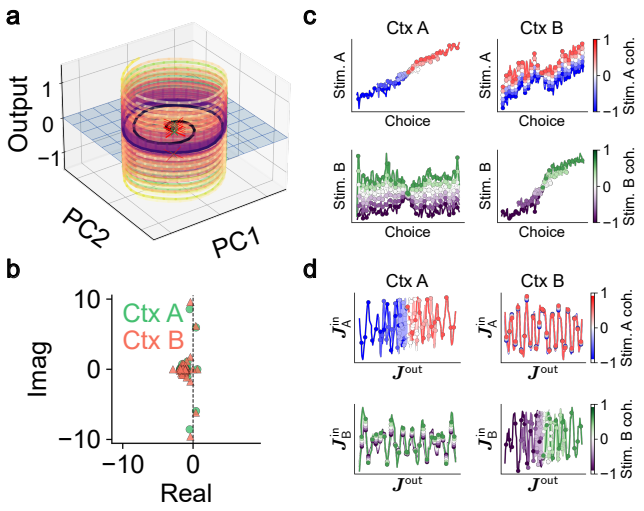

Figure A.1: **Properties of the alt-2 RNN for the context-dependent integration task. a,b.** Analogous to Figs. 1b,c. As for the alt-1 RNN, we observe oscillatory dynamics, as well as fixed points with unstable oscillatory modes. However, these oscillatory modes are of much higher frequency. **c,d.** Analogous to Fig. 2a. Unlike the alt-1 RNN, average trajectories plotted in the regression subspace to some extent maintain the relative ordering of the coherences of both stimuli. This is likely explained by the alt-2 RNN still retaining a degree of linear predictivity of standard RNN representations, something that was entirely absent for the alt-1 RNN (Fig. 1d). However, representations in the weight subspace reveal no consistent representation of stimuli coherences.

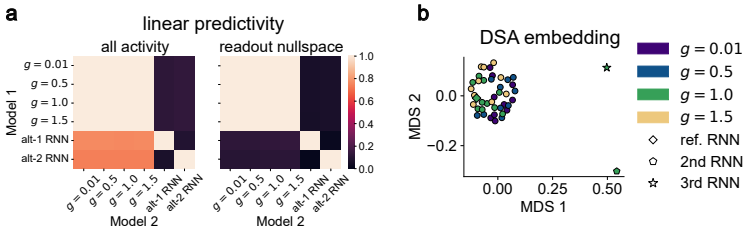

Figure A.2: **Similarity measures across standard and similarity-penalized models trained on the 3-bit flipflop task.** Figures are analogous to those in Fig. 4c,d. Similarity-penalized RNNs retain some degree of linear predictiity of standard RNNs, but that effect is ablated once representations are projected to readout nullspaces. As for other tasks, we also observe a DSA embdding that significantly separates the solutions similarity-penalized RNNs from those found by standard RNNs.

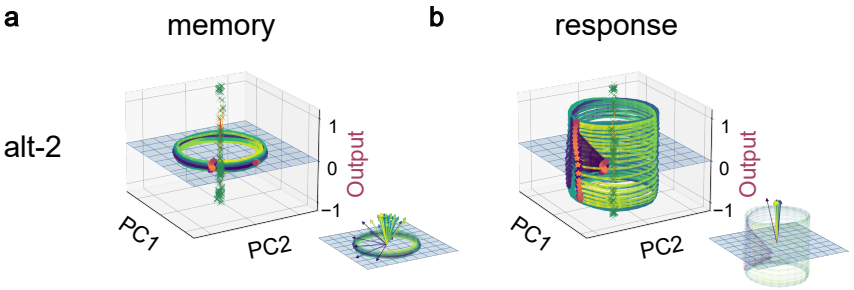

Figure A.3: **Properties of the alt-2 RNN for the MemoryPro task.** Figures are analogous to those in Fig. 4b. We again observe oscillatory dynamics supported by a center of unstable fixed points. This RNN does poorly on the task relative to the RNNs shown in Fig. 4b, as indicated by the activity itself prematurely acquiring significant output potence during the memory phase.

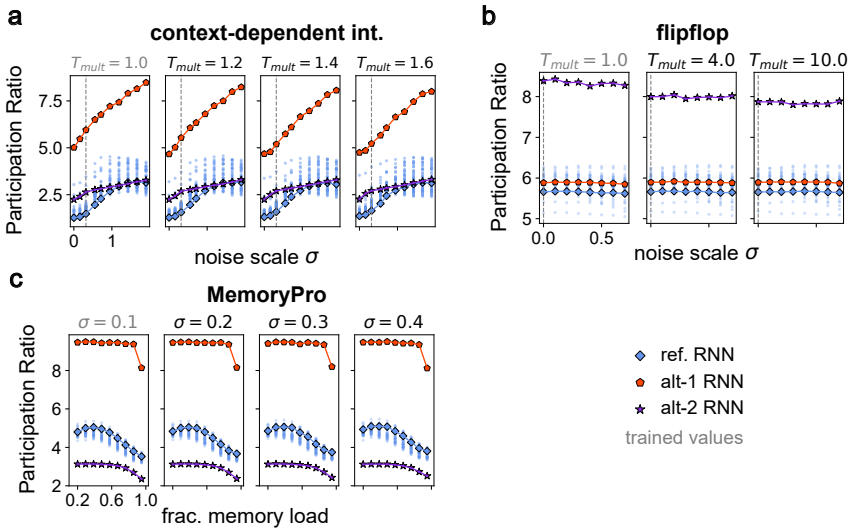

Figure A.4: **Effective dimensionality over different task conditions.** Plots are analogous to those in Fig. 5, but instead report the participation ratio, computed over neural activity during the task.

A.5 LLM USAGE STATEMENT

Large language models were used sparingly for the sole purpose of trimming and polishing text. All technical contributions, experiments, analyses, and figures are the authors' own.