Sufficient conditions for offline reactivation in recurrent neural networks

Nanda H Krishna
Mila – Quebec AI Institute
Université de Montréal
nanda.harishankar-krishna@mila.quebec

Colin Bredenberg
Mila – Quebec AI Institute
Université de Montréal
colin.bredenberg@mila.quebec

Daniel Levenstein
Mila – Quebec AI Institute
McGill University
daniel.levenstein@mila.quebec

Blake Aaron Richards
Mila – Quebec AI Institute
McGill University
blake.richards@mila.quebec

Guillaume Lajoie
Mila – Quebec AI Institute
Université de Montréal
g.lajoie@umontreal.ca

Abstract

During periods of quiescence, such as sleep, neural activity in many brain circuits resembles that observed during periods of task engagement. However, the precise conditions under which task-optimized networks can autonomously reactivate the same network states responsible for online behavior are poorly understood. In this study, we develop a mathematical framework that outlines sufficient conditions for the emergence of neural reactivation in circuits that encode features of smoothly varying stimuli. We demonstrate mathematically that noisy recurrent networks optimized to track environmental state variables using change-based sensory information naturally develop denoising dynamics, which, in the absence of input, cause the network to revisit state configurations observed during periods of online activity. We validate our findings using numerical experiments on two canonical neuroscience tasks: spatial position estimation based on self-motion cues, and head direction estimation based on angular velocity cues. Overall, our work provides theoretical support for modeling offline reactivation as an emergent consequence of task optimization in noisy neural circuits.

1 Introduction

Neural circuits in the brain are known to recapitulate task-like activity during periods of quiescence, such as sleep [1]. For example, the hippocampus "replays" sequences of represented spatial locations akin to behavioral trajectories during wakefulness [2–4]. Furthermore, frontal [5, 6], sensory [7, 8], and motor [9] cortices reactivate representations associated with recent experiences; and sleep activity in the anterior thalamus [10] and entorhinal cortex [11] is constrained to the same neural manifolds that represent head direction and spatial position in those circuits during wakefulness.

This neural reactivation phenomenon is thought to have a number of functional benefits, including the formation of long term memories [12, 13], abstraction of general rules or "schema" [14], and offline planning of future actions [15, 16]. Similarly, replay in artificial systems has been shown to

Preprint.

be valuable in reinforcement learning, when training is sparse or expensive [17], and in supervised learning, to prevent catastrophic forgetting in continual learning tasks [18]. However, where machine learning approaches tend to save sensory inputs from individual experiences in an external memory buffer, or use external networks that are explicitly trained to generate artificial training data [19], reactivation in the brain is autonomously generated in the same circuits that operate during active perception and action. Currently, it is unknown how reactivation can emerge in the same networks that encode information during active behavior, or why it is so widespread in neural circuits.

Previous approaches to modeling reactivation in neural circuits fall into two broad categories: generative models that have been explicitly trained to reproduce realistic sensory inputs [20], and models in which replay is an emergent consequence of the architecture of network models with a particular connectivity structure [21, 22] or local synaptic plasticity mechanism [23–26]. Generative modeling approaches have strong theoretical guarantees that reactivation will occur, because networks are explicitly optimized to provide this functionality. However, modeling approaches that argue for *emergent* reactivation typically rely on empirical results, and lack rigorous mathematical justification.

In this study, we demonstrate that a certain type of reactivation—diffusive reactivation—can emerge from a system attempting to optimally encode features of its environment in the presence of internal noise. We observe that continuous-time recurrent neural networks (RNNs), trained to optimally integrate and track perceptual variables based on sensations of change (angular velocity, motion through space, etc.), will naturally exhibit reactivation during quiescent states (when subject to noise but in the absence of perceptual inputs). We explain these phenomena by demonstrating that noise compensation dynamics naturally induce diffusion on task-relevant neural manifolds in optimally trained networks. We provide a mathematical derivation that outlines sufficient conditions for this phenomenon to occur. Subsequently, we follow with a series of empirical validations in the context of two ecologically relevant tasks: a spatial navigation task, and a head direction integration task.

2 Mathematical Results

In this study, we will consider a noisy discrete-time approximation of a continuous-time RNN, receiving change-based information $\frac{\mathrm{d}\mathbf{s}(t)}{\mathrm{d}t}$ about an N_s-dimensional environmental state vector $\mathbf{s}(t)$. The network's objective will be to reconstruct some function of these environmental state variables, $f\big(\mathbf{s}(t)\big) : \mathbb{R}^{N_s} \to \mathbb{R}^{N_o}$, where N_s is the number of stimulus dimensions and N_o is the number of output dimensions. An underlying demand for this family of tasks is that path integration needs to be performed, possibly followed by some computations based on that integration. These requirements are often met in natural settings, as it is widely believed that animals are able to estimate their location in space $\mathbf{s}(t)$ through path integration based exclusively on local motion cues $\frac{\mathrm{d}\mathbf{s}(t)}{\mathrm{d}t}$, and neural circuits in the brain that perform this computation have been identified (specifically the entorhinal cortex [27]). For our analysis, we will assume that the stimuli the network receives are drawn from a stationary distribution, such that $p(\mathbf{s}(t))$ does not depend on time—for navigation, this amounts to ignoring the effects of initial conditions on an animal's state occupancy statistics, and assumes that the animal's navigation policy remains constant throughout time. The RNN's dynamics are given by:

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \Delta\mathbf{r}(t) \tag{1}$$

$$\Delta\mathbf{r}(t) = \phi\left(\mathbf{r}(t), \mathbf{s}(t), \frac{\mathrm{d}\mathbf{s}(t)}{\mathrm{d}t}\right)\Delta t + \sigma\boldsymbol{\eta}(t), \tag{2}$$

where $\Delta\mathbf{r}(t)$ is a function that describes the network's update dynamics as a function of the stimulus, $\phi(\cdot)$ is a sufficiently expressive nonlinearity, $\boldsymbol{\eta}(t) \sim \mathcal{N}(0, \Delta t)$ is Brownian noise, and Δt is taken to be small as to approximate corresponding continuous-time dynamics. We work with a discrete-time approximation here for the sake of simplicity, and also to illustrate how the equations are implemented in practice during simulations. Suppose that the network's output is given by $\mathbf{o} = \mathbf{D}\mathbf{r}(t)$, where $\mathbf{D}$ is an $N_o \times N_r$ matrix that maps neural activity to outputs, and N_r is the number of neurons in the RNN.

We formalize our loss function for each time point as follows:

$$\mathcal{L}(t) = \mathbb{E}_{\boldsymbol{\eta}}\|f\big(\mathbf{s}(t)\big) - \mathbf{D}\mathbf{r}(t)\|_2, \tag{3}$$

so that as the loss is minimized over timesteps, the system is optimized to match its target at every timestep while compensating for its own intrinsic noise. We find that the greedily optimal dynamics,

in the presence of noise, for an upper bound of this loss are given by:

$$\Delta \mathbf{r}^*(t) = \left[\underbrace{\sigma^2 \frac{\mathrm{d}}{\mathrm{d}\mathbf{r}(t)} \log p(\mathbf{r}(t))}_{\text{denoising}} + \underbrace{\mathbf{D}^\dagger \frac{\mathrm{d}f(\mathbf{s}(t))}{\mathrm{d}\mathbf{s}(t)} \frac{\mathrm{d}\mathbf{s}(t)}{\mathrm{d}t}}_{\text{state updation}} \right] \Delta t + \sigma \boldsymbol{\eta}(t). \tag{4}$$

These dynamics are interpretable: any system attempting to maintain a relationship to a stimulus in the presence of noise must first perform denoising, and then use instantaneous changes in the state variable ($\frac{\mathrm{d}\mathbf{s}(t)}{\mathrm{d}t}$) to update state information. A detailed derivation is provided in Appendix A.

We are now in a position to ask: what happens in the absence of any input to the system, as would be observed in a quiescent state? We will make two assumptions for our model of the quiescent state: 1) $\frac{\mathrm{d}\mathbf{s}(t)}{\mathrm{d}t} = 0$, so that no time-varying input is being provided to the system, and 2) the variance of the noise is increased by a factor of two (deviating from this factor is not catastrophic as discussed below). This gives the following quiescent dynamics $\tilde{\mathbf{r}}$:

$$\Delta \tilde{\mathbf{r}}(t) = \left[\sigma^2 \frac{\mathrm{d}}{\mathrm{d}\mathbf{r}(t)} \log p(\mathbf{r}(t)) \right] \Delta t + \sqrt{2}\sigma \boldsymbol{\eta}(t). \tag{5}$$

Interestingly, this corresponds to Langevin sampling of $p(\mathbf{r})$. Therefore, we can predict an equivalence between the steady-state quiescent sampling distribution $\tilde{p}(\mathbf{r})$ and the active probability distribution over neural states $p(\mathbf{r})$ (so that $p(\mathbf{r}) = \tilde{p}(\mathbf{r})$, and consequently $p(\mathbf{o}) = \tilde{p}(\mathbf{o})$). There are two key components that made this occur: first, the system needed to be performing near-optimal noisy state estimation; second, the system state needed to be determined purely by integrating changes in sensory variables of interest. The final assumption—that noise is doubled during quiescent states—is necessary only to produce sampling from the *exact* same distribution $p(\mathbf{r})$. Different noise variances will result in sampling from similar steady-state distributions with different temperature parameters. When these conditions are present, we can expect to see reactivation phenomena during quiescence in optimally trained networks.

3 Numerical Experiments

To validate our mathematical results, we consider numerical experiments with noisy "vanilla" continuous-time RNNs on two canonical neuroscience tasks: spatial position estimation using motion cues, and head direction estimation using angular velocity cues. We provide task and network details in Appendix B. Both of these tasks conform to the structure of the general estimation task considered in our mathematical analysis. For each task, we minimize the mean-squared error between the network output $\mathbf{o}$ and the task-specific target given by $f(\mathbf{s}(t))$, summed across timesteps. In this section and Appendix C.1, we discuss the analyses associated with the spatial position estimation task. We refer the reader to Appendix C.2 for experiments on the head direction estimation task.

First, we visualized the decoded output activity for the active and quiescent phases (Fig. 1a-b). It is clear that decoded output activity during the quiescent phase is smooth, and tiles output space similarly to trajectories sampled during the waking phase. To quantify the similarity in the *distributions* of activity during the active and quiescent phases, we computed 2D kernel density estimates[1] (KDEs) on the output trajectories (Fig. 1e-f). We indeed found that the distribution of activity was similar across active and quiescent phases, as predicted by our mathematical results. However, output trajectories in the quiescent phase do not tile space as uniformly as those in the active phase.

Our theory additionally predicts that if the distribution of network states during the active phase is biased in some way during training, the distribution during the quiescent phase should also be biased accordingly. To test this, we modified the behavioral policy of our agent during training, introducing a drift term that caused it to occupy a ring of spatial locations in the center of the field rather than uniformly tiling space. We found again a close correspondence between decoded output trajectories of the active and quiescent phases (Fig. 1c-d), which was also reflected in the KDEs (Fig. 1g-h).

We carried out several additional analyses in order to validate our mathematical results, and these are presented in Appendix C. Our results collectively verify that during quiescence, our trained networks do indeed approximately sample from the waking trajectory distribution.

[1] We used the `stats.gaussian_kde()` method from `scipy` [28], with default values for its parameters.

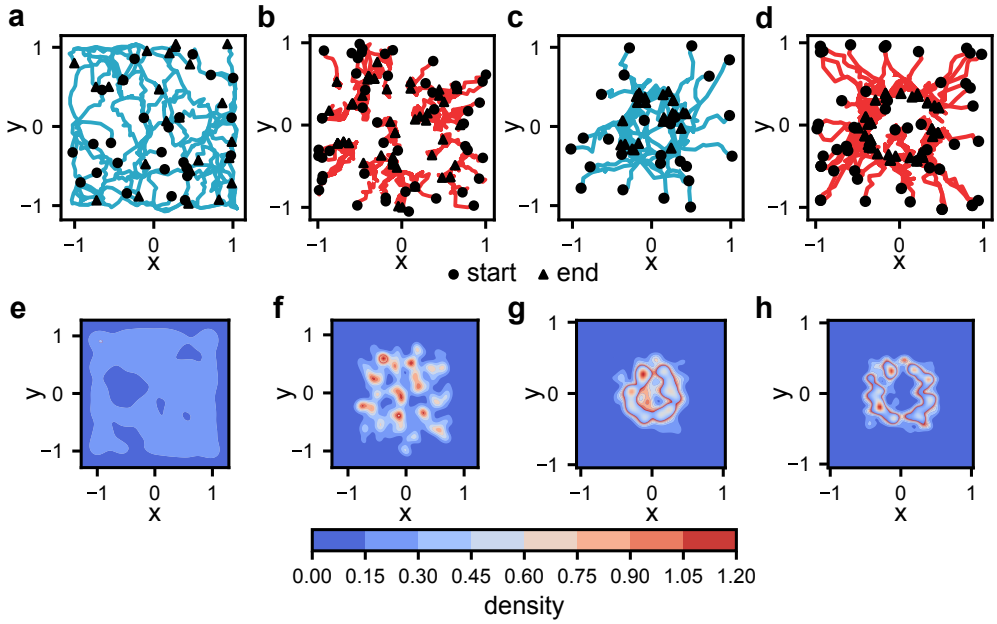

Figure 1: **Reactivation in a spatial position estimation task**. **a-b)** Sample decoded outputs during active (a) and quiescent (b) behavior for networks trained under uniform trajectories. Circles indicate the initial location, triangles indicate the final location. **c-d)** Same as (a-b), but for biased trajectories. **e-f)** 2D kernel density estimate (KDE) plot for decoded outputs in the active (e) and quiescent (f) phases, for 500 uniform trajectories. **g-h)** Same as (e-f) but for biased trajectories.

4 Discussion

In this study, we have provided mathematical conditions under which reactivation is expected to emerge in task-optimized recurrent neural circuits. Our criteria are as follows: first, the network must implement a noisy, continuous-time dynamical system; second, the network must be solving a state variable estimation task near-optimally, by integrating exclusively change-based inputs ($\frac{\mathrm{d}\mathbf{s}(t)}{\mathrm{d}t}$) to reconstruct some function of the state variables ($f(\mathbf{s}(t))$). Under these conditions, we demonstrated that a greedily optimal solution to the task involves a combination of integrating the state variables and denoising. In absence of inputs (quiescent phase), we assumed that the system would receive no stimuli ($\frac{\mathrm{d}\mathbf{s}(t)}{\mathrm{d}t} = 0$) so that the system is dominated by its denoising dynamics, and that noise variance would increase slightly (by a factor of 2). Under these conditions, we showed that the steady-state probability distribution of network states during quiescence ($\tilde{p}(\mathbf{r})$) should be equivalent to the distribution of network states during active task performance ($p(\mathbf{r})$). Thus, these conditions constitute criteria for a form of reactivation to emerge in trained neural systems.

We have validated our mathematical results empirically in two tasks with neuroscientific relevance. The first, a path integration task, required the network to identify its location in space based on motion cues. This form of path integration has been used to model the entorhinal cortex [27], a key brain area in which reactivation dynamics have been observed [11, 29]. The second task required the network to estimate a head direction orientation based on angular velocity cues. This function in the mammalian brain has been attributed to the anterodorsal thalamic nucleus (ADn) and post-subiculum (PoS) [30–32], another critical locus for reactivation dynamics [10]. Previous attempts to model these systems that have relied on hand-crafted network models have been able to reproduce reactivation dynamics. This was done by embedding a smooth attractor structure in the network's recurrent connectivity along which activity may diffuse during quiescence [33]. Similarly, we have identified attractors in our trained networks' latent activation space—we found a smooth map of space in the spatial navigation task (Fig. C.1) and a ring attractor in the head direction task (Fig. C.3). In our case, these attractors proved to be *optimal* for task performance, and consequently did not require hand crafting. Furthermore, beyond previous studies, we were able to show that the statistics of reactivation in our trained networks mimicked the statistics of activity during waking behavior, and that manipulation of

waking behavioral statistics was directly reflected in offline reactivation dynamics. Thus, our work complements these previous studies by providing a mathematical justification for the emergence of reactivation dynamics in terms of optimal task performance.

Our results suggest that reactivation in the brain could be a natural consequence of learning in the presence of noise, rather than the product of an explicit generative demand [20, 34]. Thus, reactivation during quiescence in a brain area should not be taken as evidence exclusively in favor of generative modeling: the alternative possibility, as identified by our work, is that reactivation could be an emergent consequence of optimization for certain tasks (though it could be used for other computations). Our hypothesis and generative modeling hypotheses may be experimentally dissociable: while generative models necessarily recapitulate the moment-to-moment transition statistics of sensory data, our approach only predicts that the *stationary distribution* will be identical. This opens the possibility for reactivation of sequences that do not respect the ordering of states observed during waking (e.g. reverse replay [35, 36]), as well as changes in the timescale of reactivation [2].

In addition to the tasks discussed in our numerical experiments, our work has the potential to function as a justification for a wide variety of reactivation phenomena observed in the brain. Further details have been provided in Appendix D. Beyond this, it may further provide a mechanism for inducing reactivation in neural circuits in order to support critical maintenance functions, such as memory consolidation or learning.

Acknowledgments

NHK would like to thank Mila for providing access to computational resources that enabled this work. NHK would also like to thank the IVADO and UNIQUE Excellence Scholarships, and the UniReps-Gatsby Foundation Grant for providing financial support.

References

[1] David Tingley and Adrien Peyrache. On the methods for reactivation and replay analysis. *Philosophical Transactions of the Royal Society B*, 375(1799):20190231, 2020.

[2] Zoltán Nádasdy, Hajime Hirase, András Czurkó, Jozsef Csicsvari, and György Buzsáki. Replay and time compression of recurring spike sequences in the hippocampus. *Journal of Neuroscience*, 19(21):9497–9507, 1999.

[3] Albert K Lee and Matthew A Wilson. Memory of sequential experience in the hippocampus during slow wave sleep. *Neuron*, 36(6):1183–1194, 2002.

[4] David J Foster. Replay comes of age. *Annual review of neuroscience*, 40:581–602, 2017.

[5] David R Euston, Masami Tatsuno, and Bruce L McNaughton. Fast-forward playback of recent memory sequences in prefrontal cortex during sleep. *science*, 318(5853):1147–1150, 2007.

[6] Adrien Peyrache, Mehdi Khamassi, Karim Benchenane, Sidney I Wiener, and Francesco P Battaglia. Replay of rule-learning related neural patterns in the prefrontal cortex during sleep. *Nature neuroscience*, 12(7):919–926, 2009.

[7] Tal Kenet, Dmitri Bibitchkov, Misha Tsodyks, Amiram Grinvald, and Amos Arieli. Spontaneously emerging cortical representations of visual attributes. *Nature*, 425(6961):954–956, 2003.

[8] Shengjin Xu, Wanchen Jiang, Mu-ming Poo, and Yang Dan. Activity recall in a visual cortical ensemble. *Nature neuroscience*, 15(3):449–455, 2012.

[9] Kari L Hoffman and Bruce L McNaughton. Coordinated reactivation of distributed memory traces in primate neocortex. *Science*, 297(5589):2070–2073, 2002.

[10] Adrien Peyrache, Marie M Lacroix, Peter C Petersen, and György Buzsáki. Internally organized mechanisms of the head direction sense. *Nature neuroscience*, 18(4):569–575, 2015.

[11] 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.

[12] György Buzsáki. Two-stage model of memory trace formation: a role for "noisy" brain states. *Neuroscience*, 31(3):551–570, 1989.

[13] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.

[14] Penelope A Lewis and Simon J Durrant. Overlapping memory replay during sleep builds cognitive schemata. *Trends in cognitive sciences*, 15(8):343–351, 2011.

[15] H Freyja Ólafsdóttir, Caswell Barry, Aman B Saleem, Demis Hassabis, and Hugo J Spiers. Hippocampal place cells construct reward related sequences through unexplored space. *Elife*, 4:e06063, 2015.

[16] Hideyoshi Igata, Yuji Ikegaya, and Takuya Sasaki. Prioritized experience replays on a hippocampal predictive map for learning. *Proceedings of the National Academy of Sciences*, 118(1):e2011266118, 2021.

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[18] Tyler L Hayes, Nathan D Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9769–9776. IEEE, 2019.

[19] Tyler L Hayes and Christopher Kanan. Online continual learning for embedded devices. *arXiv preprint arXiv:2203.10681*, 2022.

[20] Nicolas Deperrois, Mihai A Petrovici, Walter Senn, and Jakob Jordan. Learning cortical representations through perturbed and adversarial dreaming. *Elife*, 11:e76384, 2022.

[21] Bin Shen and Bruce L McNaughton. Modeling the spontaneous reactivation of experience-specific hippocampal cell assembles during sleep. *Hippocampus*, 6(6):685–692, 1996.

[22] Amir H Azizi, Laurenz Wiskott, and Sen Cheng. A computational model for preplay in the hippocampus. *Frontiers in computational neuroscience*, 7:161, 2013.

[23] Ashok Litwin-Kumar and Brent Doiron. Formation and maintenance of neuronal assemblies through synaptic plasticity. *Nature communications*, 5(1):5319, 2014.

[24] Panagiota Theodoni, Bernat Rovira, Yingxue Wang, and Alex Roxin. Theta-modulation drives the emergence of connectivity patterns underlying replay in a network model of place cells. *Elife*, 7:e37388, 2018.

[25] Tatsuya Haga and Tomoki Fukai. Recurrent network model for learning goal-directed sequences through reverse replay. *Elife*, 7:e34171, 2018.

[26] Toshitake Asabuki and Tomoki Fukai. Learning rules for cortical-like spontaneous replay of an internal model. *bioRxiv*, pages 2023–02, 2023.

[27] Ben Sorscher, Gabriel Mel, Surya Ganguli, and Samuel Ocko. A unified theory for the origin of grid cells through the lens of pattern formation. *Advances in neural information processing systems*, 32, 2019.

[28] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[29] Richard J Gardner, Li Lu, Tanja Wernle, May-Britt Moser, and Edvard I Moser. Correlation structure of grid cells is preserved during sleep. *Nature neuroscience*, 22(4):598–608, 2019.

[30] Jeffrey S Taube, Robert U Muller, and James B Ranck. Head-direction cells recorded from the postsubiculum in freely moving rats. ii. effects of environmental manipulations. *Journal of Neuroscience*, 10(2):436–447, 1990.

[31] Jeffrey S Taube. Head direction cells recorded in the anterior thalamic nuclei of freely moving rats. *Journal of Neuroscience*, 15(1):70–86, 1995.

[32] Jeffrey S Taube. The head direction signal: origins and sensory-motor integration. *Annu. Rev. Neurosci.*, 30:181–207, 2007.

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

[34] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The" wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

[35] David J Foster and Matthew A Wilson. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440(7084):680–683, 2006.

[36] Kamran Diba and György Buzsáki. Forward and reverse hippocampal place-cell sequences during ripples. *Nature neuroscience*, 10(10):1241–1242, 2007.

[37] Martin Raphan and Eero P Simoncelli. Least squares estimation without priors or supervision. *Neural computation*, 23(2):374–420, 2011.

[38] Uğur M Erdem and Michael Hasselmo. A goal-directed spatial navigation model using forward trajectory planning based on grid cells. *European Journal of Neuroscience*, 35(6):916–931, 2012.

[39] Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J Chadwick, Thomas Degris, Joseph Modayil, et al. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433, 2018.

[40] Tom M George, William de Cothi, Claudia Clopath, Kimberly Stachenfeld, and Caswell Barry. Ratinabox: A toolkit for modelling locomotion and neuronal activity in continuous environments. *bioRxiv*, pages 2022–08, 2022.

[41] Zahra Kadkhodaie and Eero Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. *Advances in Neural Information Processing Systems*, 34:13242–13254, 2021.

[42] Alfonso Renart, Pengcheng Song, and Xiao-Jing Wang. Robust spatial working memory through homeostatic synaptic scaling in heterogeneous cortical networks. *Neuron*, 38(3):473–485, 2003.

[43] Nicholas Cain, Andrea K Barreiro, Michael Shadlen, and Eric Shea-Brown. Neural integrators for decision making: a favorable tradeoff between robustness and sensitivity. *Journal of neurophysiology*, 109(10):2542–2559, 2013.

[44] 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.

A Derivation of greedily optimal RNN dynamics

Our analysis proceeds as follows. We first derive an upper bound for the loss in Eq. 3 that partitions the optimal update $\Delta\mathbf{r}$ into two terms: one which requires the RNN to estimate the predicted change in the target function, and one which requires the RNN to compensate for the presence of noise. Then, we derive closed-form optimal dynamics for an upper bound of this loss, which reveals a decomposition of neural dynamics into state estimation and denoising terms. As discussed earlier, these optimal dynamics can produce offline sampling of states visited during training in the absence of stimuli ($\frac{\mathrm{d}\mathbf{s}(t)}{\mathrm{d}t} = 0$) but in the presence of noise.

A.1 Upper bound of the loss

To derive our upper bound, we first assume that $\phi\big(\mathbf{r}(t), \mathbf{s}(t), \frac{\mathrm{d}\mathbf{s}(t)}{\mathrm{d}t}\big)$ can be decomposed into two different functions so that Eq. 2 becomes:

$$\Delta\mathbf{r}(t) = \Delta\mathbf{r}_1\big(\mathbf{r}(t)\big) + \Delta\mathbf{r}_2\left(\mathbf{r}(t), \mathbf{s}(t), \frac{\mathrm{d}\mathbf{s}(t)}{\mathrm{d}t}\right) + \sigma\boldsymbol{\eta}(t), \tag{A.1}$$

where we will ultimately show that both functions scale with Δt. We are assuming that these two terms have different functional dependencies; however, for notational conciseness, we will subsequently refer to both updates as $\Delta\mathbf{r}_1(t)$ and $\Delta\mathbf{r}_2(t)$. The first, $\Delta\mathbf{r}_1(t)$, is a function of $\mathbf{r}(t)$ only, and will be used to denoise $\mathbf{r}(t)$ such that the approximate equality $\mathbf{r}(t) + \Delta\mathbf{r}_1(t) + \sigma\boldsymbol{\eta}(t) \approx \mathbf{D}^\dagger f\big(\mathbf{s}(t)\big)$ still holds ($\Delta\mathbf{r}_1(t)$ cancels out the noise corruption $\sigma\boldsymbol{\eta}(t)$), where $\mathbf{D}^\dagger$ is the right pseudoinverse of $\mathbf{D}$. This maintains optimality in the presence of noise. The second, $\Delta\mathbf{r}_2(t)$, is also a function of the input, and will build upon the first update to derive a simple state update such that $\mathbf{D}\big(\mathbf{r}(t+\Delta t)\big) \approx f\big(\mathbf{s}(t+\Delta t)\big)$. To construct this two-step solution, we first consider an upper bound on our original loss, which we will label $\mathcal{L}_{upper}$. Exploiting the fact that Δt^2 is infinitesimally small relative to other terms, we will Taylor expand Eq. 3 to first order about a small timestep increment Δt:

$$\mathcal{L}(t+\Delta t) = \mathbb{E}_{\boldsymbol{\eta}}\|f\big(\mathbf{s}(t+\Delta t)\big) - \mathbf{D}\big(\mathbf{r}(t) + \Delta\mathbf{r}(t)\big)\|_2 \tag{A.2}$$

$$\approx \mathbb{E}_{\boldsymbol{\eta}}\|(f\big(\mathbf{s}(t)\big) + \frac{\mathrm{d}f\big(\mathbf{s}(t)\big)}{\mathrm{d}\mathbf{s}(t)}\Delta\mathbf{s}(t)) - \mathbf{D}\big(\mathbf{r}(t) + \Delta\mathbf{r}(t)\big)\|_2 \tag{A.3}$$

$$= \mathbb{E}_{\boldsymbol{\eta}}\|\frac{\mathrm{d}f\big(\mathbf{s}(t)\big)}{\mathrm{d}\mathbf{s}(t)}\Delta\mathbf{s}(t) - \mathbf{D}\Delta\mathbf{r}_2(t) + f\big(\mathbf{s}(t)\big) - \mathbf{D}\big(\mathbf{r}(t) + \Delta\mathbf{r}_1(t) + \sigma\boldsymbol{\eta}(t)\big)\|_2, \tag{A.4}$$

where $\Delta\mathbf{s}(t) = \frac{\mathrm{d}\mathbf{s}(t)}{\mathrm{d}t}\Delta t$. Next, using the triangle inequality, we note that the loss is upper bounded by a new loss $\mathcal{L}_2$, given by:

$$\mathcal{L} \leq \mathcal{L}_2 = \mathbb{E}_{\boldsymbol{\eta}}\left[\|\frac{\mathrm{d}f\big(\mathbf{s}(t)\big)}{\mathrm{d}\mathbf{s}(t)}\Delta\mathbf{s}(t) - \mathbf{D}\Delta\mathbf{r}_2(t)\|_2 + \|f\big(\mathbf{s}(t)\big) - \mathbf{D}\big(\mathbf{r}(t) + \Delta\mathbf{r}_1(t) + \sigma\boldsymbol{\eta}(t)\big)\|_2\right] \tag{A.5}$$

$$= \|\frac{\mathrm{d}f\big(\mathbf{s}(t)\big)}{\mathrm{d}\mathbf{s}(t)}\Delta\mathbf{s}(t) - \mathbf{D}\Delta\mathbf{r}_2(t)\|_2 + \mathbb{E}_{\boldsymbol{\eta}}\|f\big(\mathbf{s}(t)\big) - \mathbf{D}\big(\mathbf{r}(t) + \Delta\mathbf{r}_1(t) + \sigma\boldsymbol{\eta}(t)\big)\|_2, \tag{A.6}$$

which separates the loss into two independent terms: one which is a function of $\Delta\mathbf{r}_2(t)$ and the signal, while the other is a function of $\Delta\mathbf{r}_1(t)$ and the noise. The latter, $\Delta\mathbf{r}_1(t)$-dependent term, allows $\Delta\mathbf{r}_1$ to correct for noise-driven deviations between $f\big(\mathbf{s}(t)\big)$ and $\mathbf{D}\mathbf{r}(t)$. Here, we will assume that this optimization has been successful for previous timesteps, such that $\mathbf{r}(t) \approx \mathbf{D}^\dagger f\big(\mathbf{s}(t)\big)$, where $\mathbf{D}^\dagger$ is the right pseudoinverse of $\mathbf{D}$. By this assumption, we have the following approximation:

$$\mathcal{L}_2 \approx \|\frac{\mathrm{d}f\big(\mathbf{s}(t)\big)}{\mathrm{d}\mathbf{s}(t)}\Delta\mathbf{s}(t) - \mathbf{D}\Delta\mathbf{r}_2(t)\|_2 + \mathbb{E}_{\boldsymbol{\eta}}\|f\big(\mathbf{s}(t)\big) - \mathbf{D}\big(\mathbf{D}^\dagger f\big(\mathbf{s}(t)\big) + \Delta\mathbf{r}_1(t) + \sigma\boldsymbol{\eta}(t)\big)\|_2 \tag{A.7}$$

$$= \|\frac{\mathrm{d}f\big(\mathbf{s}(t)\big)}{\mathrm{d}\mathbf{s}(t)}\Delta\mathbf{s}(t) - \mathbf{D}\Delta\mathbf{r}_2(t)\|_2 + \mathbb{E}_{\boldsymbol{\eta}}\|\mathbf{D}\big(\Delta\mathbf{r}_1(t) + \sigma\boldsymbol{\eta}(t)\big)\|_2. \tag{A.8}$$

Thus, the second term in $\mathcal{L}_2$ trains $\Delta\mathbf{r}_1$ to greedily cancel out noise in the system $\sigma\boldsymbol{\eta}(t)$. To show that this is quite similar to correcting for deviations between $\mathbf{D}^\dagger f\big(\mathbf{s}(t)\big)$ and $\mathbf{r}$ in neural space (as opposed to output space), we use the Cauchy-Schwarz inequality to develop the following upper bound:

$$\mathcal{L}_2 \leq \mathcal{L}_3 = \|\frac{\mathrm{d}f\big(\mathbf{s}(t)\big)}{\mathrm{d}\mathbf{s}(t)}\Delta\mathbf{s}(t) - \mathbf{D}\Delta\mathbf{r}_2(t)\|_2 + \|\mathbf{D}\|_2\mathbb{E}_{\boldsymbol{\eta}}\|\Delta\mathbf{r}_1(t) + \sigma\boldsymbol{\eta}(t)\|_2. \tag{A.9}$$

This allows the system to optimize for denoising without having access to its outputs, allowing for computations that are more localized to the circuit. As our final step, we use Jensen's inequality for expectations to derive the final form of our loss upper bound:

$$\mathcal{L}_3 \leq \mathcal{L}_{upper} = \|\frac{\mathrm{d}f\big(\mathbf{s}(t)\big)}{\mathrm{d}\mathbf{s}(t)}\Delta\mathbf{s}(t) - \mathbf{D}\Delta\mathbf{r}_2(t)\|_2 + \|\mathbf{D}\|_2\sqrt{\mathbb{E}_{\boldsymbol{\eta}}\|\Delta\mathbf{r}_1(t) + \sigma\boldsymbol{\eta}(t)\|_2^2} \tag{A.10}$$

$$= \mathcal{L}_{signal}(\Delta\mathbf{r}_2) + \mathcal{L}_{noise}(\Delta\mathbf{r}_1), \tag{A.11}$$

where $\mathcal{L}_{signal} = \|\frac{\mathrm{d}f(\mathbf{s}(t))}{\mathrm{d}\mathbf{s}(t)}\Delta\mathbf{s}(t) - \mathbf{D}\Delta\mathbf{r}_2(t)\|_2$ is dedicated to tracking the state variable $\mathbf{s}(t)$, and $\mathcal{L}_{noise} = \|\mathbf{D}\|_2\sqrt{\mathbb{E}_{\boldsymbol{\eta}}\|\Delta\mathbf{r}_1(t) + \sigma\boldsymbol{\eta}(t)\|_2^2}$ is dedicated to denoising the network state. In the next section, we will describe how this objective function can be analytically optimized in terms of $\Delta\mathbf{r}_1(t)$ and $\Delta\mathbf{r}_2(t)$ in a way that decomposes the trajectory tracking problem into a combination of state estimation and denoising.

A.2 Optimizing the upper bound

Our optimization will be greedy, so that for each loss $\mathcal{L}(t+\Delta t)$ we will optimize only $\Delta\mathbf{r}(t)$, ignoring dependencies on updates from previous time steps. $\mathcal{L}_{noise}$ is the only term in our loss that depends on $\Delta\mathbf{r}_1(t)$. Ignoring proportionality constants and the square root (which do not affect the location of minima), we have:

$$\mathcal{L}_{noise} \equiv \mathbb{E}_{\boldsymbol{\eta}}\|\Delta\mathbf{r}_1(t) + \sigma\boldsymbol{\eta}(t)\|_2^2. \tag{A.12}$$

Essentially, the objective of $\Delta\mathbf{r}_1(t)$ is to cancel out the noise $\sigma\boldsymbol{\eta}(t)$ as efficiently as possible, given access to information about $\mathbf{r}(t)$. This is a standard denoising objective function, where an input signal is corrupted by additive Gaussian noise, with the following well-known solution [37]:

$$\Delta\mathbf{r}_1^*(t) = \sigma^2\frac{\mathrm{d}}{\mathrm{d}\mathbf{r}(t)}\log p(\mathbf{r}(t))\Delta t, \tag{A.13}$$

where $p(\mathbf{r}) = \int p(\mathbf{r}(t)|\mathbf{s}(t))p(\mathbf{s}(t))\mathrm{d}\mathbf{s}(t)$ is the probability distribution over noisy network states given input stimulus $\mathbf{s}(t)$, prior to the application of state updates $\Delta\mathbf{r}_1(t)$ and $\Delta\mathbf{r}_2(t)$. By assumption, $p(\mathbf{r}(t)|\mathbf{s}(t)) \sim \mathcal{N}(\mathbf{D}^\dagger f(\mathbf{s}(t)), \sigma^2\Delta t)$. We note that $\frac{\mathrm{d}}{\mathrm{d}\mathbf{r}(t)}\log p(\mathbf{r}(t))$ is the same function for all time points t, because for the stimulus sets we consider, $p(\mathbf{s}(t))$ does not depend on time (it is a stationary distribution); this demonstrates that the optimal greedy denoising update is not time-dependent. These dynamics move the network state towards states with higher probability, and do not require explicit access to noise information $\boldsymbol{\eta}(t)$.

Next, we optimize for $\Delta\mathbf{r}_2(t)$. $\mathcal{L}_{signal}$ is the only term in $\mathcal{L}_{upper}$ that depends on $\Delta\mathbf{r}_2(t)$, so we minimize:

$$\mathcal{L}_{signal} = \|\frac{\mathrm{d}f(\mathbf{s}(t))}{\mathrm{d}\mathbf{s}(t)}\Delta\mathbf{s}(t) - \mathbf{D}\Delta\mathbf{r}_2(t)\|_2. \tag{A.14}$$

By inspection, the optimum is given by: $\Delta\mathbf{r}_2^*(t) = \mathbf{D}^\dagger\frac{\mathrm{d}f(\mathbf{s}(t))}{\mathrm{d}\mathbf{s}(t)}\Delta\mathbf{s}(t)$. Thus the full greedily optimal dynamics, in the presence of noise, are given by:

$$\Delta\mathbf{r}^*(t) = \left[\sigma^2\frac{\mathrm{d}}{\mathrm{d}\mathbf{r}(t)}\log p(\mathbf{r}(t)) + \mathbf{D}^\dagger\frac{\mathrm{d}f(\mathbf{s}(t))}{\mathrm{d}\mathbf{s}(t)}\frac{\mathrm{d}\mathbf{s}(t)}{\mathrm{d}t}\right]\Delta t + \sigma\boldsymbol{\eta}(t). \tag{A.15}$$

This heuristic solution provides interpretability to any system attempting to maintain a relationship to a stimulus in the presence of noise. First, denoise the system ($\Delta\mathbf{r}_1^*$). Second, use instantaneous changes in the state variable ($\frac{\mathrm{d}\mathbf{s}(t)}{\mathrm{d}t}$) to update state information.

B Implementation Details

Spatial Position Estimation. In this task, the network must learn to path integrate motion cues in order to estimate an animal's spatial location in a 2D environment. We first generate an animal's motion trajectories $\mathbf{s}_{SP}(t)$ using a model described in prior work [38]. Next, we simulate the activities of n_{SP} place cells for all positions visited. The simulated place cells' receptive field centers $\mathbf{c}^{(i)}$ (where $i = 1, \ldots, n_{SP}$) are randomly and uniformly scattered across the 2D environment, and the activity of each for a position $\mathbf{s}$ is given by the following Gaussian tuning curve:

$$f_{SP}^{(i)}(\mathbf{s}) = \exp\left(-\frac{\|\mathbf{s} - \mathbf{c}^{(i)}\|_2^2}{2\sigma_{SP}^2}\right), \tag{B.1}$$

where σ_{SP} is the scale. We then train our network to output these simulated place cell activities based on velocity inputs ($\Delta\mathbf{s}_{SP}(t)$) from the simulated trajectories. To estimate the actual position in the environment from the network's outputs, we average the centers associated with the top k most active place cells. Our implementation is consistent with prior work [27, 39] and all task hyperparameters are listed in Table B.1.

Head Direction Estimation. The network's goal in this task is to estimate an animal's bearing $\mathbf{s}_{HD}(t)$ in space based on angular velocity cues $\Delta\mathbf{s}_{HD}(t)$, where $\mathbf{s}(t)$ is a 1-dimensional circular variable with domain $[-\pi, \pi)$. As in the previous task, we first generate random head rotation trajectories. The initial bearing is sampled from a uniform distribution $\mathcal{U}(-\pi, \pi)$, and random turns are sampled from a normal distribution $\mathcal{N}(0, 11.52)$—this is consistent with the trajectories used in the previous task, but we do not simulate any spatial information. We then simulate the activities of n_{HD} head direction cells whose preferred angles θ_i (where $i = 1, \ldots, n_{HD}$) are uniformly spaced between $-\pi$ and π, using an implementation similar to the RatInABox package [40]. The activity of the ith head direction cell for a bearing $\mathbf{s}$ is given by the following von Mises tuning curve:

$$f_{HD}^{(i)}(\mathbf{s}) = \frac{\exp\left(\sigma_{HD}^{-2}\cos\left(\mathbf{s} - \theta^{(i)}\right)\right)}{2\pi I_0\left(\sigma_{HD}^{-2}\right)}, \tag{B.2}$$

where σ_{HD} is the spread parameter for the von Mises distribution. With these simulated trajectories, we train the network to estimate the simulated head direction cell activities using angular velocity as input. We estimate the actual bearing from the network's outputs by taking the circular mean of the top k most active cells' preferred angles. All hyperparameters associated with this task are provided in Table B.2.

Continuous-time RNNs. For all our numerical experiments, we use noisy "vanilla" continuous-time RNNs with linear readouts. The equations for network updates and output estimates are as follows:

$$\Delta\mathbf{r}(t) = \frac{1}{\tau}\left[-\mathbf{r}(t) + \mathrm{ReLU}\left(\mathbf{W}^{rec}\mathbf{r}(t) + \mathbf{W}^{in}\Delta\mathbf{s}(t)\right)\right]\Delta t + \sigma\boldsymbol{\eta}(t) \tag{B.3}$$

$$\mathbf{o} = \mathbf{D}\mathbf{r}(t), \tag{B.4}$$

where $\mathbf{r}(t)$ represents the network activity at time t, $\Delta\mathbf{s}(t)$ is a change-based input to the network, $\mathbf{W}^{rec}$ and $\mathbf{W}^{in}$ are the recurrent and input weight matrices respectively, τ is the RNN time constant, and $\boldsymbol{\eta}(t) \sim \mathcal{N}(0, \Delta t)$ is Brownian noise. The continuous-time dynamics are approximated using the Euler-Maruyama method with integration timestep $\Delta t = 0.02$ s. The network's activity is transformed by a linear mapping $\mathbf{D}$ to predicted place cell or head direction cell activities $\mathbf{o}$. During the quiescent phase, we simulated network activity in the absence of stimuli ($\Delta\mathbf{s}(t) = 0$), and increased the noise variance by a factor of two, as prescribed by our mathematical analysis.

We tune the value of τ for each task to ensure optimal performance. Further, for each task we choose different training values for σ to scale the Brownian noise to establish an effective signal-to-noise ratio that is high enough to accurately solve the task. From this baseline noise level, quiescent trajectories were calculated with doubled variance.

Table B.1: Hyperparameters for the spatial position estimation task.

Category	Hyperparameter	Value
Task	Environment size	(2.2 m $\times$ 2.2 m)
	Border region	0.03 m
	Border slowdown factor	0.25
	Position initialization	random uniform
	Rotation velocity bias	0 rad/s
	Rotation velocity std. dev.	11.52 rad/s
	Rayleigh forward velocity	0.2 m/s
	Biasing anchor point	(0, 0)
	Biasing drift constant	0.05
	# place cells	512
	σ_{SP}	0.2
	Sequence length	active = 200; quiescent = 1000
	σ	$\frac{0.01}{\sqrt{0.02}} \approx 0.0707$
Network	# recurrent units	512
	τ	0.1
Training	Batch size	200
	# batches	2500
	Optimizer	Adam
	Learning rate	0.001

Table B.2: Hyperparameters for the head direction estimation task.

Category	Hyperparameter	Value
Task	Position initialization	random uniform
	Rotation velocity bias	0 rad/s
	Rotation velocity std. dev.	11.52 rad/s
	# head direction cells	512
	σ_{HD}	$\frac{\pi}{6}$
	Sequence length	active = 200; quiescent = 1000
	σ	$\frac{0.1}{\sqrt{0.02}} \approx 0.7071$
Network	# recurrent units	128
	τ	0.04
Training	Batch size	200
	# batches	20000
	Optimizer	Adam
	Learning rate	0.001

C Additional Experiments

C.1 Spatial Position Estimation

We found that the explained variance curves as a function of ordered principal components (PCs) for both the active and quiescent phases were highly overlapping and indicated that the activity manifold in both tasks was low-dimensional. The quiescent neural activity projected onto the first two PCs calculated during the active phase is smooth, just as with the decoded output trajectories (Fig. C.1a-b). Furthermore, the KDEs computed on neural activity projected onto the first two active phase PCs (Fig. C.1c-d) show that the distribution of activity was similar across active and quiescent phases, recapitulating the observations from Fig. 1.

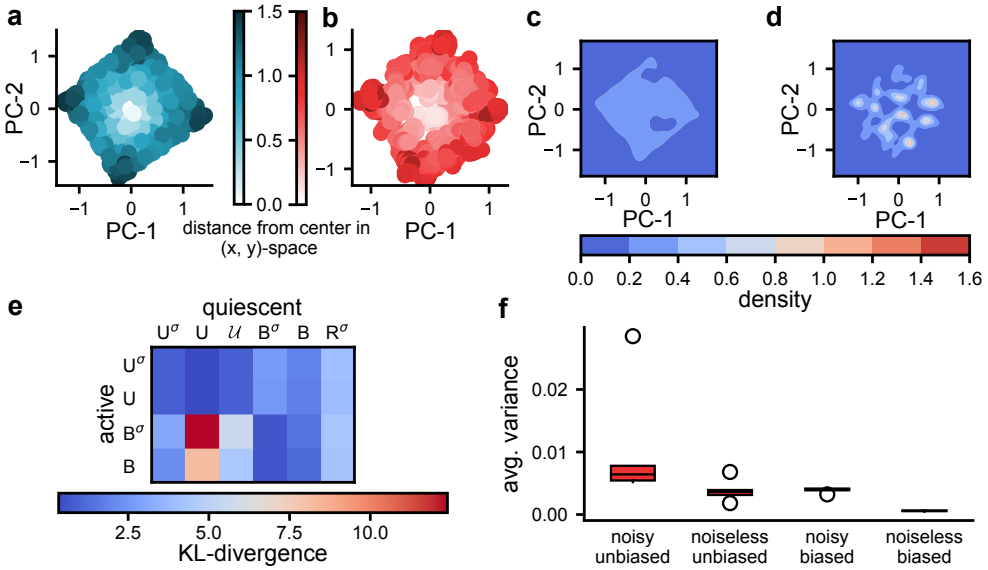

Figure C.1: **Neural activity manifold visualization and distribution comparisons. a)** Neural activity projected onto the first two PCs during the active phase. Color intensity measures the decoded output's distance from the center in space. **b)** Neural activity during the quiescent phase projected onto the same active PC axes as in (a). **c-d)** KDE plots for 500 decoded active (c) and quiescent (d) neural activity projected onto the first two active phase PCs. **e)** KL divergence (nats) between KDE estimates for active and quiescent phases. U = unbiased uniform networks, B = biased networks, $\mathcal{U}$ = the true uniform distribution, R = random networks, and the σ superscript denotes networks that are trained and tested in the presence of noise. Values are averaged over five trained networks. **f)** Box and whisker plots of the total variance (variance summed over output dimensions) of quiescent trajectories, averaged over 500 trajectories. Each plot (e-f) is for five trained networks.

To compare activity distributions more quantitatively, we estimated the KL-divergence of the distribution of active phase output positions to the distribution of quiescent phase decoded output positions using Monte Carlo approximation (Fig. C.1e). We compared outputs from both biased and unbiased distributions, and as baselines, we compared to a true uniform distribution, as well as decoded output trajectories generated by random networks. By our metric, we found that unbiased quiescent outputs were almost as close to unbiased active outputs as a true uniform distribution. Similarly, biased quiescent outputs closely resembled biased active outputs, while biased-to-unbiased, biased-to-random, and unbiased-to-random comparisons all diverged.

We decided to further test the necessity of training and generating quiescent network activity in the presence of noise. By the same KL divergence metric, we found that even trajectories generated by networks that were not trained in the presence of noise, and also were not driven by noise in the quiescent phase, still generated quiescent activity distributions that corresponded well to the active phase distributions. This is likely due to the fact that even networks trained in the absence of noise still learned attractive task manifolds that reflected the agent's trajectory sampling statistics. However, we found that networks without noise in the quiescent state exhibited less variable trajectories, as measured by their steady-state total variance (Fig. C.1f). This demonstrates that individual quiescent noiseless trajectories explored a smaller portion of the task manifold than did noisy trajectories (see Fig. C.2a-d for a comparison of example noisy and noiseless quiescent trajectories). This failure of exploration could not be resolved by adding additional noise to networks during the quiescent phase: we found that without training in the presence of noise, quiescent phase activity with an equivalent noise level generated erratic, non-smooth decoded output trajectories (Fig. C.2e-f). Therefore, noisy training stabilizes noisy quiescent activity, which in turn explores more of the task manifold than noiseless quiescent activity.

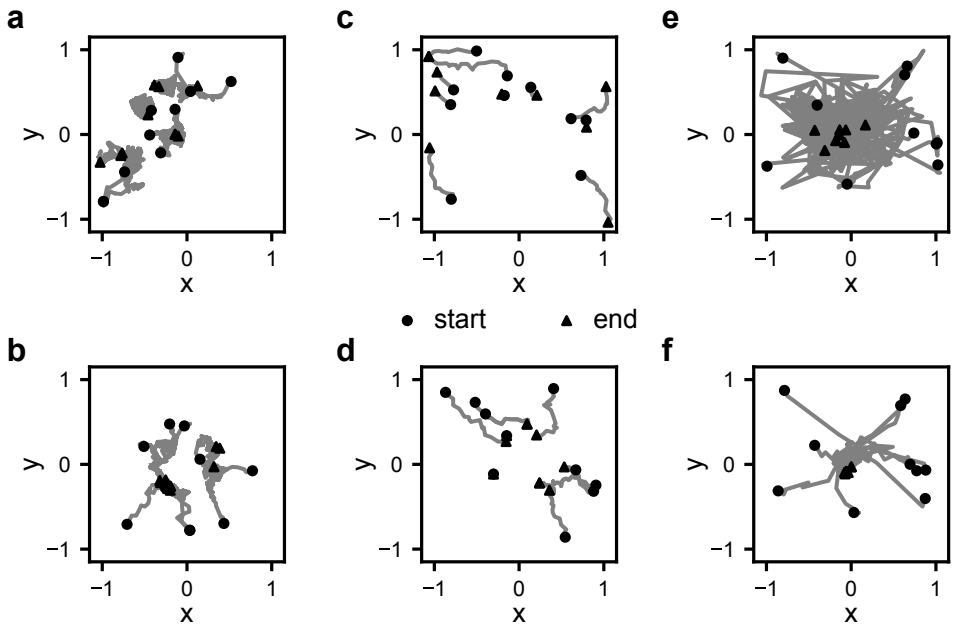

Figure C.2: **Example decoded trajectories under different noise conditions. a-b)** Example noisy quiescent trajectories for a network trained in the presence of noise, for the unbiased (a) and biased (b) spatial position estimation tasks. **c-d)** Same as (a-b), but for noiseless quiescent trajectories for a network trained without noise. **e-f)** Same as (a-b), but for noisy quiescent trajectories for a network trained without noise.

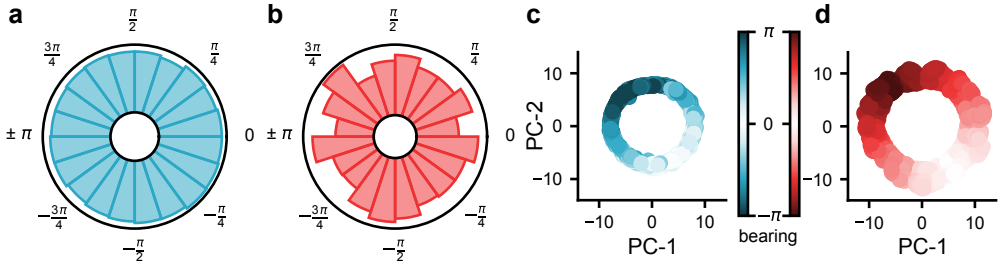

Figure C.3: **Reactivation in a head direction estimation task. a-b)** Distribution of decoded head direction bearing angles during the active (a) and quiescent (b) phases. **c-d)** Neural network activity projected onto the first two active phase PCs for active (c) and quiescent (d) phase trajectories. Color bars indicate the decoded output head direction.

C.2 Head Direction Estimation

To demonstrate that our results empirically extend beyond the spatial position estimation task, we also examined the reactivation phenomenon in the context of our head direction estimation task. Here, as in the previous task, we found that the distribution of decoded head direction bearings closely corresponded across the active and quiescent phases (Fig. C.3a-b). Furthermore, we found that the distributions of neural trajectories, projected onto the first two active phase PCs, closely corresponded across both phases (Fig. C.3c-d), showing apparent sampling along a ring attractor manifold. These results collectively demonstrate that reactivation also emerges from training our networks on our head direction estimation task.

D Applicability to other reactivation phenomena

While the experiments explored in this study focus on self-localization and head direction estimation, there are many more systems in which our results may be applicable. In particular, while the early visual system does not require sensory estimation from exclusively change-based information, denoising is a critical aspect of visual computation, having been used for deblurring, occlusion inpainting, and diffusion-based image generation [41]—the mathematical principles used for these applications are deeply related to those used to derive our denoising dynamics. As a consequence, it is possible that with further development our results could also be used to explain similar reactivation dynamics observed in the visual cortex [7, 8]. Furthermore, the task computations involved in head direction estimation are nearly identical to those used in canonical visual working memory tasks in neuroscience (both develop ring attractor structures) [42]. In addition, evidence integration in decision making involves similar state-variable integration dynamics as used in spatial navigation, where under many conditions the evidence in favor of two opposing decisions is integrated along a line attractor rather than a 2D spatial map [43, 44]. Thus our results could potentially be used to model reactivation dynamics observed in areas of the brain dedicated to higher-order cognition and decision making, such as the prefrontal cortex [6].