RNN Replay: Leakage and Underdamped Dynamics

Josue Casco-Rodriguez, Richard G. Baraniuk
Department of Electrical and Computer Engineering
Rice University

Abstract

One hallmark of neural processing is the ability to dream or replay realistic sequences without any input. Recent work shows that denoising recurrent neural networks (RNNs) implicitly learn the score function of their hidden states, and can thus dream of realistic sequences via Langevin sampling. However, the current theory of Langevin sampling in RNNs fails to identify the nature of the score function, the impact of architectural choices like leakage and adaptation, or how to improve Langevin sampling in RNNs. We rectify these failures by: (1) using Markov Gaussian processes to explain how the score function can be difficult to approximate, but admits a form that readily incorporates leakage; (2) show that adaptation induces a form of underdamped Langevin sampling; and (3) propose a more direct and effective form of underdamped Langevin sampling for RNNs.

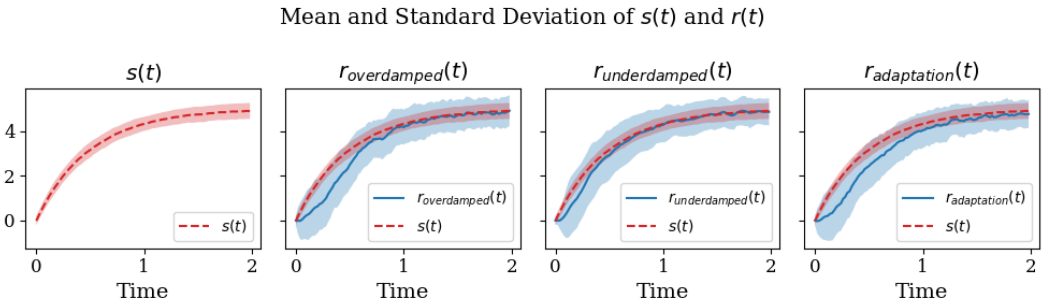

Figure 1: **Underdamped Langevin sampling accelerates offline replay.** Here we examine a linear RNN $r(t)$ that denoises an Ornstein-Uhlenbeck process $s(t)$ at each timestep t. Reconstruction of $s(t)$ entails that the RNN learn its own score function $\frac{d}{dr(t)} \log p(r(t))$, but in this case we can directly use the optimal score function from Equation 12. We evaluate three different ways of sampling from $p(r(t))$ when the RNN only observes noise at each timestep: the default *overdamped* (1st-order) sampling method, our new proposed *underdamped* (2nd-order) sampling method, and the recently proposed *adaptation* sampling method. We see that underdamped sampling is accelerates $r(t)$ towards the true statistics of $s(t)$, thus producing the highest-fidelity samples of $p(r(t))$ relative to $p(s(t))$. See Figure 3 for metrics.

1 Introduction

Recurrent neural networks (RNNs) exist with varying degrees of biological plausibility; one degree of plausibility comes from introducing noise to the hidden units of RNNs. Krishna et al. [1] recently proved that RNNs trained to reconstruct their inputs despite the presence of input noise (a procedure related to predictive coding [2]) have the capability to perform Langevin sampling over the input

Preprint. Under review.

distribution when no external input is given—otherwise known as *offline replay*, reactivation, or dreaming. Krishna et al. [1], Levenstein et al. [3] study offline replay in the context of navigation; one interesting observation from the latter is that introducing adaptation, or second-order dynamics, into the RNN increases the faithfulness of offline replay.

However, several problems exist with the current theories of offline replay and Langevin sampling in RNNs: **(1)** The relationship between the learned input distribution, RNN architectures, offline Langevin sampling is unclear: current understandings of Langevin sampling in RNNs seem limited by their assumption of stationary input statistics [1], and cannot fully explain the role of even simple architectural parameters like leakage [1, 3]. **(2)** Other works have explored the utility of adaptation [3–5], but not in the explicit context of Langevin sampling theory. **(3)** Modern Langevin-based generative models, like diffusion models, have found success in modifying Langevin dynamics [6], but it remains unclear whether RNNs could benefit from novel sampling techniques.

1.1 Contributions

1. We show that the RNN score function is non-stationary even under very simple Gaussian conditions, but these conditions reveal how linear leakage is integral to offline Langevin sampling.
2. We show that adaptation induces a form of second-order Langevin sampling in offline RNNs.
3. We introduce a novel, straightforward offline sampling scheme for RNNs that requires no additional training and is better at offline sampling from simple Gaussian input distributions.

2 Setup

2.1 Langevin Dynamics

Langevin dynamics sample from a distribution $p(x)$ through an iterate $x(t)$ that evolves as follows:

$$x'(t) = x(t) + \nabla \log p(x) + \sigma\eta(t), \tag{1}$$

where $\eta(t)$ is Gausian white noise. The dynamics in Equation 1 are also known as *overdamped* Langevin dynamics, since there also exist *underdamped* Langevin dynamics that converge much faster to the target distribution $p(x)$ and are better at utilizing noisy gradients [7]. Underdamped Langevin dynamics can be written equivalently as Equation 2 or 3 (Chapter 6 from Pavliotis [8]):

$$x''(t) = \nabla \log p(x) - \gamma x'(t) + \sqrt{2\gamma}\eta(t), \quad \text{or} \tag{2}$$

$$x'(t) = v(t), \quad v'(t) = \nabla \log p(x) - \gamma v(t) + \sqrt{2\gamma}\eta(t) \tag{3}$$

2.2 Offline Replay in RNNs

The proofs for this section can be found in Appendix B.

Suppose an RNN has a vector of neural responses $\boldsymbol{r}(t)$ whose evolution over time can be decomposed into three separate update functions: $\Delta\boldsymbol{r}_1(t), \Delta\boldsymbol{r}_2(t)$, and Gaussian noise $\sigma_r\boldsymbol{\eta}(t) \sim \mathcal{N}(0, \sigma_r^2\Delta t)$:

$$\boldsymbol{r}(t + \Delta t) - \boldsymbol{r}(t) \equiv \Delta\boldsymbol{r}(t) \equiv \Delta\boldsymbol{r}_1(t) + \Delta\boldsymbol{r}_2(t) + \sigma_r\boldsymbol{\eta}(t) \tag{4}$$

Krishna et al. [1] found that, if an RNN with activations $\boldsymbol{r}(t)$ minimizes the mean-squared error $\mathcal{L}(t)$ between a projection of its activations and an observation $f(\boldsymbol{s}(t))$ of the environment process $\boldsymbol{s}(t)$, then $\mathcal{L}(t)$ has an upper bound consisting of signal-tracking and denoising components:

$$\mathcal{L}(t) \equiv \boldsymbol{E}_{\boldsymbol{\eta}}\|f(\boldsymbol{s}(t)) - \boldsymbol{D}\boldsymbol{r}(t)\|_2 \leq \mathcal{L}_{signal}(\Delta\boldsymbol{r}_2(t)) + \mathcal{L}_{noise}(\Delta\boldsymbol{r}_1(t)) \tag{5}$$

Successful minimzation of $\mathcal{L}_{signal}, \mathcal{L}_{noise}$ induces the following distribution of $\boldsymbol{r}(t)$, which can be used to find the optimal updates $\Delta\boldsymbol{r}_1(t), \Delta\boldsymbol{r}_2(t)$:

$$p(\boldsymbol{r}(t)|\boldsymbol{s}(t)) \sim \mathcal{N}(\boldsymbol{D}^\dagger f(\boldsymbol{s}(t)),\, \boldsymbol{I}\sigma_r^2\Delta t) \tag{6}$$

$$\Delta\boldsymbol{r}_1^*(t) = \sigma_r^2 \frac{d}{d\boldsymbol{r}(t)} \log p(\boldsymbol{r}(t))\Delta t, \quad \Delta\boldsymbol{r}_2^*(t) = \boldsymbol{D}^\dagger(f \circ \boldsymbol{s})'(t)\Delta t \tag{7}$$

Finally, Krishna et al. [1] observe that, if the RNN is devoid of input ($\boldsymbol{s}'(t) = \boldsymbol{0}$) and the variance of the input noise is scaled by 2, then the RNN performs Langevin sampling of $p(\boldsymbol{r}(t))$ (Equation 1):

$$\Delta\tilde{\boldsymbol{r}}(t) = \sigma_r^2 \frac{d}{d\boldsymbol{r}(t)} \log p(\boldsymbol{r}(t))\Delta t + \sqrt{2}\sigma_r\boldsymbol{\eta}(t), \quad p(\tilde{\boldsymbol{r}}(t)) \approx p(\boldsymbol{r}(t)) \tag{8}$$

3 Demystifying the RNN Score Function

Previous work on RNNs and Langevin sampling has assumed that the RNN can learn the *score function* $\frac{d}{d\boldsymbol{r}(t)} \log p(\boldsymbol{r}(t))$ using only $\boldsymbol{r}(t)$ [1]. However, we refute this assumption and in doing so reveal the role of leakage in Langevin sampling: even if the RNN simply observes a Gaussian process, the score function of $\boldsymbol{r}(t)$ requires information beyond just $\boldsymbol{r}(t)$, but the manner in which it uses that information is through linear leakage of $\boldsymbol{r}(t)$.

General Case and Leakage. In a general case where we merely constrain $p(f(\boldsymbol{s}(t)))$ to be Gaussian—with no assumptions over how $p(f(\boldsymbol{s}(t)))$ changes over time—we can already see that the score function of $\boldsymbol{r}(t)$ is simple with respect to $\boldsymbol{r}(t)$, but complex with respect to the parameters of $p(f(\boldsymbol{s}(t)))$. In Appendix A.2, we show that if $p(f(\boldsymbol{s}(t))) \sim \mathcal{N}(\boldsymbol{\mu}_{f(\boldsymbol{s}(t))}, \boldsymbol{\Sigma}_{f(\boldsymbol{s}(t))})$, then:

$$\frac{d}{d\boldsymbol{r}(t)} \log p(\boldsymbol{r}(t)) = -\left(\boldsymbol{I}\sigma_r^2 \Delta t + \boldsymbol{D}^\dagger \boldsymbol{\Sigma}_{f(\boldsymbol{s}(t))} (\boldsymbol{D}^\dagger)^T\right)^{-1} \left(\boldsymbol{r}(t) - \boldsymbol{D}^\dagger \boldsymbol{\mu}_{f(\boldsymbol{s}(t))}\right) \tag{9}$$

With $\boldsymbol{\Lambda}(t) = \sigma_r^2 \Delta t \left(\boldsymbol{I}\sigma_r^2 \Delta t + \boldsymbol{D}^\dagger \boldsymbol{\Sigma}_{f(\boldsymbol{s}(t))} (\boldsymbol{D}^\dagger)^T\right)^{-1}$ as the *leakage matrix* of $\Delta\tilde{\boldsymbol{r}}(t)$ (Equation 8), we can already observe three insights from Equation 9:

1. **Linear Leakage.** If $p(f(\boldsymbol{s}(t)))$ is Gaussian, then the score function of $\boldsymbol{r}(t)$ simply requires linear leakage of $\boldsymbol{r}(t)$ and addition of a bias vector. However, the leakage and bias can be nonlinear functions of time.
2. **Bounded, Interpretable Eigenvalues.** The eigenvalues of the leakage matrix $\boldsymbol{\Lambda}(t)$ of are always between 0 and 1 (see Appendix A.2), validating leakage parameterizations in biologically plausible RNNs. Moreover, the eigenvalues of $\boldsymbol{\Lambda}(t)$ and $\boldsymbol{D}^\dagger \boldsymbol{\Sigma}_{f(\boldsymbol{s}(t))} (\boldsymbol{D}^\dagger)^T$ inversely correlate.
3. **Time-Varying Complexity.** Although the score function of $\boldsymbol{r}(t)$ may be linear with respect to $\boldsymbol{r}(t)$, it is nonlinear with respect to the covariance of $f(\boldsymbol{s}(t))$, and only as stationary as $p(f(\boldsymbol{s}(t)))$.

Challenges. To concretely illustrate how estimating the score function of $\boldsymbol{r}(t)$ (Equation 9) can be challenging, we now turn to Wiener and Ornstein-Uhlenbeck processes. For clarity, let us suppose $\boldsymbol{s}(t) = s(t)$ is a scalar process with Ornstein-Uhlenbeck dynamics and parameters θ, μ, σ_s:

$$s'(t) = \theta(\mu - s(t)) + \sigma_s \eta(t) \tag{10}$$

Let us also suppose that $f(s(t)) = s(t)$, $\boldsymbol{D} = 1$, and $\boldsymbol{r}(t) = r(t)$ is also a scalar function.

Wiener Processes. One simple stochastic process is the Wiener process, which in terms of navigation represents an undirected random walk ($\theta = 0$). Assuming that $s(0) = 0$, then $s(t) \sim \mathcal{N}(0, \sigma_s^2 t)$, and therefore $p(r(t)) \sim \mathcal{N}(0, \sigma_s^2 t + \sigma_r^2 \Delta t)$ from Equation 6, producing the following $\Delta r_1^*(t)$:

$$\Delta r_1^*(t) = \sigma_r^2 \Delta t \frac{-r(t)}{\sigma_s^2 t + \sigma_r^2 \Delta t} \tag{11}$$

Even from a simple Wiener process, we observe that the per-timestep optimal denoiser $\Delta r_1^*(t)$ is not constant with respect to t: $\lim_{t\to 0} \Delta r_1^*(t) = -r(t)$, while $\lim_{t\to\infty} \Delta r_1^*(t) = 0$.

Ornstein-Uhlenbeck Processes. Now we incorporate non-zero leakage ($\theta > 0$) to model a directed random walk navigating from an arbitrary starting point $s(0)$ towards a mean destination μ. If $p(s(0)) \sim \mathcal{N}(0, \sigma_0^2)$, then $p(s(t)) \sim \mathcal{N}\left(\mu(1 - e^{-\theta t}), \frac{\sigma_s^2}{2\theta}(1 - e^{-2\theta t}) + \sigma_0^2 e^{-\theta t}\right)$, and $\Delta r_1^*(t)$ is:

$$\Delta r_1^*(t) = \sigma_r^2 \Delta t \frac{-(r(t) - \mu(1 - e^{-\theta t}))}{\frac{\sigma_s^2}{2\theta}(1 - e^{-2\theta t}) + \sigma_0^2 e^{-\theta t} + \sigma_r^2 \Delta t} \tag{12}$$

The optimal correction Δr_1^* is evidently quite complex and not stationary: $\lim_{t\to 0} \Delta r_1^*(t) = -\frac{\sigma_r^2 \Delta t}{\sigma_r^2 \Delta t + \sigma_0^2} r(t)$, while $\lim_{t\to\infty} \Delta r_1^*(t) = -\frac{\sigma_r^2 \Delta t}{\sigma_r^2 \Delta t + \sigma_s^2/2\theta}(r(t) - \mu)$. While one could try to deem $\Delta r_1^*(t)$ as "stationary" by implicitly assuming that the process starts at $s(0) = \mu$, or assuming that the steady-state dynamics ($t \to \infty$) are the most important, we argue that any such approach would miss a fundamentally relevant aspect of the Ornstein-Uhlenbeck process for navigation: *intention*. Unlike the Wiener process, the Ornstein-Uhlenbeck process can describe a random walk that *intentionally* navigates from $s(0)$ to μ, rather than one that simply wanders around μ. Thus, for navigation, the non-stationary, or "early", dynamics of the Ornstein-Uhlenbeck process are the most salient.

4 Adaptation as Underdamped Langevin Dynamics

Definition. Now we add a linear *adaptation* term [3–5] to Equation 4, whose role could be explained via an auxiliary term on $\mathcal{L}(t)$ (Equation 5) that minimizes the exponential moving average of $\boldsymbol{r}(t)$:

$$\boldsymbol{r}(t+\Delta t) = \boldsymbol{r}(t) + \Delta\boldsymbol{r}_1(t) + \Delta\boldsymbol{r}_2(t) + \sigma_r\boldsymbol{\eta}(t) - \boldsymbol{c}(t), \tag{13}$$

$$\boldsymbol{c}(t+\Delta t) = \boldsymbol{c}(t) + \frac{1}{\tau_a}(-\boldsymbol{c}(t) + b_a\boldsymbol{r}(t)), \tag{14}$$

$$\mathcal{L}_a(t) \equiv \mathcal{L}(t) + \frac{1}{2}\|\boldsymbol{c}(t)\|_2^2 \tag{15}$$

Interpretation as Underdamped Langevin Dynamics. Let us now focus on the effects of adaptation in the contexts of offline replay and Langevin dynamics. After successful minimization of $\mathcal{L}_a(t)$, the optimal update $\Delta\tilde{\boldsymbol{r}}(t+\Delta t)$ in the absence of changing $f(\boldsymbol{s}(t))$ is (from Equation 8):

$$\Delta\tilde{\boldsymbol{r}}(t+\Delta t) = \sigma_r^2\frac{d}{d\boldsymbol{r}(t)}\log p(\boldsymbol{r}(t))\Delta t - \boldsymbol{c}(t) + \sqrt{2}\sigma_r\boldsymbol{\eta}(t) \tag{16}$$

For the clearest illustration of the effects of adaptation, let us examine a stationary $p(\boldsymbol{r}(t)) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$—a simplification which we have argued earlier is not realistic, but is nonetheless intuitive. Then, the equation above becomes (see Appendix A):

$$\Delta\tilde{\boldsymbol{r}}(t+\Delta t) = \sigma_r^2\Delta t\boldsymbol{\Sigma}^{-1}(-\boldsymbol{r}(t) + \boldsymbol{\mu}) - \boldsymbol{c}(t) + \sqrt{2}\sigma\boldsymbol{\eta}(t), \tag{17}$$

with which we prove in Appendix C that $\boldsymbol{c}(t)$ induces these second-order stochastic dynamics:

$$\begin{aligned}\boldsymbol{r}''(t) = \left(\frac{b_a}{\tau_a}\boldsymbol{I} + \sigma_r^2\Delta t\frac{d^2}{d\boldsymbol{r}(t)^2}\log p(\boldsymbol{r}(t))\right)\boldsymbol{r}'(t) - \frac{b_a}{\tau_a}\sigma_r^2\Delta t\frac{d}{d\boldsymbol{r}(t)}\log p(\boldsymbol{r}(t)) + \frac{1}{\tau_a}\boldsymbol{r}(t) \\ -\sigma\frac{b_a}{\tau_a}\boldsymbol{\eta}(t) + \sigma\boldsymbol{\eta}'(t)\end{aligned} \tag{18}$$

Comparing Equation 18 with Equation 2, the two indeed resemble each other: adaptation seems to induce a form of underdamped Langevin dynamics. This may help to explain the observed utility of adaptation for offline sampling from $p(\boldsymbol{r})$ [3–5]. Moreover, since we established in Section 3 that the score function of even a basic stochastic process is difficult to estimate, the effectiveness of underdamped Langevin sampling for working with noisy gradients [7] may be useful in realistic settings where $\frac{d}{d\boldsymbol{r}(t)}\log p(\boldsymbol{r}(t))$ is poorly estimated.

5 Offline Replay via Explicit Underdamped Langevin Dynamics

Shortcomings of Adaptation. The parameters of underdamped Langevin dynamics prescribed in Equations 13 and 14 may undermine its efficacy as a sampling method. A few concerns include:

1. The coefficient of $\boldsymbol{r}'(t)$ is usually constant, and should be positive to ensure convergence (Pavliotis [8], pg. 183), but $\frac{b_a}{\tau_a}\boldsymbol{I} + \sigma_r^2\Delta t\frac{d^2}{d\boldsymbol{r}(t)^2}\log p(\boldsymbol{r}(t))$ is not constant and could be negative.
2. Underdamped Langevin sampling from $\boldsymbol{r}(t)$ should not have a negative sign in front of $\frac{d}{d\boldsymbol{r}(t)}\log p(\boldsymbol{r}(t))$ if the intention is to maximize $p(\boldsymbol{r}(t))$.
3. If $\frac{1}{\tau_a}, \frac{b_a}{\tau_a} \ll 1$ (as they are in Levenstein et al. [3], Itskov et al. [4], Levenstein et al. [5]), then the dynamics of $\boldsymbol{r}(t)$ should largely obey first-order Langevin dynamics.

While adaptation is a biologically plausible mechanism of performing underdamped Langevin dynamics in RNNs, we propose an alternative method, following Equation 3, to more clearly and directly perform underdamped Langevin sampling via $\Delta\boldsymbol{r}_1^*(t) = \sigma_r^2\Delta t\frac{d}{d\boldsymbol{r}(t)}\log p(\boldsymbol{r}(t))$:

$$\boldsymbol{r}(t+\Delta t) = \boldsymbol{r}(t) + \boldsymbol{v}(t), \quad \boldsymbol{v}(t+\Delta t) = \boldsymbol{v}(t) + \Delta\boldsymbol{r}_1^*(t) - \lambda\boldsymbol{v}(t) + \sqrt{2\lambda}\sigma_r\boldsymbol{\eta}(t) \tag{19}$$

In practice, instead of calculating $\boldsymbol{r}(t+\Delta t), \boldsymbol{v}(t+\Delta t)$ simultaneously from $\boldsymbol{r}(t), \boldsymbol{v}(t)$ in Equation 19, we calculate $\boldsymbol{r}(t+\Delta t)$ from $\boldsymbol{v}(t)$ and then use $\boldsymbol{r}(t+\Delta t)$ to calculate $\boldsymbol{v}(t+\Delta t)$ (i.e., symplectic Euler discretization [9]) for stability.

Experiments. To evaluate the efficacy of Equation 19 as an underdamped Langevin sampling method for RNNs, we compare it to overdamped sampling (Equation 8) and sampling with adaptation (Equations 13 and 14). We evaluate these three sampling methods with respect to sampling from 1D and 2D Ornstein-Uhlenbeck processes with constant means (see Appendix D for details). In Figures 2 and 3, we see that our proposed underdamped sampling method steadily outperforms overdamped and adaptation sampling with respect to Wasserstein distance from $p(\boldsymbol{s}(t))$, and is fairly robust to the choice of λ. These findings and the behavior of overdamped, underdamped, and adaptation sampling are summarized by Figure 1: underdamped sampling accelerates the trajectory of $\boldsymbol{r}(t)$ in the direction of $p(\boldsymbol{s}(t))$, overdamped sampling is a baseline, and adaptation sampling slows the evolution of $\boldsymbol{r}(t)$.

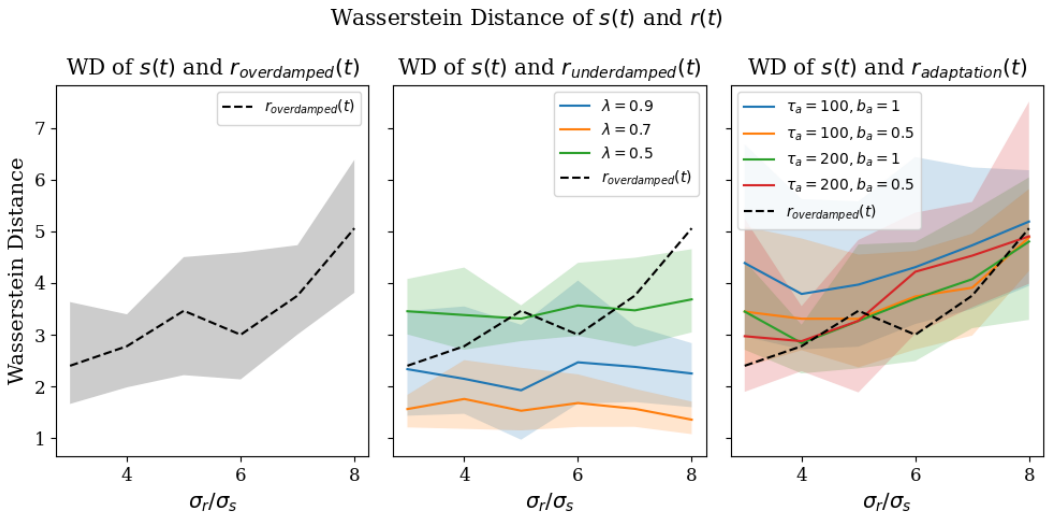

Figure 2: **Underdamped Langevin sampling improves the realism of offline replay.** Plotted above are the mean (solid or dashed), minimum, and maximum Wasserstein distances between $\boldsymbol{s}(t)$ and $\boldsymbol{r}_{overdamped}(t), \boldsymbol{r}_{underdamped}(t), \boldsymbol{r}_{adaptation}(t)$ when $\boldsymbol{s}(t)$ is a 2D Ornstein-Uhlenbeck process and the score function $\frac{d}{d\boldsymbol{r}(t)} \log p(\boldsymbol{r}(t))$ is estimated by a linear RNN (see Appendix D for details). As reflected in Figure 1, our underdamped sampling (Equation 19) outperforms overdamped and adaptation sampling (Equations 8, 13, 14), and is robust with respect to λ and σ_r. For completeness, some samples from $\boldsymbol{s}(t)$ and the various $\boldsymbol{r}(t)$ are in Figure 4.

6 Conclusions and Future Work

We have shown that: (1) the per-timestep score function of even a Markov Gaussian process can be challenging to estimate, but does reveal the utility of linear leakage; (2) adaptation induces a variant of underdamped Langevin sampling; and (3) our proposed underdamped Langevin sampling seems effective for offline replay in RNNs. Future work should compare the efficacy of overdamped, underdamped, and adaptation sampling in environments where the mean $\boldsymbol{\mu}_t$ of an Ornstein-Uhlenbeck process is non-stationary or $p(f(\boldsymbol{s}(t))$ is non-Gaussian (e.g., the experiments of Krishna et al. [1]). Furthermore, in the context of ReLU RNNs, a closer study into the efficacy of adaptation compared to leakage alone is warranted [1, 3]. Lastly, a close examination of other proposed offline replay techniques in RNNs (e.g., *rollout* training [3]), as well as an inspection of the potential utility of selective RNNs [10] for modeling the score of $\boldsymbol{r}(t)$ (Equation 9), would be interesting.

7 Acknowledgements

This work was supported by NSF grants CCF-1911094 and IIS-1730574; ONR grants N00014-23-1-2714, N00014-24-1-2225, and MURI N00014-20-1-2787; AFOSR grant FA9550-22-1-0060; DOE grant DE-SC0020345; DOI grant 140D0423C0076; and a Vannevar Bush Faculty Fellowship, ONR grant N00014-18-1-2047.

References

[1] Nanda H Krishna, Colin Bredenberg, Daniel Levenstein, Blake Aaron Richards, and Guillaume Lajoie. Sufficient conditions for offline reactivation in recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2024.

[2] Yanping Huang and Rajesh PN Rao. Predictive coding. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2(5), 2011.

[3] Daniel Levenstein, Aleksei Efremov, Roy Henha Eyono, Adrien Peyrache, and Blake Richards. Sequential predictive learning is a unifying theory for hippocampal representation and replay. *bioRxiv*, 2024.

[4] Vladimir Itskov, Carina Curto, Eva Pastalkova, and György Buzsáki. Cell assembly sequences arising from spike threshold adaptation keep track of time in the hippocampus. *Journal of Neuroscience*, 31(8), 2011.

[5] Daniel Levenstein, György Buzsáki, and John Rinzel. NREM sleep in the rodent neocortex and hippocampus reflects excitable dynamics. *Nature communications*, 10(1), 2019.

[6] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[7] Xiang Cheng, Niladri S. Chatterji, Peter L. Bartlett, and Michael I. Jordan. Underdamped Langevin MCMC: A non-asymptotic analysis. In *Proceedings of the 31st Conference On Learning Theory*, volume 75, 2018.

[8] Grigorios A Pavliotis. Stochastic processes and applications. *Texts in Applied Mathematics*, 60, 2014.

[9] Ernst Hairer, Christian Lubich, and Gerhard Wanner. Structure-preserving algorithms for ordinary differential equations. *Geometric numerical integration*, 31, 2006.

[10] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

[11] Martin Raphan and Eero P Simoncelli. Least squares estimation without priors or supervision. *Neural Computation*, 23(2), 2011.

A Score Functions of Gaussian Distributions

For any matrix calculus involved, we use denominator layout.

A.1 Multivariate Gaussian Distribution

Let's suppose $\boldsymbol{r} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. If $\boldsymbol{r} \in \mathbb{R}^d$, then:

$$p(\boldsymbol{r}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{r}-\boldsymbol{\mu})\right) \tag{20}$$

$$\log p(\boldsymbol{r}) \propto -\frac{1}{2}(\boldsymbol{r}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{r}-\boldsymbol{\mu}) \tag{21}$$

$$\frac{d}{d\boldsymbol{r}} \log p(\boldsymbol{r}) = -\frac{1}{2}((\boldsymbol{\Sigma}^{-1})^T + \boldsymbol{\Sigma}^{-1})(\boldsymbol{r}-\boldsymbol{\mu}) \tag{22}$$

$$= -\boldsymbol{\Sigma}^{-1}(\boldsymbol{r}-\boldsymbol{\mu}) \tag{23}$$

$$= -\sigma^{-2}(r-\mu) \text{ if } r \in \mathbb{R} \tag{24}$$

A.2 Score Function of $r(t)$ for Gaussian $f(s(t))$

Recall that $p(\boldsymbol{r}(t)|\boldsymbol{s}(t)) \sim \mathcal{N}(\boldsymbol{D}^\dagger f(\boldsymbol{s}(t)),\ \boldsymbol{I}\sigma_r^2 \Delta t)$ from Equation 6. If we suppose that $f(\boldsymbol{s}(t))$ is normally distributed with mean and covariance $\boldsymbol{\mu}_{f(\boldsymbol{s}(t))}, \boldsymbol{\Sigma}_{f(\boldsymbol{s}(t))}$, then we can obtain $p(\boldsymbol{r}(t))$:

$$p(\boldsymbol{r}(t)) \sim \mathcal{N}(\boldsymbol{D}^\dagger \boldsymbol{\mu}_{f(\boldsymbol{s}(t))},\ \boldsymbol{I}\sigma_r^2 \Delta t + \boldsymbol{D}^\dagger \boldsymbol{\Sigma}_{f(\boldsymbol{s}(t))} (\boldsymbol{D}^\dagger)^T), \tag{25}$$

which we can plug into Equation 23 to get $\frac{d}{d\boldsymbol{r}(t)} \log p(\boldsymbol{r}(t))$:

$$\frac{d}{d\boldsymbol{r}(t)} \log p(\boldsymbol{r}(t)) = -\left(\boldsymbol{I}\sigma_r^2 \Delta t + \boldsymbol{D}^\dagger \boldsymbol{\Sigma}_{f(\boldsymbol{s}(t))} (\boldsymbol{D}^\dagger)^T\right)^{-1} \left(\boldsymbol{r}(t) - \boldsymbol{D}^\dagger \boldsymbol{\mu}_{f(\boldsymbol{s}(t))}\right) \tag{26}$$

Moreover, we can use the above score function to calculate $\Delta \boldsymbol{r}^*(t+\Delta t)$ from Equation 44:

$$\begin{aligned} \Delta \boldsymbol{r}^*(t+\Delta t) = \sigma_r^2 \Delta t \left(\boldsymbol{I}\sigma_r^2 \Delta t + \boldsymbol{D}^\dagger \boldsymbol{\Sigma}_{f(\boldsymbol{s}(t))} (\boldsymbol{D}^\dagger)^T\right)^{-1} \left(-\boldsymbol{r}(t) + \boldsymbol{D}^\dagger \boldsymbol{\mu}_{f(\boldsymbol{s}(t))}\right) \\ + \boldsymbol{D}^\dagger (f \circ \boldsymbol{s})'(t) \Delta t + \sigma \boldsymbol{\eta}(t) \end{aligned} \tag{27}$$

Some properties of the leakage matrix $\sigma_r^2 \Delta t \left(\boldsymbol{I}\sigma_r^2 \Delta t + \boldsymbol{D}^\dagger \boldsymbol{\Sigma}_{f(\boldsymbol{s}(t))} (\boldsymbol{D}^\dagger)^T\right)$ include:

1. The covariance matrix $\boldsymbol{\Sigma}_{f(\boldsymbol{s}(t))}$ is positive semidefinite (PSD): all its eigenvalues are ≥ 0.
2. $\boldsymbol{D}^\dagger \boldsymbol{\Sigma}_{f(\boldsymbol{s}(t))} (\boldsymbol{D}^\dagger)^T$ is also PSD [1].
3. The eigenvalues of $(\boldsymbol{I}\sigma_r^2 \Delta t + \boldsymbol{D}^\dagger \boldsymbol{\Sigma}_{f(\boldsymbol{s}(t))} (\boldsymbol{D}^\dagger)^T)^{-1}$ are thus all $\leq (\sigma_r^2 \Delta t)^{-1}$ [2].
4. The eigenvalues of $\sigma_r^2 \Delta t (\boldsymbol{I}\sigma_r^2 \Delta t + \boldsymbol{D}^\dagger \boldsymbol{\Sigma}_{f(\boldsymbol{s}(t))} (\boldsymbol{D}^\dagger)^T)^{-1}$ are thus all ≤ 1.
5. If the off-diagonal entries of the leakage matrix above are sufficiently small in magnitude, then all the diagonal entries should be less than 1, as justified by the Gershgorin Circle Theorem. In fact, if the leakage matrix is diagonal, then it must have all values less than 1 (which could be achieved via sigmoid functions or perhaps spectral normalization).
6. As for interpretation, smaller leakage eigenvalues means higher eigenvalues of $\boldsymbol{D}^\dagger \boldsymbol{\Sigma}_{f(\boldsymbol{s}(t))} (\boldsymbol{D}^\dagger)^T$, or essentially, more noise. The maximum determinant of the leakage matrix is 1, when there is essentially no noise in $f(\boldsymbol{s}(t))$.

B Sufficient Conditions for Offline Reactivation in RNNs

The following proof is from Krishna et al. [1].

[1] Proof: If $\boldsymbol{B}$ is PSD, then $x^T \boldsymbol{A}\boldsymbol{B}\boldsymbol{A}^T x = (\boldsymbol{A}^T x)^T \boldsymbol{B} (\boldsymbol{A}^T x) = v^T \boldsymbol{B} v \geq 0$.

[2] Proof: For diagonalizable $\boldsymbol{A}$, the i-th eigenvalue of $(\lambda \boldsymbol{I} + \boldsymbol{A})^{-1} = (\boldsymbol{Q}(\lambda \boldsymbol{I} + \boldsymbol{\Lambda})\boldsymbol{Q}^{-1})^{-1}$ is equal to $(\lambda + \boldsymbol{\Lambda}_{ii})^{-1}$, which can be no larger than λ^{-1} if $\boldsymbol{A}$ is PSD.

For $\boldsymbol{s}(t)$, let $\boldsymbol{s}'(t) = \frac{d\boldsymbol{s}(t)}{dt}$. Equivalently, for $f(\boldsymbol{s}(t))$, let $(f \circ \boldsymbol{s})'(t) = \frac{df(\boldsymbol{s}(t))}{dt} = \frac{df(\boldsymbol{s}(t))}{d\boldsymbol{s}(t)}\boldsymbol{s}'(t)$.

$$\begin{aligned}
\boldsymbol{r}(t + \Delta t) &\equiv \boldsymbol{r}(t) + \Delta\boldsymbol{r}(t), && (28)\\
\Delta\boldsymbol{r}(t) &\equiv \phi\left(\boldsymbol{r}(t), \boldsymbol{s}(t), \boldsymbol{s}'(t)\right) + \sigma_r\boldsymbol{\eta}(t) && (29)\\
&\equiv \Delta\boldsymbol{r}_2\left(\boldsymbol{r}(t), \boldsymbol{s}(t), \boldsymbol{s}'(t)\right) + \Delta\boldsymbol{r}_1(\boldsymbol{r}(t)) + \sigma_r\boldsymbol{\eta}(t) && (30)\\
&\equiv \Delta\boldsymbol{r}_2(t) + \Delta\boldsymbol{r}_1(t) + \sigma_r\boldsymbol{\eta}(t), && (31)
\end{aligned}$$

where $\boldsymbol{\eta}(t) \sim \mathcal{N}(0, \Delta t)$.

Krishna et al. [1] aim to find an upper bound on the following loss:

$$\mathcal{L}(t) \equiv \mathbb{E}_{\boldsymbol{\eta}}\|f(\boldsymbol{s}(t)) - \boldsymbol{D}\boldsymbol{r}(t)\|_2, \tag{32}$$

and the key steps and assumptions they use to do so in the following equations are:

1. Taylor expand $\mathcal{L}(t + \Delta t)$ around t and keep only the linear terms.

2. Assume that, after sufficiently successful training to minimize $\mathcal{L}(t)$, $p(\boldsymbol{r}(t)|\boldsymbol{s}(t)) \sim \mathcal{N}(\boldsymbol{D}^\dagger f(\boldsymbol{s}(t)), \sigma_r^2\Delta t)$.

3. Assume that $f(\boldsymbol{s}(t)) \approx \boldsymbol{D}\boldsymbol{D}^\dagger f(\boldsymbol{s}(t))$.

$$\begin{aligned}
\mathcal{L}(t + \Delta t) &= \mathbb{E}_{\boldsymbol{\eta}}\|f(\boldsymbol{s}(t + \Delta t)) - \boldsymbol{D}(\boldsymbol{r}(t) + \Delta\boldsymbol{r}(t))\|_2 && (33)\\
&\approx \mathbb{E}_{\boldsymbol{\eta}}\|f(\boldsymbol{s}(t)) + (f \circ \boldsymbol{s})'(t)\Delta t - \boldsymbol{D}(\boldsymbol{r}(t) + \Delta\boldsymbol{r}(t))\|_2 \quad \text{(Taylor expansion around } t\text{)} && (34)\\
&= \mathbb{E}_{\boldsymbol{\eta}}\|(f \circ \boldsymbol{s})'(t)\Delta t - \boldsymbol{D}\Delta\boldsymbol{r}_2(t) + f(\boldsymbol{s}(t)) - \boldsymbol{D}(\boldsymbol{r}(t) + \Delta\boldsymbol{r}_1(t) + \sigma_r\boldsymbol{\eta}(t))\|_2 && (35)\\
&\leq \mathbb{E}_{\boldsymbol{\eta}}\|(f \circ \boldsymbol{s})'(t)\Delta t - \boldsymbol{D}\Delta\boldsymbol{r}_2(t)\|_2 + \mathbb{E}_{\boldsymbol{\eta}}\|f(\boldsymbol{s}(t)) - \boldsymbol{D}(\boldsymbol{r}(t) + \Delta\boldsymbol{r}_1(t) + \sigma_r\boldsymbol{\eta}(t))\|_2 && (36)\\
&= \|(f \circ \boldsymbol{s})'(t)\Delta t - \boldsymbol{D}\Delta\boldsymbol{r}_2(t)\|_2 + \mathbb{E}_{\boldsymbol{\eta}}\|f(\boldsymbol{s}(t)) - \boldsymbol{D}(\boldsymbol{r}(t) + \Delta\boldsymbol{r}_1(t) + \sigma_r\boldsymbol{\eta}(t))\|_2 && (37)\\
&\approx \|(f \circ \boldsymbol{s})'(t)\Delta t - \boldsymbol{D}\Delta\boldsymbol{r}_2(t)\|_2 + \mathbb{E}_{\boldsymbol{\eta}}\|f(\boldsymbol{s}(t)) - \boldsymbol{D}(\boldsymbol{D}^\dagger f(\boldsymbol{s}(t)) + \Delta\boldsymbol{r}_1(t) + \sigma_r\boldsymbol{\eta}(t))\|_2 && (38)\\
&\approx \|(f \circ \boldsymbol{s})'(t)\Delta t - \boldsymbol{D}\Delta\boldsymbol{r}_2(t)\|_2 + \mathbb{E}_{\boldsymbol{\eta}}\|\boldsymbol{D}(\Delta\boldsymbol{r}_1(t) + \sigma_r\boldsymbol{\eta}(t))\|_2 && (39)\\
&\leq \|(f \circ \boldsymbol{s})'(t)\Delta t - \boldsymbol{D}\Delta\boldsymbol{r}_2(t)\|_2 + \|\boldsymbol{D}\|_F\mathbb{E}_{\boldsymbol{\eta}}\|\Delta\boldsymbol{r}_1(t) + \sigma_r\boldsymbol{\eta}(t)\|_2 && (40)\\
&\leq \|(f \circ \boldsymbol{s})'(t)\Delta t - \boldsymbol{D}\Delta\boldsymbol{r}_2(t)\|_2 + \|\boldsymbol{D}\|_F\sqrt{\mathbb{E}_{\boldsymbol{\eta}}\|\Delta\boldsymbol{r}_1(t) + \sigma_r\boldsymbol{\eta}(t)\|_2^2} \quad \text{(Jensen's ineq.)} && (41)\\
&\equiv \mathcal{L}_{signal}(\Delta\boldsymbol{r}_2(t)) + \mathcal{L}_{noise}(\Delta\boldsymbol{r}_1(t)) && (42)
\end{aligned}$$

$\mathcal{L}_{noise}$ minimizes $\mathbb{E}_F\|\Delta\boldsymbol{r}_1(t) + \sigma_r\boldsymbol{\eta}(t)\|_2$, but could also be thought of as minimizing the variation of (i.e., denoising) $\boldsymbol{r}_{noisy}(t) \equiv \boldsymbol{r}(t) + \sigma_r\boldsymbol{\eta}(t)$ by adding $\Delta\boldsymbol{r}_1(t)$. The optimal value for $\Delta\boldsymbol{r}_1(t)$ is thus:

$$\Delta\boldsymbol{r}_1^*(t) = \sigma_r^2\Delta t\frac{d}{d\boldsymbol{r}(t)}\log p(\boldsymbol{r}(t)), \tag{43}$$

as proved in Section 2 of Raphan and Simoncelli [11], letting $y \leftarrow \boldsymbol{r}_{noisy}(t)$ and $x \leftarrow \boldsymbol{r}(t)$. Note that $\Delta\boldsymbol{r}_1^*(t)$ is simply the *score function* of $\boldsymbol{r}(t)$ at time t, scaled by $\sigma_r^2\Delta t$. For multivariate Gaussian distributions, the score function has been derived in Appendix A.

The optimal RNN update equation can be obtained from Equations 4 and 7:

$$\Delta\boldsymbol{r}^*(t + \Delta t) = \left[\sigma_r^2\frac{d}{d\boldsymbol{r}(t)}\log p(\boldsymbol{r}(t)) + \boldsymbol{D}^\dagger(f \circ \boldsymbol{s})'(t)\right]\Delta t + \sigma_r\boldsymbol{\eta}(t) \tag{44}$$

C Adaptation as a Second-Order Stochastic Differential Equation

Let us first combine the following two coupled linear stochastic differential equations into one second-order equation:

$$dX_t = (AX_t + BY_t + M)dt + \sigma dB_t, \quad dY_t = (CX_t + DY_t)dt \tag{45}$$

$$d^2X_t = AdX_t + BdY_t + \sigma d(dB_t) \tag{46}$$

$$= AdX_t + BdY_t + \sigma d^2B_t \tag{47}$$

$$= AdX_t + B(CX_t + DY_t)dt + \sigma d^2B_t, \;\; Y_t = \frac{1}{Bdt}(dX_t - \sigma dB_t - AX_t dt - Mdt) \tag{48}$$

$$= AdX_t + BCX_t dt + BD\frac{1}{Bdt}(dX_t - \sigma dB_t - AX_t dt - Mdt)dt + \sigma d^2B_t \tag{49}$$

$$= AdX_t + BCX_t dt + D(dX_t - \sigma dB_t - AX_t dt - Mdt) + \sigma d^2B_t \tag{50}$$

$$= (A + D)dX_t + (BC - AD)X_t dt - DMdt - \sigma DdB_t + \sigma d^2B_t \tag{51}$$

Replacing all variables involved (except dt, σ) with matrices and vectors yields the same equation as long as $\boldsymbol{B}$ is invertible:

$$d^2\boldsymbol{x}_t = (\boldsymbol{A} + \boldsymbol{D})d\boldsymbol{x}_t + (\boldsymbol{BC} - \boldsymbol{AD})\boldsymbol{x}_t dt - \boldsymbol{Dm}dt - \sigma\boldsymbol{D}d\boldsymbol{B}_t + \sigma d^2\boldsymbol{B}_t \tag{52}$$

For consistency with the notation used throughout the paper, the equation above can be written as:

$$\boldsymbol{x}''(t) = (\boldsymbol{A} + \boldsymbol{D})\boldsymbol{x}'(t) + (\boldsymbol{BC} - \boldsymbol{AD})\boldsymbol{x}(t) - \boldsymbol{Dm} - \sigma\boldsymbol{D}\boldsymbol{\eta}(t) + \sigma\boldsymbol{\eta}'(t) \tag{53}$$

If we apply the following substitutions from Equations 14 and 17:

- $\boldsymbol{x}(t) \leftarrow \boldsymbol{r}(t)$,
- $\boldsymbol{A} \leftarrow -\sigma_r^2\Delta t\boldsymbol{\Sigma}^{-1}$,
- $\boldsymbol{B} \leftarrow -\boldsymbol{I}$,
- $\boldsymbol{m} \leftarrow \sigma_r^2\Delta t\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\mu}$,
- $\boldsymbol{C} \leftarrow -\frac{1}{\tau_a}\boldsymbol{I}$,
- $\boldsymbol{D} \leftarrow \frac{b_a}{\tau_a}\boldsymbol{I}$,

then $\boldsymbol{r}''(t)$ is:

$$\begin{aligned}\boldsymbol{r}''(t) = &\left(\frac{b_a}{\tau_a}\boldsymbol{I} - \sigma_r^2\Delta t\boldsymbol{\Sigma}^{-1}\right)\boldsymbol{r}'(t) \\ &+ \left(\frac{1}{\tau_a}\boldsymbol{I} + \frac{b_a}{\tau_a}\sigma_r^2\Delta t\boldsymbol{\Sigma}^{-1}\right)\boldsymbol{r}(t) \\ &- \frac{b_a}{\tau_a}\sigma_r^2\Delta t\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \sigma\frac{b_a}{\tau_a}\boldsymbol{\eta}(t) + \sigma\boldsymbol{\eta}'(t)\end{aligned} \tag{54}$$

Recall that, if $\boldsymbol{r}(t)$ follows a stationary Gaussian distribution, then $\frac{d}{d\boldsymbol{r}(t)}\log p(\boldsymbol{r}(t)) = \boldsymbol{\Sigma}^{-1}(-\boldsymbol{r}(t) + \boldsymbol{\mu})$ (Equation 23), and therefore $\frac{d^2}{d\boldsymbol{r}(t)^2}\log p(\boldsymbol{r}(t)) = -\boldsymbol{\Sigma}^{-1}$. Then,

$$\begin{aligned}\boldsymbol{r}''(t) = &\left(\frac{b_a}{\tau_a}\boldsymbol{I} + \sigma_r^2\Delta t\frac{d^2}{d\boldsymbol{r}(t)^2}\log p(\boldsymbol{r}(t))\right)\boldsymbol{r}'(t) \\ &- \frac{b_a}{\tau_a}\sigma_r^2\Delta t\frac{d}{d\boldsymbol{r}(t)}\log p(\boldsymbol{r}(t)) + \frac{1}{\tau_a}\boldsymbol{r}(t) \\ &- \sigma\frac{b_a}{\tau_a}\boldsymbol{\eta}(t) + \sigma\boldsymbol{\eta}'(t)\end{aligned} \tag{55}$$

D Experiments

1D Experiment Details. $s(t)$ is simulated with $\Delta t = 0.02, \sigma_s = 0.1, \sigma_0 = 0.2, \theta = 2, \mu = 5$ for $T = 100$ iterations. We used Equation 12 as $\Delta r_1^*(t)$ since a 1D Orstein-Uhlenbeck process admits a straightforward expression for $\frac{d}{dr(t)} \log p(r(t))$. The code for our 1D experiment is at: `https://colab.research.google.com/drive/1lsNagu6LhsLGsAQO7c2SGJG4YZ3hQpx7?usp=sharing`

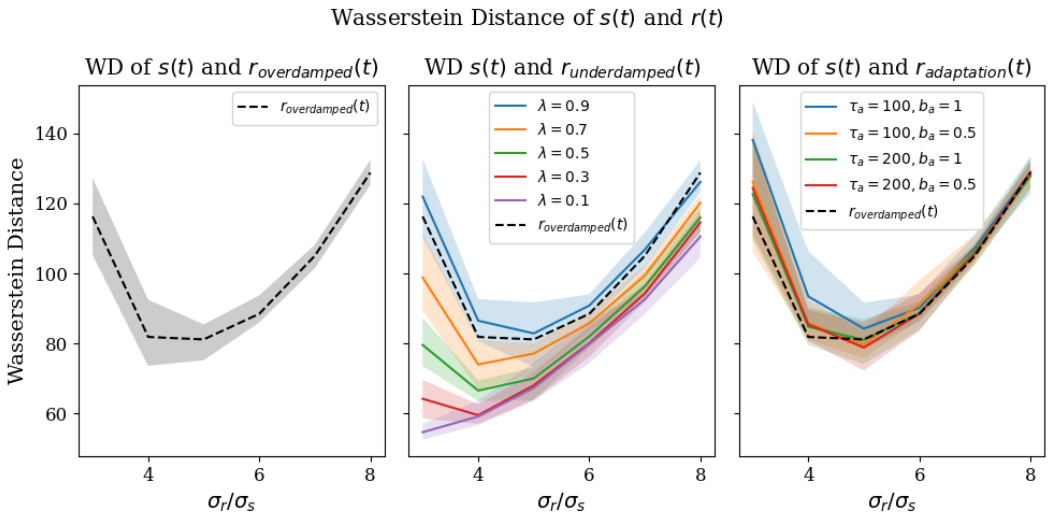

Figure 3: **Underdamped sampling improves the realism of offline replay of a 1D Ornstein-Uhlenbeck process.** Just as in Figure 2, plotted above are the mean (dashed or solid), minimum, and maximum Wasserstein distances between $s(t)$ and $r_{overdamped}(t), r_{underdamped}(t), r_{adaptation}(t)$. However, here $s(t)$ is a 1D Ornstein-Uhlenbeck process, and the score function $\frac{d}{d\boldsymbol{r}(t)} \log p(\boldsymbol{r}(t))$ is given from Equation 12 instead of being estimated from $s(t)$.

2D Experiment Details. $\boldsymbol{s}(t)$ is simulated with $\Delta t = 0.02, \sigma_s = 0.05, \sigma_0 = 0.7, \theta = 2$, and a randomly generated $\boldsymbol{\mu} \sim \mathcal{N}(0, 4\boldsymbol{I}) \in \mathbb{R}^2$ for $T = 50$ iterations. We estimated $\Delta r_1^*(t)$ using a h-neuron linear RNN parameterized by its leakage weights $\boldsymbol{w}_r \in \mathbb{R}^h$, its bias $\boldsymbol{b} \in \mathbb{R}^h$, its input weights $\boldsymbol{W}_s \in \mathbb{R}^{2,h}$, and its output weights $\boldsymbol{D} \in \mathbb{R}^{h,2}$:

$$\boldsymbol{r}(t + \Delta t) = \boldsymbol{r}(t) - \sigma(\boldsymbol{w}_r) \odot \boldsymbol{r}(t) + \boldsymbol{W}_s \boldsymbol{s}'(t) + \sigma_r \boldsymbol{\eta}(t) \tag{56}$$

where $\sigma(\cdot)$ is a sigmoid function. The RNN minimizes the reconstruction loss $\mathcal{L}(t)$ from Equation 5. The code for our 2D experiment is at: `https://colab.research.google.com/drive/1Pg41RQPjYDyuiMuUm_w-QJPQhanYHoxK?usp=sharing`

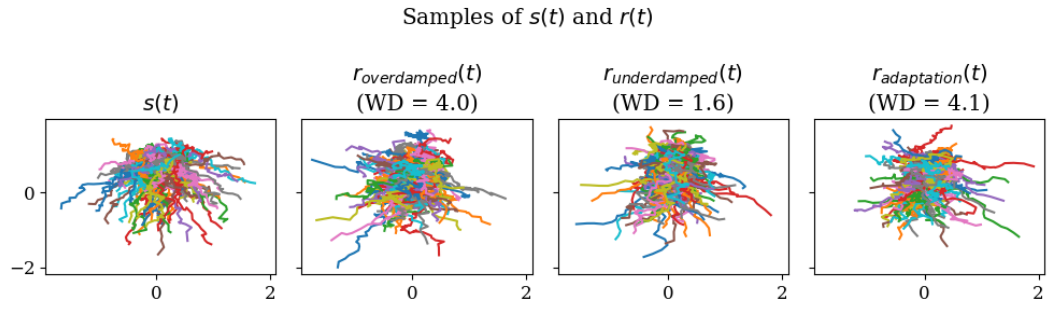

Figure 4: **Individual samples from $\boldsymbol{s}(t), \boldsymbol{r}_{overdamped}(t), \boldsymbol{r}_{underdamped}(t), \boldsymbol{r}_{adaptation}(t)$.** Plotted above are samples from Figure 2, and Wasserstein distance they incur with respect to $p(\boldsymbol{s}(t))$.