MITIGATING DIFFUSION MODEL MEMORIZATION WITH HIGHER ORDER LANGEVIN DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion/score-based models have emerged as powerful generative models capable of generating high-quality samples that mimic the distribution of training data. However, these models have been observed to reproduce training samples, a phenomenon known as "memorization," potentially violating copyright and privacy. In this paper, we study the effect of Higher-Order Langevin Dynamics (HOLD) on this phenomenon. HOLD diffusion processes introduce auxiliary variables; if the data variable is interpreted as "position," then the auxiliary variables can be interpreted as "velocity" and "acceleration," depending on the chosen order of the model. They were originally proposed based on the intuition that these auxiliary variables regularize the trajectories of the data variable by implicitly imposing additional dynamical constraints. Our work provides, to our knowledge, the first theoretical characterization of this time-domain regularization effect of HOLD. Specifically, we show that in HOLD, the dynamics of the data variable are governed by a low-pass-filtered version of the learned score function, with smoothness increasing with the order of HOLD. We then analyze the optimal empirical score of HOLD and using spectral analysis reveal that, at low noise levels, it is hard to learn. Together, our results explain how increasing the model order mitigates memorization. Finally, we present an empirical study on real-world data that supports our theory and highlights this distinct advantage of HOLD over standard diffusion models in practice.

1 INTRODUCTION

First introduced in (Sohl-Dickstein et al., 2015) and later refined in (Song and Ermon, 2019; Ho et al., 2020; Song et al., 2021), Denoising Diffusion Models (DDMs) have demonstrated remarkable success in generating high quality samples from intractable high-dimensional data distributions. They are based on generating samples from a tractable latent distribution (typically, standard normal) and converting them to samples from the target distribution via iterative denoising with gradually decreasing noise level. Trained DDMs have also emerged as powerful generative priors for editing and restoring signals and images (Meng et al., 2022; Kawar et al., 2022; Garber and Tirer, 2024). However, recently, it has been observed that DDMs are prone to exactly reproducing training samples (Carlini et al., 2023; Somepalli et al., 2023). This phenomenon, termed "memorization", can have severe negative implications, such as violating copyright and privacy.

Memorization of DDMs has recently become a topic of interest both experimentally and theoretically. (Yoon et al., 2023) suggested that diffusion models either "generalize" or "memorize" in a mutually exclusive manner, and that overfitting in this context is not "benign" (unlike certain phenomena in linear regression (Bartlett et al., 2020)). In fact, a key factor for diffusion memorization is that, if one considers an unconstrained denoiser/score function, the optimum of the empirical mean squared error loss used in training has an analytical form (Karras et al., 2022), which necessarily outputs a sample from the training set as the noise level decreases (Yi et al., 2023; Zhang et al., 2024) (more details in Section 2). The work of (Kadkhodaie et al., 2024) argues that diffusion models generalize due to inductive biases; their network architecture encourages shrinkage towards geometry-adaptive harmonic bases that are informed by the data distribution. (Zhang et al., 2024) performs similar experiments, demonstrating a separation between the memorization and generalization regimes depending on the size of the data set. They argue that in the generalization regime, the learned distributions are largely independent of model architectures and are instead most heav-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ily influenced by the training data. (Gu et al., 2025) empirically studies the relationship between the size of the data set, the memorization ratio, and different state-of-the-art architectures. Other works on diffusion memorization (Bonnaire et al., 2025; George et al., 2026) include theoretical analyses based on the random features model. In particular, (Bonnaire et al., 2025) empirically and theoretically studies the level of memorization along the optimization. Their results suggest that memorization can be avoided or mitigated using early stopping within a time range that depends on the size of the data set. However, in practice, especially in the presence of a biased validation dataset, it is hard to gauge when to stop. This study motivates the question of whether there are other ways to reduce memorization in addition to early stopping.

In parallel to the work analyzing memorization in diffusion models, separate methods have been developed with the goals of shortening and smoothing the diffusion model's path from target and latent distributions through the introduction of higher-order auxiliary variables. Intuitively, if the data variable is interpreted as "position," then the auxiliary variables can be interpreted as "velocity" and "acceleration," depending on the chosen order of the model. (Dockhorn et al., 2022) proposed critically-damped Langevin dynamics (CLD), which introduces a velocity variable, and studies the effect of critical damping on the diffusion dynamics. In this context, critical-damping refers to choosing parameters so that the diffusion forward process's matrix has a single geometric eigenvalue and thus increases the speed of convergence along diffusion time. Inspired by this work, higher-order Langevin dynamics (HOLD) (Shi and Liu, 2024) was introduced to simplify the diffusion process of CLD as an Ornstein–Uhlenbeck process superimposed with a skew symmetric variable coupling. Follow ups of this work have shown that it is possible to critically damp HOLD (Sterling and Bugallo, 2025; Sterling et al., 2025).

In this paper, we study the effect of higher-order diffusion modeling on memorization. First, we close a gap in the literature on the regularization effect of these models. These models were originally proposed on the basis of the intuition that they regularize the trajectories of the data variable by implicitly imposing additional dynamical constraints; this regularization effect lacks theoretical justification. We provide, to our knowledge, the first rigorous characterization of such time-domain regularization effect of HOLD. Specifically, we utilize the Laplace transform to demonstrate the temporal low-pass filtering effect of the model order on the learned score function. Then, we analyze the optimal empirical score function of HOLD models and present a result on memorization delay based on a frequency analysis of the target function at low noise levels. Together, these findings explain the mitigation of memorization as the model order increases. Finally, we present an empirical study on real-world data that supports our theory. That is, we measure the quality of generation and memorization rate of models of different orders along training for three different datasets: the CelebA dataset (extending the setup of (Bonnaire et al., 2025)), the Animal Faces dataset (Choi et al., 2020), and the CIFAR-10 dataset (Krizhevsky et al., 2009). The experiments show that HOLD models offer similar Fréchet Inception Distance (FID) results as plain (first-order) diffusion models, while memorization, measured as in previous work, is delayed and attenuated as the model order increases, aligned with our theory.

2 Background

2.1 Score-based generative modeling and memorization by the optimal empirical score

Previous works (Karras et al., 2022; Yi et al., 2023) have derived that regular score matching, which uses an empirical training objective, has a closed form optimal score function that is based on a weighted average of the training data and even converges to being dependent solely on the single training sample closest to its input as $t \to 0$.

Let us present this formally for the Ornstein–Uhlenbeck (OU) process, a standard DDM framework which possesses the following stochastic differential equation (SDE):

$$d\mathbf{x}_t = -\xi \mathbf{x}_t dt + \sqrt{2\xi L^{-1}} d\mathbf{w}_t,$$

where L^{-1} and ξ are algorithmic parameters and $\mathbf{w}_t$ represents the standard Wiener process. In this forward process, given a data sample $\mathbf{x}_0$, a sample $\mathbf{x}_t$ can be drawn according to $\mathbf{x}_t = \exp(-\xi t)\mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$, where $\sigma_t^2 = L^{-1}(1 - \exp(-2\xi t))$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

The generation of data samples by DDMs is based on the fact that there is an associated backward SDE with the same marginal distribution of $\mathbf{x}_t$ (Anderson, 1982; Song et al., 2021):

$$d\mathbf{x}_t = -\xi \left(\mathbf{x}_t + 2L^{-1}\mathbf{s}(\mathbf{x}_t, t)\right) dt + \sqrt{2\xi L^{-1}} d\bar{\mathbf{w}}_t, \tag{1}$$

where $\mathbf{s}(\cdot, t) = \nabla_{\mathbf{x}} \log p_t(\cdot)$ is the score function of $\mathbf{x}_t$ (with p_t denoting the distribution of $\mathbf{x}_t$). The score function is modeled by $\mathbf{s}_\theta(\cdot, t)$ and learned during training. For the standard training objective,

$$\mathcal{L} = \mathbb{E}_{t\sim\mathcal{U}(0,1),\mathbf{x}_0,\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})} ||\boldsymbol{\epsilon} + \sigma_t \mathbf{s}_\theta(\mathbf{x}_t, t)||^2, \tag{2}$$

with the empirical distribution $p(\mathbf{x}_0) = \frac{1}{n_{\text{train}}} \sum_{k=1}^{n_{\text{train}}} \delta(\mathbf{x}_0 - \mathbf{x}_0^{(k)})$, the optimal (unconstrained) score function is given by $\mathbf{s}_{\text{emp}}^{\text{OU}}(\mathbf{x}, t) = \nabla_{\mathbf{x}} \log p_t^{\text{emp, OU}}(\mathbf{x})$, where

$$p_t^{\text{emp, OU}}(\mathbf{x}) = \frac{1}{n_{\text{train}}} \sum_{k=1}^{n_{\text{train}}} (2\pi\sigma_t^2)^{-h/2} \exp\left(-\frac{1}{2\sigma_t^2} \left\| \mathbf{x} - \exp(-\xi t)\mathbf{x}_0^{(k)} \right\|^2\right).$$

That is,

$$s_{\text{emp}}^{\text{OU}}(\mathbf{x}, t) = -\frac{1}{\sigma_t^2} \left(\mathbf{x} - \exp(-\xi t) \frac{\sum_{k=1}^{n_{\text{train}}} \mathcal{N}(\mathbf{x} \mid \exp(-\xi t)\mathbf{x}_0^{(k)}, \sigma_t^2 \mathbf{I})\ \mathbf{x}_0^{(k)}}{\sum_{k=1}^{n_{\text{train}}} \mathcal{N}(\mathbf{x} \mid \exp(-\xi t)\mathbf{x}_0^{(k)}, \sigma_t^2 \mathbf{I})}\right).$$

which follows the description above (i.e., weighted average of the training samples that converges to the nearest training sample as $t \to 0$).

This conveys that diffusion models generalize due to explicit and implicit inductive biases, while approaching the unconstrained empirical optimum, e.g., due to a small number of training samples (Kadkhodaie et al., 2024; Zhang et al., 2024) and/or prolonged training (Bonnaire et al., 2025; George et al., 2026), increases memorization.

2.2 Overview of HOLD

We start this section by providing an overview of HOLD (Shi and Liu, 2024; Sterling et al., 2025) following the variable conventions used in (Dockhorn et al., 2022). Suppose that a data point is expressed as a vector $\mathbf{x}_0 \in \mathbb{R}^h$. Let $\alpha, L^{-1}, \{\gamma_k\}_{1\le k\le n-1}$, and ξ denote algorithmic parameters, where n denotes the order of the model. In HOLD, we define the initial diffusion variable $\mathbf{u}_0 = \text{vec}(\mathbf{x}_0, \mathbf{v}_0^{(1)}, \mathbf{v}_0^{(2)} \dots \mathbf{v}_0^{(n-1)})$, where $\mathbf{v}_0^{(1)}, \mathbf{v}_0^{(2)} \dots \mathbf{v}_0^{(n-1)} \sim_{\text{iid}} \mathcal{N}(\mathbf{0}, \alpha L^{-1}\mathbf{I})$ are auxiliary variables. Take the following system:

$$\mathbf{F} = \sum_{i=1}^{n-1} \gamma_i \left(\mathbf{E}_{i,i+1} - \mathbf{E}_{i+1,i}\right) - \xi \mathbf{E}_{n,n}, \quad \mathbf{G} = \sqrt{2\xi L^{-1}} \mathbf{E}_{n,n},$$

where $\mathbf{E}_{i,j} \in \mathbb{R}^{n\times n}$ is the matrix of all zeros with a one at index pair (i, j). The forward process evolves according to the SDE:

$$d\mathbf{u}_t = \mathcal{F}\mathbf{u}_t dt + \mathcal{G} d\mathbf{w}_t, \tag{3}$$

where $\mathcal{F} = \mathbf{F} \otimes \mathbf{I}_h$, $\mathcal{G} = \mathbf{G} \otimes \mathbf{I}_h$ (with $\otimes$ denoting the Kronecker product), and $\mathbf{w}_t$ represents the standard Wiener process. For example, when $n = 3$, the dynamics are governed by:

$$\begin{cases} d\mathbf{x}_t &= \gamma_1 \mathbf{v}_t^{(1)} dt, \\ d\mathbf{v}_t^{(1)} &= \left(-\gamma_1 \mathbf{x}_t + \gamma_2 \mathbf{v}_t^{(2)}\right) dt, \\ d\mathbf{v}_t^{(2)} &= \left(-\gamma_2 \mathbf{v}_t^{(1)} - \xi \mathbf{v}_t^{(2)}\right) dt + \sqrt{2\xi L^{-1}} d\mathbf{w}_t. \end{cases}$$

Here, the dynamics of $\mathbf{x}_t$ are modeled by the variable "velocity" $\mathbf{v}_t^{(1)}$ and the variable "acceleration" $\mathbf{v}_t^{(2)}$. In the notation of equation 3, the $\mathbf{F}$ and $\mathbf{G}$ matrices for $n = 3$ become:

$$\mathbf{F} = \begin{pmatrix} 0 & \gamma_1 & 0 \\ -\gamma_1 & 0 & \gamma_2 \\ 0 & -\gamma_2 & -\xi \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sqrt{2\xi L^{-1}} \end{pmatrix}.$$

It can be shown that $\mathbf{u}_t$ that satisfies equation 3 must be normally distributed (conditioned on $\mathbf{x}_0$), and expressions for the mean and covariance of $\mathbf{u}_t$ must satisfy the following differential equations, which can be derived from the Fokker-Planck equations (Särkkä and Solin, 2019),

$$\frac{d\boldsymbol{\mu}_t}{dt} = \mathcal{F}\boldsymbol{\mu}_t, \quad \frac{d\boldsymbol{\Sigma}_t}{dt} = \mathcal{F}\boldsymbol{\Sigma}_t + (\mathcal{F}\boldsymbol{\Sigma}_t)^T + \mathcal{G}\mathcal{G}^T.$$

In (Sterling et al., 2025) it is proven that $\mathbf{u}_t$ must also have the following distribution:

$$\mathbf{u}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t), \quad \boldsymbol{\mu}_t = \exp(\mathcal{F}t)\mathbf{u}_0,$$
$$\boldsymbol{\Sigma}_t = L^{-1}\mathbf{I} + \exp(\mathcal{F}t)\left(\boldsymbol{\Sigma}_0 - L^{-1}\mathbf{I}\right)\exp(\mathcal{F}t)^T.$$

In the forward diffusion process, samples are generated according to $\mathbf{u}_t = \exp(\mathcal{F}t)\mathbf{u}_0 + \mathbf{L}_t\boldsymbol{\epsilon}_{\text{full}}$, where $\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2 \ldots \boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_h)$, $\boldsymbol{\epsilon}_{\text{full}} = \text{vec}(\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2 \ldots \boldsymbol{\epsilon}_n)$, and $\mathbf{L}_t = \text{cholesky}(\boldsymbol{\Sigma}_t)$. The computation of the term $\exp(\mathcal{F}t)$ is not obvious and (Shi and Liu, 2024) uses Putzer's Lemma (Putzer, 1966) to compute it. However, when the parameters of $\mathcal{F}$ are chosen so that it possesses a single geometric eigenvalue $s_* = -\sqrt{2n-3}$, as in (Sterling et al., 2025), then it may be calculated by a finite Taylor series, with the understanding that $\mathcal{F} - s_*\mathbf{I}$ is nilpotent:

$$\exp(\mathcal{F}t) = \exp(s_*t)\sum_{k=0}^{n-1}\frac{(\mathcal{F} - s_*\mathbf{I})^k t^k}{k!}. \tag{4}$$

The loss function in this forward SDE is the same score matching objective that (Dockhorn et al., 2022; Shi and Liu, 2024) uses. Instead of modeling the score of the data variable $\mathbf{x}_t$ or the score of the entire $\mathbf{u}_t$, it suffices to model the score of the last auxiliary variable $\mathbf{v}_t^{(n-1)}$ by $\mathbf{s}_\theta(\mathbf{u}_t, t)$, and learn it with the objective

$$\mathcal{L} = \mathbb{E}_{t\sim\mathcal{U}(0,1),\mathbf{u}_0,\boldsymbol{\epsilon}_{\text{full}}}\|\boldsymbol{\epsilon}_n + \mathbf{s}_\theta(\mathbf{u}_t, t)\left(\mathbf{L}_t[nh, nh]\right)\|^2. \tag{5}$$

The following ordinary differential equation (ODE) (Song et al., 2021) shares the same approximate marginal probability densities as the backward SDE and may be used for sample generation:

$$d\mathbf{u}_t = \left(\mathcal{F}\mathbf{u}_t - \frac{1}{2}\mathcal{G}\mathcal{G}^T\mathbf{S}_\theta(\mathbf{u}_t, t)\right)dt, \tag{6}$$

where $\mathbf{S}_\theta(\mathbf{u}_t, t) = \text{vec}(\mathbf{0}_{(n-1)h}, \mathbf{s}_\theta(\mathbf{u}_t, t))$. In (Sterling et al., 2025) it is further proven that the system is critically damped for $n \geq 2$, meaning $\mathbf{F}$ has a single geometric eigenvalue, if and only if

$$\gamma_{n-i} = \sqrt{2n-3}\sqrt{\frac{n^2 - i^2}{4i^2 - 1}}, \quad \xi = n\sqrt{2n-3},$$

assuming a scaling choice of $\gamma_1 = 1$. For further convenience in this paper, define $\bar{\gamma} = \prod_{i=1}^{n-1}\gamma_i$.

3 Theoretical Analysis

This section presents our theoretical analysis of the regularization introduced by the HOLD model and its ability to mitigate memorization.

3.1 In HOLD the score is being low-pass filtered

This subsection argues that HOLD introduces a low-pass filtering effect on the score during the generation procedure. Although neither the reverse SDE nor the reverse ODE have closed form solutions, we study the reverse ODE given by equation 6 and analyze the relationship between the score function and generated samples from a signal and systems perspective (Oppenheim et al., 1997). We start with Lemma 1 below, which helps in the proof of our key Theorem 1. Essentially, it is based on taking the Laplace transform of both sides of equation 6 and identifying recursive patterns.

Lemma 1. *Consider the ODE equation 6. If the Laplace transform of $\mathbf{x}_t$ is $\tilde{\mathbf{x}}(s)$, the Laplace transform of $\mathbf{s}_\theta(\mathbf{u}_t, t)$ is $\tilde{\mathbf{s}}(s)$ (which represents t from both arguments) and $P(s)$ is the characteristic polynomial of $\mathbf{F}$, then*

$$\tilde{\mathbf{x}}(s) = \tilde{\mathbf{x}}^{\text{natural}}(s) - \frac{\bar{\gamma}\xi L^{-1}\tilde{\mathbf{s}}(s)}{P(s)},$$

where $\tilde{\mathbf{x}}^{\text{natural}}(s)$ arises from nonzero $\mathbf{u}_0$ and is independent of the score. Its full expression is available in the appendix.

See Appendix A.2 for the proof. Now that we have this result, we may proceed to the following theorem.

Theorem 1. *Let $h_t^{(n)} = \frac{-\bar{\gamma} n \sqrt{2n-3} L^{-1}}{(n-1)!} t^{n-1} \exp(-t\sqrt{2n-3})$ for $n \geq 2$. For model order $n \geq 2$, the solution to equation 6 for the data variable $\mathbf{x}_t$ is as follows:*

$$\mathbf{x}_t = \mathbf{x}_t^{\text{natural}} + h_t^{(n)} * \mathbf{s}_\theta(\mathbf{u}_t, t),$$

where '$*$' denotes the convolution operation (in the time domain).

See Appendix A.3 for the proof. The proof uses the Residue Theorem applied to the equation in Lemma 1. The variables $\tilde{\mathbf{x}}^{\text{natural}}(s)$ and $\mathbf{x}_t^{\text{natural}}$ are referred to as "natural" parameters because they both come from nonzero $\mathbf{u}_0$, as opposed to the "forcing" term, $-\frac{1}{2}\mathcal{G}\mathcal{G}^T\mathbf{S}_\theta(\mathbf{u}_t, t)$ in equation 6. It should also be noted that the score $\mathbf{s}_\theta(\mathbf{u}_t, t)$ affects $\mathbf{x}_t$ only through convolution with $h_t^{(n)}$. Theorem 1 is a novel result that is most useful for theoretical analysis and may also benefit the design of the forward process itself. It does not directly lead to an algorithm to solve $\mathbf{x}_t$ because $\mathbf{s}_\theta$ is a function of $\mathbf{u}_t$, which is itself a function of $\mathbf{x}_t$, as well as the fact that $\mathbf{x}_t^{\text{natural}}$ depends on $\mathbf{x}_0$ (which is unknown during the backward process). Both results speak to the optimality of choosing critically damped parameters. In (Sterling et al., 2025) it was proven that the critically damped parameters are optimal for the following design objective involving the forward matrix $\mathbf{F}$:

$$\min_{\gamma_2, \gamma_3 \cdots \gamma_{n-1}, \xi} \max\left(\text{Re}\left(\text{eig}(\mathbf{F})\right)\right). \tag{7}$$

With Lemma 1 and Theorem 1, one may understand that not choosing critically damped parameters would result in the characteristic polynomial $P(s) = \prod_{i=1}^{n}(s - s_i)$, which leads to $h_t^{(n)} = \sum_{i=1}^{n} c_i \exp(s_i t)$ (the c_i are the coefficients obtained from the residue theorem). Because the critically damped parameters minimize equation 7, not using the critically damped parameter choices requires the existence of at least one eigenvalue of $\mathbf{F}$, s_k such that $s_k > s_*$. The consequence of such a mode $c_k \exp(s_k t)$ is a slower convergence along diffusion time, or from a frequency perspective: a mode that allows more high frequencies from $\mathbf{s}_\theta$ to pass into $\mathbf{x}_t$. The critically damped parameters therefore represent the filter with the sharpest frequency cutoff.

A simple derivation in Appendix A.5 yields the result that the solution of the backward ODE associated with the (first-order) OU process: $d\mathbf{x}_t = -\xi\mathbf{x}_t dt + \sqrt{2\xi L^{-1}} d\mathbf{w}_t$ may be expressed with $h_t^{\text{OU}} = -\xi L^{-1} \exp(-\xi t)$:

$$\mathbf{x}_t = \exp(-\xi t)\mathbf{x}_0 - \xi L^{-1} \int_0^t \exp(-\xi(t-\tau))\mathbf{s}_\theta(\mathbf{x}_\tau, \tau) d\tau = \exp(-\xi t)\mathbf{x}_0 + h_t^{\text{OU}} * \mathbf{s}_\theta(\mathbf{x}_t, t).$$

Contrasting this result with Theorem 1 sheds new light on the inherent regularization of HOLD. It formally shows the smoothing effect of HOLD on the generated sample trajectories, where "smoothing" is manifested by $h_t^{(n)}$ acting on the score function as a low-pass filter with stronger attenuation of high frequencies than h_t^{OU}. This function, which constrains the generation trajectories, along with the theory developed in the following section, explains why HOLD reduces memorization. We turn to visually demonstrate our findings on the filters that are convolved with the score function. Figure 1 plots the Fourier transforms of h_t^{OU} and $h_t^{(n)}$ for $n = 2, 3, 4$, which may be obtained from each Laplace transform by inserting $s = i\omega$. An ideal low-pass filter takes the form of an indicator function: $H_{\text{ideal}}(\omega) = \begin{cases} 1, & |\omega| \leq \omega_{\text{cutoff}} \\ 0, & \text{else} \end{cases}$; this is achieved by selecting $h_t^{(n)}$ as a scaled $\text{sinc}(t) = \frac{\sin(2\pi t)}{2\pi t}$. However, this cannot be perfectly implemented as a causal $h_t^{(n)}$, resulting in a complex valued Fourier transform, and it is hard to model such a function with a set of differential equations as is being done here. Therefore, the better low-pass filters in Figure 1 allow a wider range of low frequencies to pass while more clearly attenuating higher frequencies. This sharper attenuation is achieved by the term t^{n-1} in $h_t^{(n)}$, which is associated with a pole in the Laplace space of multiplicity n. Clearly, the HOLD filters are stronger low-pass filters than the OU filter.

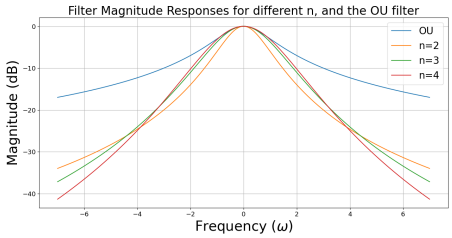


Figure 1: Magnitudes of the Fourier Transforms $|H(i\omega)|$ for different HOLD diffusion model orders n, and the Ornstein–Uhlenbeck filter with $\xi = 1$. One may observe that the HOLD filters are better at attenuating higher frequencies than OU, while still allowing a wide band of lower frequencies.

3.2 HOLD MITIGATES MEMORIZATION

We have shown above that HOLD regularizes the generation process. In this subsection, we provide reasoning for its reduced memorization (shown empirically to be attenuated and delayed) by adding to our regularized dynamics result the fact that the HOLD optimal empirical score function is harder to learn than the optimal empirical score of a standard (first-order) DDM. We do so by comparing the Non-Uniform Fourier Transform (NUFT), i.e., Fourier transform that is weighted by the density, of the optimal empirical score function between HOLD and the OU process and then showing that the HOLD spectrum is wider than the OU spectrum in the low noise regime. The "F-Principle"/"spectral bias" theory (Rahaman et al., 2019; Basri et al., 2019; 2020; Xu et al., 2020) then predicts that the HOLD higher-frequency components are learned more slowly, which delays memorization.

3.2.1 THE HOLD OPTIMAL EMPIRICAL SCORE FUNCTION IS HARD TO LEARN

Here, we present the HOLD optimal empirical score function. That is, we derive the optimal unconstrained score function that minimizes the HOLD loss objective in equation 5 under the empirical data distribution.

Proposition 1. *Consider the empirical distribution* $p_{\text{data}}(\mathbf{x}_0) = \frac{1}{n_{\text{train}}} \sum_{k=1}^{n_{\text{train}}} \delta(\mathbf{x}_0 - \mathbf{x}_0^{(k)})$. *Assume that each training data point is assigned* B *realizations of auxiliary variables once at the algorithm's initialization. Let* $\mathbf{u}_0^{(k,l)} = \text{vec}(\mathbf{x}_0^{(k)}, \mathbf{v}_0^{(1),(k,l)}, \ldots, \mathbf{v}_0^{(n-1),(k,l)})$ *denote this assignment. The empirical distribution of* $\mathbf{u}_t$, *denoted by* $p_t^{\text{emp,HOLD}}$, *obeys*

$$p_t^{\text{emp,HOLD}}(\mathbf{u}) = \frac{1}{Bn_{\text{train}}} \sum_{k=1}^{n_{\text{train}}} \sum_{l=1}^{B} (2\pi \det \boldsymbol{\Sigma}_t)^{-h/2} \exp\left(-\frac{1}{2} \left|\left| \boldsymbol{\Sigma}_t^{-1/2} \left(\mathbf{u} - \exp(\mathcal{F}t)\mathbf{u}_0^{(k,l)} \right) \right|\right|^2 \right)$$

$$\nabla_{\mathbf{u}} \log p_t^{\text{emp,HOLD}}(\mathbf{u}) = -\boldsymbol{\Sigma}_t^{-1}\mathbf{u} + \boldsymbol{\Sigma}_t^{-1} \exp(\mathcal{F}t) \frac{\sum_{k=1}^{n_{\text{train}}} \sum_{l=1}^{B} \mathcal{N}(\mathbf{u} \mid \exp(\mathcal{F}t)\mathbf{u}_0^{(k,l)}, \boldsymbol{\Sigma}_t)\ \mathbf{u}_0^{(k,l)}}{\sum_{k=1}^{n_{\text{train}}} \sum_{l=1}^{B} \mathcal{N}(\mathbf{u} \mid \exp(\mathcal{F}t)\mathbf{u}_0^{(k,l)}, \boldsymbol{\Sigma}_t)},$$

The unconstrained score that minimizes equation 5 is given by $\mathbf{s}_{\text{emp}}^{\text{HOLD}}(\mathbf{u}_t, t) = \left[\nabla_{\mathbf{u}_t} \log p_t^{\text{emp,HOLD}}(\mathbf{u}_t) \right]_{n(h-1):nh}$.

See Appendix A.1 for the proof. It proceeds in a very similar manner to the proof for the standard diffusion model, but in a multivariate non-isotropic setting. This proposition sheds some light on the hypothesized regularization of the training procedure. Specifically, note that the score of the empirical distribution is a function of each $\mathbf{u}_0^{(k,l)}$, which contains initial auxiliary variables $(\mathbf{v}_0^{(1),(k,l)}, \ldots, \mathbf{v}_0^{(n-1),(k,l)})$ that are completely independent of the training data $\mathbf{x}_0^{(k)}$. Memorization under HOLD thereby requires the network to fit a more complex, higher-dimensional target function than in a standard OU process. In practice, the auxiliary variables are drawn per optimization iteration (interpreted as $B \to \infty$), which further hardens the memorization of the training data points. This intuition is studied in more detail below.

The "F-Principle" delays HOLD Memorization. Here, we formally show another distinction between HOLD and the standard first-order diffusion process. Specifically, we demonstrate that for

small t, the Non-Uniform Fourier Transform (NUFT) spectrum of the empirical score function is wider for HOLD compared to the OU process.

Note that the classical Fourier Transform of a score function is undefined, even for Gaussian data. (The score in this case would be proportional to $-\mathbf{x}$, and $\int_{\mathbb{R}^h} |\mathbf{x}| d\mathbf{x}$ diverges.) Instead, the NUFT of a score $\mathbf{s}(\mathbf{x})$ associated with some distribution $p(\mathbf{x})$ that vanishes at $\pm\infty$ is given by (and simplifies with integration by parts):

$$\mathcal{F}\{\mathbf{s}\}(\mathbf{k}) := \mathbb{E}_{\mathbf{x}}(\mathbf{s}(\mathbf{x}) \exp(i\mathbf{k}^T\mathbf{x})) = \int \nabla_{\mathbf{x}} p(\mathbf{x}) \exp(i\mathbf{k}^T\mathbf{x}) d\mathbf{x} = -i\mathbf{k}\mathbb{E}_{\mathbf{x}}(\exp(i\mathbf{k}^T\mathbf{x})). \quad (8)$$

This connects to the characteristic function $\Phi_{\mathbf{x}}(\mathbf{k}) = \mathbb{E}(\exp(i\mathbf{k}^T\mathbf{x}))$. Next, we have the following observation on the NUFT of the HOLD's score computed at $\mathbf{k}_{\text{full}} = \text{vec}(\mathbf{k}, \mathbf{0} \dots \mathbf{0})$:

$$\left[\mathbb{E}_{\mathbf{u}\sim p_t^{\text{emp,HOLD}}}(\nabla_{\mathbf{u}} \log p_t^{\text{emp,HOLD}}(\mathbf{u}) \exp(i\mathbf{k}_{\text{full}}^T\mathbf{u}))\right]_{1:h} = \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x},t}^{\text{emp,HOLD}}}(\mathbf{s}_{\text{emp}}^{\text{HOLD}}(\mathbf{x}, t) \exp(i\mathbf{k}^T\mathbf{x}))$$

where $p_{\mathbf{x},t}^{\text{emp,HOLD}} = \int p_t^{\text{emp,HOLD}}(\mathbf{u}_t) d\mathbf{v}_t^{(1)} \dots d\mathbf{v}_t^{(n-1)}$ is HOLD's marginal data distribution, and $\mathbf{s}_{\text{emp}}^{\text{HOLD}}(\mathbf{x}, t)$ denotes its score. As implied by equation 8, the entries $h+1 : nh$ are zeros for this choice of $\mathbf{k}_{\text{full}}$.

Proposition 2. *The norm of the NUFT of the HOLD empirical score function, $\mathbf{s}_{\text{emp}}^{\text{HOLD}}(\mathbf{x}, t)$, of the marginal $p_{\mathbf{x},t}^{\text{emp,HOLD}}$ is given by:*

$$\left|\left|\mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x},t}^{\text{emp,HOLD}}}(\mathbf{s}_{\text{emp}}^{\text{HOLD}}(\mathbf{x}, t) \exp(i\mathbf{k}^T\mathbf{x}))\right|\right|$$
$$= \frac{||\mathbf{k}||}{n_{\text{train}}} \exp\left(-\frac{1}{2}\mathbf{k}^T\boldsymbol{\Sigma}_t^{(0,0)}\mathbf{k}\right) \sqrt{\sum_{j=1}^{n_{\text{train}}} \sum_{k=1}^{n_{\text{train}}} \cos\left((\boldsymbol{\mu}_t^{(0),(k)} - \boldsymbol{\mu}_t^{(0),(j)})^T\mathbf{k}\right)}$$

where $\boldsymbol{\mu}_t^{(0),(k)} = \left[\exp(\mathcal{F}t)\mathbf{u}_0^{(k)}\right]_{0:h}$, $\boldsymbol{\Sigma}_t^{(0,0)} = \left[\mathbf{I} - \exp(\mathcal{F}t)\exp(\mathcal{F}t)^T\right]_{(0:h,0:h)}$. *Furthermore, as $t \to 0^+$, $\boldsymbol{\Sigma}_t^{(0,0)} = \mathcal{O}(t^3)$. Contrarily, the norm of the Non-Uniform Fourier Transform of the OU empirical score function is given by:*

$$\left|\left|\mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x},t}^{\text{emp,OU}}}(\mathbf{s}_{\text{emp}}^{\text{OU}}(\mathbf{x}, t) \exp(i\mathbf{k}^T\mathbf{x}))\right|\right|$$
$$= \frac{||\mathbf{k}||}{n_{\text{train}}} \exp\left(-\frac{\sigma_t^2}{2}||\mathbf{k}||^2\right) \sqrt{\sum_{j=1}^{n_{\text{train}}} \sum_{k=1}^{n_{\text{train}}} \cos\left(\exp(-\xi t)(\mathbf{x}_0^{(k)} - \mathbf{x}_0^{(j)})^T\mathbf{k}\right)}$$

where $\sigma_t^2 = 1 - \exp(-2\xi t)$, and as $t \to 0^+$, $\sigma_t^2 = \mathcal{O}(t)$.

See Appendix A.4 for the proof. The significance of Proposition 2 comes from the "F-Principle": deep neural networks fit target functions from low to high frequencies during the training process. As $t \to 0^+$ HOLD's $\boldsymbol{\Sigma}_t^{(0,0)} = \mathcal{O}(t^3)$ and the OU process's $\sigma_t^2 = \mathcal{O}(t)$. This implies that $\exp\left(-\frac{1}{2}\mathbf{k}^T\boldsymbol{\Sigma}_t^{(0,0)}\mathbf{k}\right)$ attenuates the HOLD frequency response less than $\exp\left(-\frac{\sigma_t^2}{2}||\mathbf{k}||^2\right)$ does for the frequency response of the OU process for small t. See Appendix C.1 for a visualization of the 1D GMM data example (Figure 8). Note that while Proposition 2 seems to contradict Theorem 1, it actually takes the NUFT with respect to spatial frequency $\mathbf{k}$, and not temporal frequency. Therefore, Proposition 2 predicts that HOLD even further delays memorization because its empirical score function possesses a higher frequency target function.

4 Experiments

In order to validate the theory that HOLD regularizes training and sample generation processes, we perform experiments in a setup similar to (Bonnaire et al., 2025). We compare the memorization rate and generation quality along the training of HOLD diffusion models and first-order diffusion models based on the widely used Variance Preserving Stochastic Differential Equation (VPSDE) framework. Successful regularization should force memorization lower while preserving similar image qualities,

measured by Fréchet Inception Distance (FID) (Heusel et al., 2017). Memorization measurement is performed at the sample level, as in (Bonnaire et al., 2025). The distances between every generated sample and each training sample are calculated. For every generated sample, the gap ratio, the ratio of distances between the first and second closest training samples, is computed. If it falls below a certain threshold, then that generated sample is declared to be memorized. Each percentage of memorization reported is the ratio of memorized generated samples to the total number of generated samples in that batch. Formally, we use a batch size $B = 1024$ and a gap ratio threshold $\tau = 0.333$ (as in (Bonnaire et al., 2025; Yoon et al., 2023; Gu et al., 2025)), and define $d_k^{(j)}$ as the ℓ_2 distance between the kth generated sample and the jth closest image in the training set. The memorization metric, termed Fmem, is calculated as follows:

$$M_k = \begin{cases} 1, & d_k^{(1)}/d_k^{(2)} < \tau \\ 0, & \text{else} \end{cases}, \quad \text{Fmem} = \frac{1}{B}\sum_{k=1}^{B} M_k.$$

Confidence intervals are obtained recognizing that Fmem is a sample proportion. To ensure that this methodology works on more realistic color datasets, additional experiments are performed on the Animal Faces dataset (Choi et al., 2020) and the CIFAR-10 dataset (Krizhevsky et al., 2009) in the Appendix B. In addition, further experimental details appear in Appendix E.

4.1 CelebA Dataset

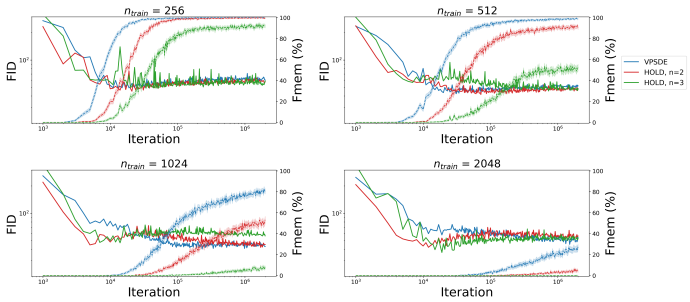
Our main experimental setup examines the memorization behavior of HOLD during training on the CelebA dataset (Liu et al., 2015) for different data set sizes. To facilitate extensive examination, before training, the images are reduced from the center to size 32×32, and converted from RGB to grayscale. These are the exact same preprocessing steps that (Bonnaire et al., 2025) uses; we also use the same UNet architecture and data splits. The only architectural deviation was the additional input required for the auxiliary variables.

Figure 2 presents the results of the FID and Fmem during training and as Pareto curves for four different dataset sizes n_{train}. From inspection of this figure, HOLD orders 2 and 3 respectively improve memorization resistance for comparable FID levels at each selected n_{train}. The data samples generated in Figure 3 confirm these findings (See Appendix D for more visual results). Eight generated samples are drawn from each model and compared with their nearest neighbors in the training set. Most of the images generated by the VPSDE diffusion model resemble certain training data images either closely or identically. The HOLD case $n = 2$ significantly reduces the percentage of memorization and, for the most part, produces images that look quite different from their nearest training neighbors. Finally, HOLD $n = 3$ further improves HOLD $n = 2$ in the same way. Each diffusion model produces nearly equal FIDs at this number of training iterations, and visual image quality is similar overall. To verify that these results are not simply a result of using a network that is too small, the experiments on the CelebA dataset were rerun after quadrupling the network width, increasing it from $W = 32$ to $W = 128$. These results are presented in Appendix C.2. It is clearly visible that HOLD still delays memorization further than the VPSDE model, even after this increase in network width.

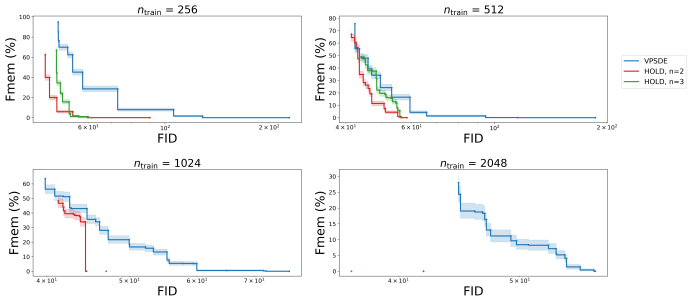
5 Conclusion

Although diffusion models offer unparalleled generated image quality compared to earlier methods, they are also susceptible to memorization. Previous works have suggested early stopping as a solution, but when this is inconvenient to do, there are not many other techniques in the literature that are designed to prevent model memorization. The theoretical and empirical results presented in this work suggest that HOLD models may be used to perform statistical shrinkage on both the training and sampling procedures, alleviating this problem that is otherwise difficult to measure. One future point of work would be further exploration of diffusion models that implicitly act as low-pass filters on their score function. This work did not explore optimizing the forward SDE to obtain optimal filtering properties. It would be worthwhile to explore alternate models inspired by classical signal processing, and to explore whether they could potentially further reduce memorization without compromising the generation quality.

Some limitations of HOLD are that it requires a slightly larger memory footprint. However, the number of necessary parameters does not increase by a factor of n (the model order); in our ex-



(a) CelebA FIDs and fraction memorized (Fmem) percentages by the number of training samples. Memorization is presented with 95% confidence intervals.

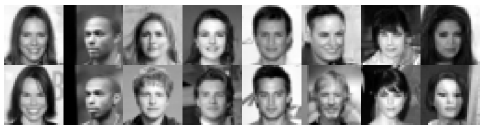


(b) Lower envelopes of the Pareto curves of the fraction memorized (Fmem) percentages by FID for CelebA. The envelopes of superior models possess a Pareto curve to the left of other curves.

Figure 2: Experiments run on the CelebA dataset. Using HOLD and increasing the model order helps move the respective Pareto frontiers to the left.



(a) VPSDE. Fmem: 27.339%, FID: 53.823



(b) HOLD $n = 2$. Fmem: 4.029%, FID: 55.819



(c) HOLD $n = 3$. Fmem: 0.101%, FID: 52.223

Figure 3: Nearest training neighbors for different models at $2e6$ training iterations with 2048 training images on the CelebA dataset. Each first row contains the generated images, and each second row contains the corresponding nearest neighbors. The VPSDE generated samples are heavily memorized, while the HOLD generated images, for the most part, are not nearly as memorized.

periments, only extra inputs were added to the network to include the auxiliary variables. It is also worth noting that $n = 4$ still achieves reasonable image qualities, but the qualities start to taper off for model orders higher than this (Sterling et al., 2025). Regarding the broader impact of our work, reducing memorization in diffusion models is largely beneficial: when training data is used with consent, it helps protect copyright and user privacy by making such data harder to extract from the trained model. On the other hand, making diffusion models less prone to memorization can also make it harder to detect when user data has been used for training without their consent.

AI Assistance

AI was used as a check for research ideation, and for dataset pre-processing in the code.

Reproducibility Statement

We have included complete hyperparameter configurations in Appendix E and provided the source code in the supplementary material to ensure full reproducibility.

Ethics Statement

This paper presents foundational theoretical work using public benchmarks. It presents no human subjects, sensitive data, or direct societal harms.

References

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020. doi: 10.1073/pnas.1907378117. URL https://www.pnas.org/doi/abs/10.1073/pnas.1907378117.

Ronen Basri, David Jacobs, Yoni Kasten, and Shira Kritchman. The convergence rate of neural networks for learned functions of different frequencies. In *Advances in Neural Information Processing Systems*, 2019.

Ronen Basri, Meirav Galun, Amnon Geifman, David Jacobs, Yoni Kasten, and Shira Kritchman. Frequency bias in neural networks for input of non-uniform density. In *International conference on machine learning*, pages 685–694. PMLR, 2020.

Tony Bonnaire, Raphaël Urfin, Giulio Biroli, and Marc Mezard. Why diffusion models don't memorize: The role of implicit dynamical regularization in training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=BSZqpqgqM0.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX security symposium (USENIX Security 23)*, pages 5253–5270, 2023.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse Image Synthesis for Multiple Domains . In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8185–8194, Los Alamitos, CA, USA, June 2020. IEEE Computer Society. doi: 10.1109/CVPR42600.2020.00821. URL https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.00821.

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped Langevin diffusion. In *International Conference on Learning Representations*, 2022.

Tomer Garber and Tom Tirer. Image restoration by denoising diffusion models with iteratively preconditioned guidance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 25245–25254, 2024.

Anand Jerry George, Rodrigo Veiga, and Nicolas Macris. Denoising score matching with random features: Insights on diffusion models from precise learning curves. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026. URL https://openreview.net/forum?id=ZnplHm2uRt.

Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in diffusion models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Zahra Kadkhodaie, Florentin Guth, Eero P Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *The Twelfth International Conference on Learning Representations*, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=k7FuTOWMOc7.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in neural information processing systems*, 35:23593–23606, 2022.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

Alan V Oppenheim, Alan S Willsky, and Syed Hamid Nawab. *Signals & systems*. Pearson Educación, 1997.

Eugene J Putzer. Avoiding the Jordan canonical form in the discussion of linear systems with constant coefficients. *The American Mathematical Monthly*, 73(1):2–7, 1966.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR, 2019.

Ziqiang Shi and Rujie Liu. Generative modelling with higher-order Langevin dynamics. *arXiv preprint arXiv:2404.12814*, 2024.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6048–6058, 2023.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Benjamin Sterling and Mónica F. Bugallo. Critically-damped third-order Langevin dynamics. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10889657.

Benjamin Sterling, Chad Gueli, and Mónica F. Bugallo. Critically-damped higher-order Langevin dynamics, 2025. URL https://arxiv.org/abs/2506.21741.

S. Särkkä and A. Solin. *Applied Stochastic Differential Equations*, volume 10. Cambridge University Press, 2019.

Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks, 2020. URL https://openreview.net/forum?id=Skgb5h4KPH.

Mingyang Yi, Jiacheng Sun, and Zhenguo Li. On the generalization of diffusion model. *arXiv preprint arXiv:2305.14712*, 2023.

TaeHo Yoon, Joo Young Choi, Sehyun Kwon, and Ernest K Ryu. Diffusion probabilistic models generalize when they fail to memorize. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023.

Huijie Zhang, Jinfan Zhou, Yifu Lu, Minzhe Guo, Peng Wang, Liyue Shen, and Qing Qu. The emergence of reproducibility and consistency in diffusion models. In *Forty-first International Conference on Machine Learning*, 2024.

A TECHNICAL PROOFS

This section of the appendix will be used to rigorously prove the theoretical claims of paper.

A.1 PROOF OF PROPOSITION 1

Proof. Start with training samples $\{\mathbf{x}_0^{(k)}\}, 1 \leq k \leq n_{\text{train}}$. The data distribution may be expressed as the following linear combinations of Dirac delta distributions:

$$p_{\text{data}}(\mathbf{x}) = \frac{1}{n_{\text{train}}} \sum_{k=1}^{n_{\text{train}}} \delta(\mathbf{x} - \mathbf{x}_0^{(k)})$$

Let us define the collection of auxiliary variables $\mathbf{v}_0^{(k)} = (\mathbf{v}_0^{(1),(k)}, \ldots, \mathbf{v}_0^{(n-1),(k)})$ such that $\mathbf{u}_0^{(k)} = \text{vec}(\mathbf{x}_0^{(k)}, \mathbf{v}_0^{(k)})$, and if we consider the lth sample of $\mathbf{v}_0^{(k)}$, $\mathbf{v}_0^{(k,l)}$, then define $\mathbf{u}_0^{(k,l)} = \text{vec}(\mathbf{x}_0^{(k)}, \mathbf{v}_0^{(k,l)})$.

$$\begin{aligned}\mathcal{L}(\mathbf{s}_\theta) &= \mathbb{E}_{t\sim\mathcal{U}(0,1),\mathbf{u}_0,\boldsymbol{\epsilon}_{\text{full}}} ||\boldsymbol{\epsilon}_n + \mathbf{s}_\theta(\mathbf{u}_t, t)\left(\mathbf{L}_t[nh, nh]\right)||^2 \\ &= \int \mathbb{E}_{t\sim\mathcal{U}(0,1),\mathbf{u}_0} ||\boldsymbol{\epsilon}_n + \mathbf{s}_\theta(\mathbf{u}_t, t)\left(\mathbf{L}_t[nh, nh]\right)||^2 p(\boldsymbol{\epsilon}_{\text{full}}) d\boldsymbol{\epsilon}_{\text{full}} = \int \mathcal{L}(\mathbf{s}_\theta, \mathbf{u}_t) d\boldsymbol{\epsilon}_{\text{full}}.\end{aligned}$$

Minimizing $\mathcal{L}(\mathbf{s}_\theta)$ may be achieved by minimizing $\mathcal{L}(\mathbf{s}_\theta, \mathbf{u}_t)$ instead. We do this by minimizing this loss over all $\mathbf{s}_\theta$ as follows. We define $\boldsymbol{\epsilon}_{\text{full}}^{(k)} = \mathbf{L}_t^{-1}\left(\mathbf{u}_t - \exp(\mathcal{F}t)\mathbf{u}_0^{(k)}\right)$ and $\boldsymbol{\epsilon}_n^{(k)} = \left[\boldsymbol{\epsilon}_{\text{full}}^{(k)}\right]_{n(h-1):nh}$; that is the final n entries of the vector $\boldsymbol{\epsilon}_{\text{full}}^{(k}$. Both come from the kth sample.

$$\mathcal{L}(\mathbf{s}_\theta, \mathbf{u}_t) = \frac{1}{n_{\text{train}}} \sum_{k=1}^{n_{\text{train}}} \mathbb{E}_{\mathbf{v}_0}\left(||\boldsymbol{\epsilon}_n^{(k)} + \mathbf{s}_\theta(\mathbf{u}_t, t)\left(\mathbf{L}_t[nh, nh]\right)||^2 p(\boldsymbol{\epsilon}_{\text{full}}^{(k)})\right).$$

Furthermore, if we approximate $\mathbb{E}_{\mathbf{v}_0}$ with a Monte Carlo approximation with batch size B instead of a true expectation, then:

$$\mathcal{L}(\mathbf{s}_\theta, \mathbf{u}_t) = \frac{1}{Bn_{\text{train}}} \sum_{k=1}^{n_{\text{train}}} \sum_{l=1}^{B} ||\boldsymbol{\epsilon}_n^{(k,l)} + \mathbf{s}_\theta(\mathbf{u}_t, t)\left(\mathbf{L}_t[nh, nh]\right)||^2 p(\boldsymbol{\epsilon}_{\text{full}}^{(k,l)}).$$

Take the gradient with respect to $\mathbf{s}_\theta$ and set it to zero:

$$\sum_{k=1}^{n_{\text{train}}} \sum_{l=1}^{B} \left(\boldsymbol{\epsilon}_n^{(k,l)} + \mathbf{s}_\theta(\mathbf{u}_t, t)\left(\mathbf{L}_t[nh, nh]\right)\right) p(\boldsymbol{\epsilon}_{\text{full}}^{(k,l)}) = 0$$

$$\mathbf{s}_\theta(\mathbf{u}_t, t) = -\frac{\sum_{k=1}^{n_{\text{train}}} \sum_{l=1}^{B} p(\boldsymbol{\epsilon}_{\text{full}}^{(k,l)}) \boldsymbol{\epsilon}_n^{(k,l)} / \mathbf{L}_t[nh, nh]}{\sum_{k=1}^{n_{\text{train}}} \sum_{l=1}^{B} p(\boldsymbol{\epsilon}_{\text{full}}^{(k,l)})} = \frac{\sum_{k=1}^{n_{\text{train}}} \sum_{l=1}^{B} p(\boldsymbol{\epsilon}_{\text{full}}^{(k,l)}) \nabla_{\mathbf{v}_t^{(n-1)}} \log p(\mathbf{u}_t \mid \mathbf{u}_0^{(k,l)})}{\sum_{k=1}^{n_{\text{train}}} \sum_{l=1}^{B} p(\boldsymbol{\epsilon}_{\text{full}}^{(k,l)})}$$

$$= \left[\frac{\sum_{k=1}^{n_{\text{train}}} \sum_{l=1}^{B} p(\boldsymbol{\epsilon}_{\text{full}}^{(k,l)}) \nabla_{\mathbf{u}_t} \log p(\mathbf{u}_t \mid \mathbf{u}_0^{(k,l)})}{\sum_{k=1}^{n_{\text{train}}} \sum_{l=1}^{B} p(\boldsymbol{\epsilon}_{\text{full}}^{(k,l)})}\right]_{n(h-1):nh}.$$

If one calculates the score function corresponding to p_t^{emp}, then it follows that $\mathbf{s}_\theta(\mathbf{u}_t, t) = \left[\nabla_{\mathbf{u}_t} \log p_t^{\text{emp}}(\mathbf{u}_t)\right]_{n(h-1):nh}$. Where:

$$p_t^{\text{emp}}(\mathbf{u}) = \frac{1}{Bn_{\text{train}}} \sum_{k=1}^{n_{\text{train}}} \sum_{l=1}^{B} (2\pi \det \boldsymbol{\Sigma}_t)^{-h/2} \exp\left(-\frac{1}{2}\left|\left|\boldsymbol{\Sigma}_t^{-1/2}\left(\mathbf{u} - \exp(\mathcal{F}t)\mathbf{u}_0^{(k,l)}\right)\right|\right|^2\right)$$

□

A.2 PROOF OF LEMMA 1

Proof. Note that $\tilde{\mathbf{u}}(s) = \text{vec}(\tilde{\mathbf{x}}(s), \tilde{\mathbf{v}}^{(1)}(s), \ldots \tilde{\mathbf{v}}^{(n-1)}(s)), \tilde{\mathbf{S}}(s) = \mathbf{S}_\theta(\mathbf{u}_t, t)$.

$$\begin{aligned}\frac{d\mathbf{u}_t}{dt} &= \mathcal{F}\mathbf{u}_t - \xi L^{-1}\mathbf{E}_{n,n}\mathbf{S}_\theta(\mathbf{u}_t, t) \\ s\tilde{\mathbf{u}}(s) - \mathbf{u}_0 &= \mathcal{F}\tilde{\mathbf{u}}(s) - \xi L^{-1}\mathbf{E}_{n,n}\tilde{\mathbf{S}}(s) \\ ((\mathbf{F} - s\mathbf{I}) \otimes \mathbf{I}_h)\tilde{\mathbf{u}}(s) &= \xi L^{-1}\mathbf{E}_{n,n}\tilde{\mathbf{S}}(s) - \mathbf{u}_0.\end{aligned}$$

Now, use Cramer's Rule to solve for $\tilde{\mathbf{x}}(s)$. Take $\mathbf{0}, \mathbf{1} \in \mathbb{R}^h$ as the vector of all zeros and ones respectively. blockdet takes determinants across each coordinate of the following vectors and returns them in a single vector. $P(s) = \det(\mathbf{F} - s\mathbf{I})$ is the characteristic polynomial of $\mathbf{F}$. The second line follows from the multilinearity of the determinant.

$$\begin{aligned}
\tilde{\mathbf{x}}(s) &= \frac{1}{P(s)}\text{blockdet}\begin{pmatrix} -\mathbf{x}_0 & \gamma_1\mathbf{1} & \dots & \mathbf{0} \\ -\mathbf{v}_0^{(1)} & -s\mathbf{1} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ -\mathbf{v}_0^{(n-2)} & \dots & -s\mathbf{1} & \gamma_{n-1}\mathbf{1} \\ \xi L^{-1}\tilde{\mathbf{s}}(s) - \mathbf{v}_0^{(n-1)} & \dots & -\gamma_{n-1}\mathbf{1} & (-s-\xi)\mathbf{1} \end{pmatrix} \\
&= \frac{\xi L^{-1}}{P(s)}\text{blockdet}\begin{pmatrix} \mathbf{0} & \gamma_1\mathbf{1} & \dots & \mathbf{0} \\ \mathbf{0} & -s\mathbf{1} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ \mathbf{0} & \dots & -s\mathbf{1} & \gamma_{n-1}\mathbf{1} \\ \tilde{\mathbf{s}}(s) & \dots & -\gamma_{n-1}\mathbf{1} & (-s-\xi)\mathbf{1} \end{pmatrix} \\
&\quad + \frac{1}{P(s)}\text{blockdet}\begin{pmatrix} -\mathbf{x}_0 & \gamma_1\mathbf{1} & \dots & \mathbf{0} \\ -\mathbf{v}_0^{(1)} & -s\mathbf{1} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ -\mathbf{v}_0^{(n-2)} & \dots & -s\mathbf{1} & \gamma_{n-1}\mathbf{1} \\ -\mathbf{v}_0^{(n-1)} & \dots & -\gamma_{n-1}\mathbf{1} & (-s-\xi)\mathbf{1} \end{pmatrix} \\
&= \frac{(-1)^{n-1}\xi L^{-1}\tilde{\mathbf{s}}(s)}{P(s)}\det\begin{pmatrix} \gamma_1 & 0 & \dots & 0 \\ -s & \gamma_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & 0 & -s & \gamma_{n-1} \end{pmatrix} + \tilde{\mathbf{x}}^{\text{natural}}(s) \\
\tilde{\mathbf{x}}(s) &= -\frac{\bar{\gamma}\xi L^{-1}\tilde{\mathbf{s}}(s)}{P(s)} + \tilde{\mathbf{x}}^{\text{natural}}(s).
\end{aligned}$$

where $\tilde{\mathbf{x}}^{\text{natural}}(s)$ is given by:

$$\tilde{\mathbf{x}}^{\text{natural}}(s) = \frac{1}{P(s)}\text{blockdet}\begin{pmatrix} -\mathbf{x}_0 & \gamma_1\mathbf{1} & \dots & \mathbf{0} \\ -\mathbf{v}_0^{(1)} & -s\mathbf{1} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ -\mathbf{v}_0^{(n-2)} & \dots & -s\mathbf{1} & \gamma_{n-1}\mathbf{1} \\ -\mathbf{v}_0^{(n-1)} & \dots & -\gamma_{n-1}\mathbf{1} & (-s-\xi)\mathbf{1} \end{pmatrix}.$$

The last determinant is simply the product of gammas because the matrice's main diagonal contains the gammas, and one minor row filled with $-s$. Note that the negative characteristic polynomial is also a valid characteristic polynomial, but we take the polynomial with a leading positive coefficient. □

A.3 Proof of Theorem 1

Proof. According to Lemma 1,

$$\tilde{\mathbf{x}}(s) = -\frac{\bar{\gamma}\xi L^{-1}\tilde{\mathbf{s}}(s)}{(s-s_*)^n} + \tilde{\mathbf{x}}^{\text{natural}}(s).$$

This equation assumes that we use the critically-damped parameter selection. Further define $H(s) = -\frac{\bar{\gamma}\xi L^{-1}}{(s-s_*)^n}$. To solve for h_t, we must take the inverse Laplace transform:

$$h_t = \frac{1}{2\pi i}\lim_{T\to\infty}\int_{\gamma-iT}^{\gamma+iT} H(s)\exp(st)ds$$

for some $\gamma > 0$. Now we utilize the Residue Theorem:

$$\begin{aligned}
h_t &= \text{Res}(H(s)\exp(st)) \mid_{s=s_*} \\
&= \frac{1}{(n-1)!}\lim_{s\to s_*}\frac{d^{n-1}}{ds^{n-1}}\left(H(s)\exp(st)(s-s_*)^n\right) \\
&= \frac{-\bar{\gamma}\xi L^{-1}}{(n-1)!}\lim_{s\to s_*}\frac{d^{n-1}}{ds^{n-1}}\left(\exp(st)\right) \\
&= \frac{-\bar{\gamma}\xi L^{-1}t^{n-1}\exp(s_*t)}{(n-1)!} = \frac{-\bar{\gamma}n\sqrt{2n-3}L^{-1}t^{n-1}\exp(-t\sqrt{2n-3})}{(n-1)!}.
\end{aligned}$$

Finally, the Convolution Theorem applied to $\tilde{\mathbf{x}}(s) = H(s)\tilde{\mathbf{s}}(s) + \tilde{\mathbf{x}}^{\text{natural}}(s)$ proves the theorem. □

A.4 Proof of Proposition 2

Proof. For the purposes of this analysis, use $\mathbf{F}$ instead of $\mathcal{F} = \mathbf{F} \otimes \mathbf{I}_h$. The proof works out identically since for any matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\exp(\mathbf{A}\otimes\mathbf{I}_h) = \exp(\mathbf{A})\otimes\mathbf{I}_h$. Let us analyze the covariance matrix as $t \to 0^+$:

$$\boldsymbol{\Sigma}_t = \mathbf{I} - \exp(\mathbf{F}t)\exp(\mathbf{F}t)^T$$

A Taylor Expansion of each $\exp(\mathbf{F}t)$ at $t = 0$ yields

$$\mathbf{I} - \exp(\mathbf{F}t)\exp(\mathbf{F}t)^T = -t(\mathbf{F}+\mathbf{F}^T) - \frac{t^2}{2}\left((\mathbf{F}+\mathbf{F}^T)^2 + \mathbf{F}\mathbf{F}^T - \mathbf{F}^T\mathbf{F}\right) + \mathcal{O}(t^3)$$

We note that $\mathbf{F} + \mathbf{F}^T = 2\xi L^{-1}\mathbf{E}_{n,n}$. Define $\delta_{j,k}(i,l) = \delta(i-j)\delta(l-k)$, and $A_{j,k} = \sum_{i=1}^{n-1}\gamma_i\left(\delta_{j,k}(i,i+1) - \delta_{j,k}(i+1,i)\right)$.

$$\begin{aligned}
\mathbf{F} &= \sum_{i=1}^{n-1}\gamma_i\left(\mathbf{E}_{i,i+1} - \mathbf{E}_{i+1,i}\right) - \xi\mathbf{E}_{n,n} \\
\mathbf{F}_{j,k} &= \sum_{i=1}^{n-1}\gamma_i\left(\delta_{j,k}(i,i+1) - \delta_{j,k}(i+1,i)\right) - \xi\delta_{j,k}(n,n) \\
&= A_{j,k} - \xi\delta_{j,k}(n,n) \\
(\mathbf{F}\mathbf{F}^T - \mathbf{F}^T\mathbf{F})_{j,k} &= \sum_{l=1}^{n}\left(\mathbf{F}_{j,l}\mathbf{F}_{k,l} - \mathbf{F}_{l,j}\mathbf{F}_{l,k}\right) \\
&= \sum_{l=1}^{n}(A_{j,l} - \xi\delta_{j,l}(n,n))(A_{k,l} - \xi\delta_{k,l}(n,n)) \\
&\quad - \sum_{l=1}^{n}(A_{l,j} - \xi\delta_{l,j}(n,n))(A_{l,k} - \xi\delta_{l,k}(n,n))
\end{aligned}$$

Note that $A_{j,k}$ are the entries from a skew symmetric matrix, therefore $A_{j,k} = -A_{k,j}$.

$$\begin{aligned}
(\mathbf{F}\mathbf{F}^T - \mathbf{F}^T\mathbf{F})_{j,k} &= \sum_{l=1}^{n}(A_{j,l}A_{k,l} + \xi^2\delta_{j,l}(n,n)\delta_{k,l}(n,n) - A_{j,l}\xi\delta_{k,l}(n,n) - A_{k,l}\xi\delta_{j,l}(n,n)) \\
&\quad - \sum_{l=1}^{n}(A_{l,j}A_{l,k} + \xi^2\delta_{l,j}(n,n)\delta_{l,k}(n,n) - A_{l,j}\xi\delta_{l,k}(n,n) - A_{l,k}\xi\delta_{l,j}(n,n))
\end{aligned}$$

Using the aforementioned property:

$$\begin{aligned}(\mathbf{F}\mathbf{F}^T - \mathbf{F}^T\mathbf{F})_{j,k} = &\sum_{l=1}^{n}(-A_{j,l}\xi\delta_{k,l}(n,n) - A_{k,l}\xi\delta_{j,l}(n,n)) \\ &+ \sum_{l=1}^{n}(A_{l,j}\xi\delta_{l,k}(n,n) + A_{l,k}\xi\delta_{l,j}(n,n))\end{aligned}$$

By inspection for $j < n-1, k < n-1$, and $j = k = n-1$, $(\mathbf{F}\mathbf{F}^T - \mathbf{F}^T\mathbf{F})_{j,k} = 0$. This leaves 3 possible entries to consider $(j,k) = \{(n-1,n),(n,n-1),(n,n)\}$. For these cases:

$$\begin{aligned}(\mathbf{F}\mathbf{F}^T - \mathbf{F}^T\mathbf{F})_{n-1,n} &= \sum_{l=1}^{n} A_{l,n-1}\xi\delta_{l,n}(n,n) - A_{n-1,l}\xi\delta_{n,l}(n,n) \\ &= A_{n,n-1}\xi - A_{n-1,n}\xi \\ &= 2\xi A_{n,n-1} = 2\xi\gamma_{n-1}\end{aligned}$$

Similar calculation yields $(\mathbf{F}\mathbf{F}^T - \mathbf{F}^T\mathbf{F})_{n,n-1} = 2\xi\gamma_{n-1}$. Lastly:

$$\begin{aligned}(\mathbf{F}\mathbf{F}^T - \mathbf{F}^T\mathbf{F})_{n,n} = &\sum_{l=1}^{n}(-A_{n,l}\xi\delta_{n,l}(n,n) - A_{n,l}\xi\delta_{n,l}(n,n)) \\ &+ \sum_{l=1}^{n}(A_{l,n}\xi\delta_{l,n}(n,n) + A_{l,n}\xi\delta_{l,n}(n,n)) \\ &= -2A_{n,n}\xi + 2A_{n,n}\xi = 0\end{aligned}$$

This analysis concludes that every entry of $\mathbf{I} - \exp(\mathbf{F}t)\exp(\mathbf{F}t)^T$ is $\mathcal{O}(t^3)$ as $t \to 0^+$, except those at $(n-1,n)$ and $(n,n-1)$.

Lastly, derive the characteristic function of $p^{\text{emp}}_{\mathbf{x},t} = \int p^{\text{emp}}_t d\mathbf{v}_0^{(1)}, \ldots, d\mathbf{v}_0^{(n-1)}$ here. Firstly, evaluate this marginal, with a slight abuse of notation:

$$\begin{aligned}p^{\text{emp}}_t(\mathbf{u}) &= \frac{1}{n_{\text{train}}}\sum_{k=1}^{n_{\text{train}}}\mathcal{N}(\mathbf{u} \mid \boldsymbol{\mu}_t^{(k)}, \boldsymbol{\Sigma}_t) \\ p^{\text{emp}}_{\mathbf{x},t}(\mathbf{x}) &= \frac{1}{n_{\text{train}}}\sum_{k=1}^{n_{\text{train}}}\int\mathcal{N}(\mathbf{x}, \mathbf{v}^{(1)}, \ldots \mathbf{v}^{(n-1)} \mid \boldsymbol{\mu}_t^{(k)}, \boldsymbol{\Sigma}_t)d\mathbf{v}^{(1)}\ldots d\mathbf{v}^{(n-1)} \\ &= \frac{1}{n_{\text{train}}}\sum_{k=1}^{n_{\text{train}}}\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_t^{(0),(k)}, \boldsymbol{\Sigma}_t^{(0,0)}).\end{aligned}$$

The characteristic function of $p^{\text{emp}}_{\mathbf{x},t}(\mathbf{x})$, $\Phi_{\mathbf{x}}(\mathbf{k}) = \mathbb{E}\left(\exp(i\mathbf{k}^T\mathbf{x}_t)\right)$, may be computed directly from its expectation definition. Note that the characteristic function of the distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is $\exp(i\boldsymbol{\mu}^T\mathbf{k} - \frac{1}{2}\mathbf{k}^T\boldsymbol{\Sigma}\mathbf{k})$

$$
\begin{aligned}
\Phi_{\mathbf{x}}(\mathbf{k}) &= \int \exp(i\mathbf{k}^T\mathbf{x}_t)\left(\frac{1}{n_{\text{train}}}\sum_{k=1}^{n_{\text{train}}}\mathcal{N}(\mathbf{x}_t \mid \boldsymbol{\mu}_t^{(0),(k)}, \boldsymbol{\Sigma}_t^{(0,0)})\right)d\mathbf{x}_t \\
&= \frac{1}{n_{\text{train}}}\sum_{k=1}^{n_{\text{train}}}\int \exp(i\mathbf{k}^T\mathbf{x}_t)\left(\mathcal{N}(\mathbf{x}_t \mid \boldsymbol{\mu}_t^{(0),(k)}, \boldsymbol{\Sigma}_t^{(0,0)})\right)d\mathbf{x}_t \\
&= \frac{1}{n_{\text{train}}}\sum_{k=1}^{n_{\text{train}}}\exp\left(i(\boldsymbol{\mu}_t^{(0),(k)})^T\mathbf{k} - \frac{1}{2}\mathbf{k}^T\boldsymbol{\Sigma}_t^{(0,0)}\mathbf{k}\right) \\
&= \frac{1}{n_{\text{train}}}\exp\left(-\frac{1}{2}\mathbf{k}^T\boldsymbol{\Sigma}_t^{(0,0)}\mathbf{k}\right)\sum_{k=1}^{n_{\text{train}}}\exp\left(i(\boldsymbol{\mu}_t^{(0),(k)})^T\mathbf{k}\right) \\
|\Phi_{\mathbf{x}}(\mathbf{k})| &= \frac{1}{n_{\text{train}}}\exp\left(-\frac{1}{2}\mathbf{k}^T\boldsymbol{\Sigma}_t^{(0,0)}\mathbf{k}\right)\left|\sum_{k=1}^{n_{\text{train}}}\exp\left(i(\boldsymbol{\mu}_t^{(0),(k)})^T\mathbf{k}\right)\right|.
\end{aligned}
$$

Calculate the following, recalling $|z|^2 = zz^*$

$$
\begin{aligned}
\left|\sum_{k=1}^{n_{\text{train}}}\exp\left(i(\boldsymbol{\mu}_t^{(0),(k)})^T\mathbf{k}\right)\right|^2 &= \sum_{j=1}^{n_{\text{train}}}\exp\left(i(\boldsymbol{\mu}_t^{(0),(j)})^T\mathbf{k}\right)\sum_{k=1}^{n_{\text{train}}}\exp\left(-i(\boldsymbol{\mu}_t^{(0),(k)})^T\mathbf{k}\right) \\
&= \sum_{j=1}^{n_{\text{train}}}\sum_{k=1}^{n_{\text{train}}}\exp\left(i(\boldsymbol{\mu}_t^{(0),(j)} - \boldsymbol{\mu}_t^{(0),(k)})^T\mathbf{k}\right) \\
&= \sum_{j=1}^{n_{\text{train}}}\sum_{k=1}^{n_{\text{train}}}\left(\cos\left((\boldsymbol{\mu}_t^{(0),(j)} - \boldsymbol{\mu}_t^{(0),(k)})^T\mathbf{k}\right) + i\sin\left((\boldsymbol{\mu}_t^{(0),(j)} - \boldsymbol{\mu}_t^{(0),(k)})^T\mathbf{k}\right)\right)
\end{aligned}
$$

However, note that sin is an odd function, therefore:

$$
\begin{aligned}
&\sum_{j=1}^{n_{\text{train}}}\sum_{k=1}^{n_{\text{train}}} i\sin\left((\boldsymbol{\mu}_t^{(0),(j)} - \boldsymbol{\mu}_t^{(0),(k)})^T\mathbf{k}\right) \\
&= \sum_{1\le j<k\le n_{\text{train}}}\left(i\sin\left((\boldsymbol{\mu}_t^{(0),(j)} - \boldsymbol{\mu}_t^{(0),(k)})^T\mathbf{k}\right) - i\sin\left((\boldsymbol{\mu}_t^{(0),(k)} - \boldsymbol{\mu}_t^{(0),(j)})^T\mathbf{k}\right)\right) \\
&= 0
\end{aligned}
$$

Thus:

$$
|\Phi_{\mathbf{x}}(\mathbf{k})| = \frac{1}{n_{\text{train}}}\exp\left(-\frac{1}{2}\mathbf{k}^T\boldsymbol{\Sigma}_t^{(0,0)}\mathbf{k}\right)\sqrt{\sum_{j=1}^{n_{\text{train}}}\sum_{k=1}^{n_{\text{train}}}\cos\left((\boldsymbol{\mu}_t^{(0),(k)} - \boldsymbol{\mu}_t^{(0),(j)})^T\mathbf{k}\right)}.
$$

To connect this formula back to the score, express the empirical score function as:

$$
\mathbf{s}_{\text{emp}}(\mathbf{x}) = \nabla_{\mathbf{x}}\log p_{\mathbf{x},t}^{\text{emp}}(\mathbf{x}) = \frac{\nabla_{\mathbf{x}}p_{\mathbf{x},t}^{\text{emp}}(\mathbf{x})}{p_{\mathbf{x},t}^{\text{emp}}(\mathbf{x})}
$$

Note that the classical Fourier Transform of the score is undefined, even for Gaussian data. The score in this case would be proportional to $-\mathbf{x}$, and $\int_{\mathbb{R}^h} |-\mathbf{x}|d\mathbf{x}$ diverges. Instead, define the Fourier Transform with the following inner product (it simplifies with integration by parts):

$$
\begin{aligned}
\mathcal{F}\{\mathbf{s}_{\text{emp}}\}(\mathbf{k}) &= \mathbb{E}(\mathbf{s}_{\text{emp}}(\mathbf{x})\exp(i\mathbf{k}^T\mathbf{x})) \\
&= \int \nabla_{\mathbf{x}}p_{\mathbf{x},t}^{\text{emp}}(\mathbf{x})\exp(i\mathbf{k}^T\mathbf{x})d\mathbf{x} \\
&= -i\mathbf{k}\mathbb{E}(\exp(i\mathbf{k}^T\mathbf{x})) \\
&= -i\mathbf{k}\Phi_{\mathbf{x}}(\mathbf{k})
\end{aligned}
$$

Finally, we arrive at:

$$||\mathcal{F}\{\mathbf{s}_{\text{emp}}\}(\mathbf{k})|| = \frac{||\mathbf{k}||}{n_{\text{train}}} \exp\left(-\frac{1}{2}\mathbf{k}^T \boldsymbol{\Sigma}_t^{(0,0)} \mathbf{k}\right) \sqrt{\sum_{j=1}^{n_{\text{train}}} \sum_{k=1}^{n_{\text{train}}} \cos\left((\boldsymbol{\mu}_t^{(0),(k)} - \boldsymbol{\mu}_t^{(0),(j)})^T \mathbf{k}\right)}$$

□

A.5 Derivation of Ornstein–Uhlenbeck Convolution

The Ornstein–Uhlenbeck convolution formula is derived from the deterministic ODE as follows

$$\begin{aligned} d\mathbf{x}_t &= \left(-\xi \mathbf{x}_t - \frac{2\xi L^{-1}}{2} \mathbf{s}_\theta(\mathbf{x}_t, t)\right) dt \\ \frac{d\mathbf{x}_t}{dt} + \xi \mathbf{x}_t &= -\xi L^{-1} \mathbf{s}_\theta(\mathbf{x}_t, t) \\ \frac{d}{dt}\left(\exp(\xi t)\mathbf{x}_t\right) &= -\xi L^{-1} \exp(\xi t) \mathbf{s}_\theta(\mathbf{x}_t, t) \\ \mathbf{x}_t &= \exp(-\xi t)\mathbf{x}_0 - \xi L^{-1} \int_0^t \exp(-\xi(t-\tau)) \mathbf{s}_\theta(\mathbf{x}_\tau, \tau) d\tau. \end{aligned}$$

B Additional Color Experiments

B.1 Animal Faces dataset

In order to verify that CelebA results generalize to larger color datasets, we perform the same experiments on the Animal Faces (AFHQ) dataset (Choi et al., 2020), shrunk to 64×64 pixels, and a network width of $W = 64$ is used. The FID and Fmem plots for this dataset are presented in Figure 4. For $n_{\text{train}} = 256, 512$, a clear benefit can be observed by choosing $n = 2$ over the standard VPSDE process. $n = 3$ does not converge well in $n_{\text{train}} = 256$, but outperforms the standard VPSDE process in $n_{\text{train}} = 512$. At $n_{\text{train}} = 1024, 2048$, memorization does not occur in a large way for any order of the model, but all the qualities of the images are roughly comparable. The data samples generated in Figure 5 visually confirm the improved memorization in $n_{\text{train}} = 512$. The VPSDE samples are largely memorized, whereas the samples in $n = 2$ are good quality, but are not learned from a single training image. At $n = 3$, no memorization occurs, but image quality also drops. The Pareto curves in Figure 4b summarize these results.

B.2 CIFAR-10 Dataset

We perform similar experiments with the same UNet architecture in the CIFAR-10 dataset (Krizhevsky et al., 2009), and similar conclusions hold here. The only preprocessing step employed is that we keep only the car and truck classes (other images are discarded from both training and validation datasets), so as not to replicate the types of images from Animal Faces and to narrow the pool of images so that the networks are capable of generating meaningful images with $n_{\text{train}} = 256, 512, 1024, 2048$. If all ten classes were kept, then each network would be exposed to roughly $n_{\text{train}} \div 10$ images of each class. Figures 6 and 7, similarly to Figures 2 and 3, demonstrate that higher order dynamics are less prone to memorization.

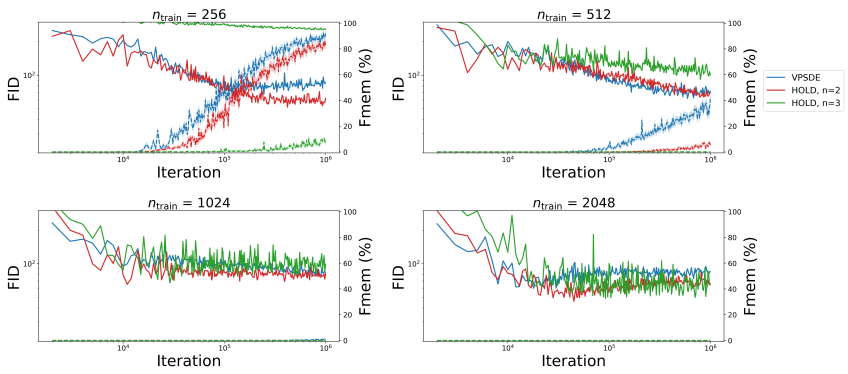
C Mini-Experiments

C.1 NUFT spectrum on a 1-D two-mode Gaussian mixture: HOLD vs. OU

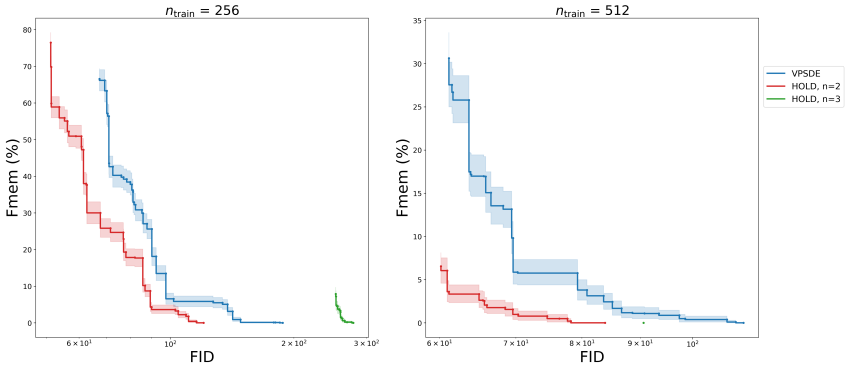
In order to visualize the differences in frequency spectra between the HOLD and OU processes, Figure 8 graphs these spectra for simple 1-dimensional data that comes from the following distribution:

$$\pi(x) = \frac{1}{2}\mathcal{N}(x \mid 1, 0.1) + \frac{1}{2}\mathcal{N}(x \mid -1, 0.1)$$

It is clear that HOLD preserves higher frequencies for the early-time diffusion regime.



(a) AFHQ FIDs and fraction memorized (Fmem) percentages by the number of training samples. Memorization is presented with 95% confidence intervals.



(b) Lower envelopes of the Pareto curves of the fraction memorized (Fmem) percentages by FID for AFHQ. The envelopes of superior models possess a Pareto curve to the left of other curves. Only $n_{\text{train}} = 256, 512$ are presented because $n_{\text{train}} = 1024, 2048$ do not memorize, which do not make for meaningful Pareto curves.

Figure 4: Experiments run on the AFHQ dataset. Using HOLD instead of the VPSDE helps move the respective Pareto frontier of $n = 2$ to the left.

C.2 CelebA Experiment on a Larger Network

When increasing the network width of the Celeba experiments from $W = 32$ to $W = 128$, memorization of each SDE type occurs faster, but the HOLD models still largely outperform the VPSDE models as evidenced by Figure 9. While the VPSDE Pareto curves touch the HOLD curves at $n_{\text{train}} = 512, 2048$, they largely occur due to the faster onset of memorization, and the HOLD curves are generally further to the left than the VPSDE curves.

C.3 Training Losses

This section contains Figure 10, that presents the training losses for $n_{\text{train}} = 256$ for the VPSDE and HOLD $n = 2, 3$. The slightly higher training losses of HOLD $n = 2$ and $n = 3$ respectively may be explained by the difficulty of learning the optimal empirical score and statistical regularization that is proposed in this work.

D Additional CelebA images with FIDs and Fmem

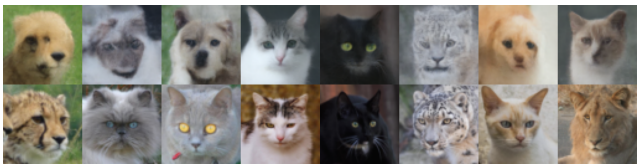
This appendix section presents more generated samples from the CelebA experiment for different number of training images n_{train}.



(a) VPSDE. Fmem: 41.069%, FID: 72.162



(b) HOLD $n = 2$. Fmem: 5.438%, FID: 69.562

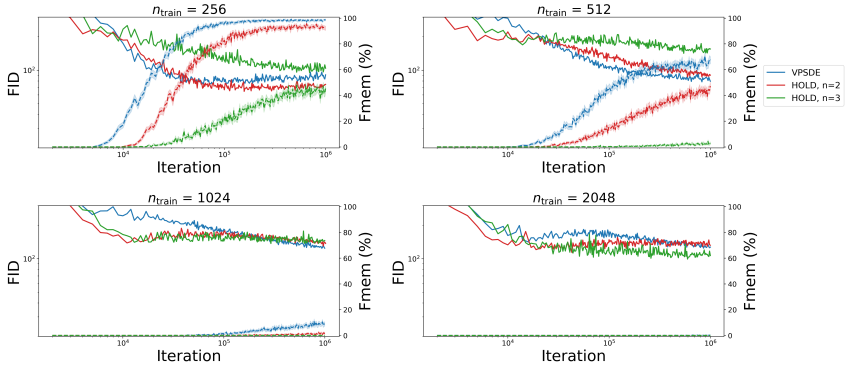


(c) HOLD $n = 3$. Fmem: 0.000%, FID: 100.767

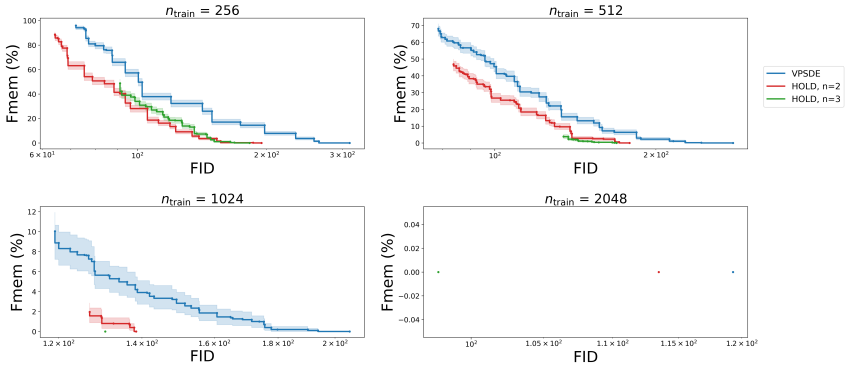
Figure 5: Nearest training neighbors for different models at $1e6$ training iterations with 512 training images on the Animal Faces 64×64 dataset. Each first row contains the generated images, and each second row contains the corresponding nearest neighbors. The VPSDE generated samples are heavily memorized and contain several dominant images, while the HOLD generated images are not nearly as memorized and contain better sample diversity. While the $n = 3$ images are not as high quality as the other two image sets, they are not memorized at all, indicating some quality-memorization trade-off. However, $n = 2$ produces better quality samples with less memorization than the VPSDE process.

E MISCELLANEOUS DETAILS

For all experiments, a learning rate of 1×10^{-4}, the Adam optimizer, a dropout rate of 0.1, and 1000 diffusion model steps were used. The standard CelebA and CIFAR-10 experiments used 32 base channels, and the Animal Faces experiment used 64 base channels. For the VPSDE, a linear noise schedule with $\beta_0 = 1 \times 10^{-3}, \beta_1 = 10.0$ was used, and the HOLD runs used $L^{-1} = 1.0$. The UNet (the same architecture as in (Bonnaire et al., 2025)) uses channel multipliers (1, 2, 4) with self-attention applied at the middle two resolution levels. Experiments were conducted on a single NVIDIA H200 SXM5 GPU, with 141 GB of VRAM. For each experiment, training and generation combined took approximately 5 days of wall clock time, and separate FID and Fmem calculations took negligible amounts of time.



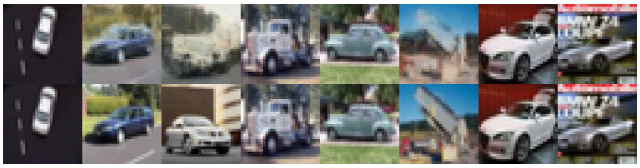
(a) CIFAR-10 FIDs and fraction memorized (Fmem) percentages by the number of training samples. Memorization presented with 95% confidence intervals.



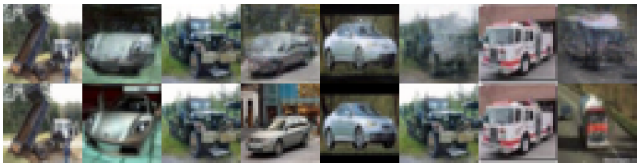
(b) Lower envelopes of the Pareto curves of the fraction memorized (Fmem) percentages by FID for CIFAR-10. The envelopes of superior models possess a Pareto curve to the left of other curves.

Figure 6: Experiments run on the CIFAR-10 dataset. Using HOLD and increasing the model order helps move the respective Pareto frontiers to the left

.



(a) VPSDE. Fmem: 67.293%, FID: 79.441



(b) HOLD $n = 2$. Fmem: 47.209%, FID: 90.108



(c) HOLD $n = 3$. Fmem: 2.837%, FID: 154.836

Figure 7: Nearest training neighbors for different models at $1{,}000{,}000$ training iterations with 512 training images on the CIFAR-10 dataset excluded to cars and trucks. Each first row contains the generated images, and each second row contains the corresponding nearest neighbors.

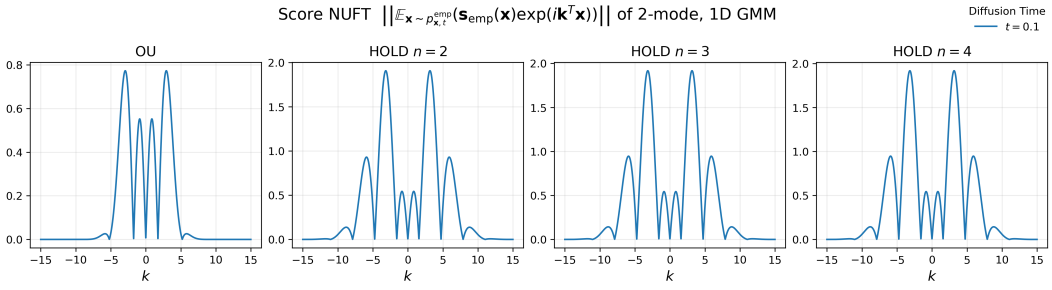
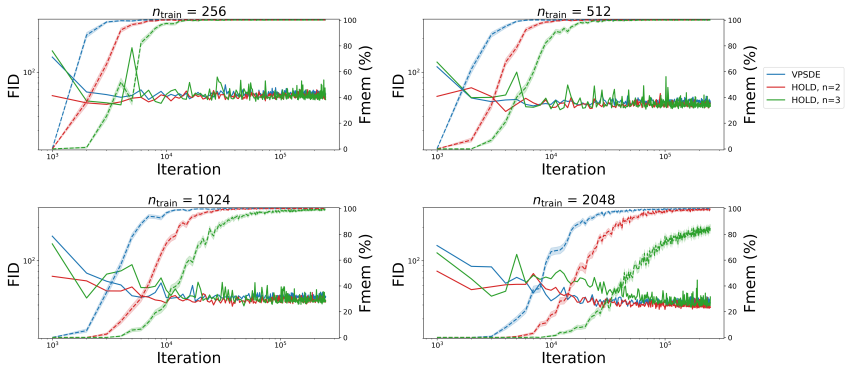
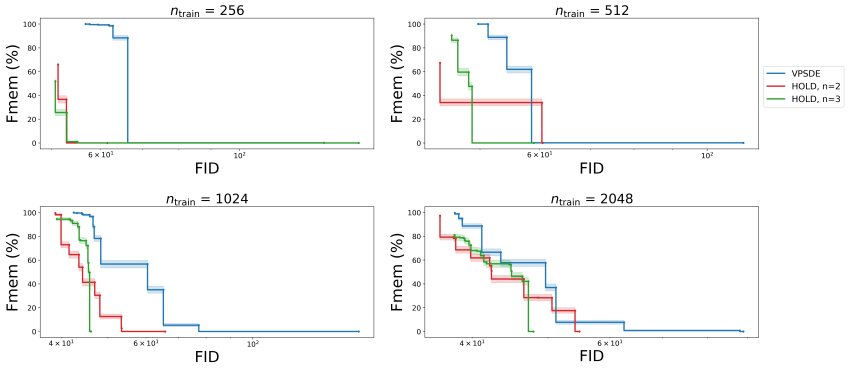


Figure 8: Non-Uniform Fourier Transforms of the OU and HOLD score functions at time $t = 0.1$. This illustrates that the HOLD processes possess wider frequency spectra at early times.



(a) CelebA FIDs and fraction memorized (Fmem) percentages by the number of training samples after quadrupling the network width in Figure 2. Memorization is presented with 95% confidence intervals. The comparative memorization mitigation observed with HOLD from the initial experiment presented in Figure 2 still holds.



(b) Lower envelopes of the Pareto curves of the fraction memorized (Fmem) percentages by FID for CelebA with the quadrupled network width. The envelopes of superior models possess a Pareto curve to the left of other curves.

Figure 9

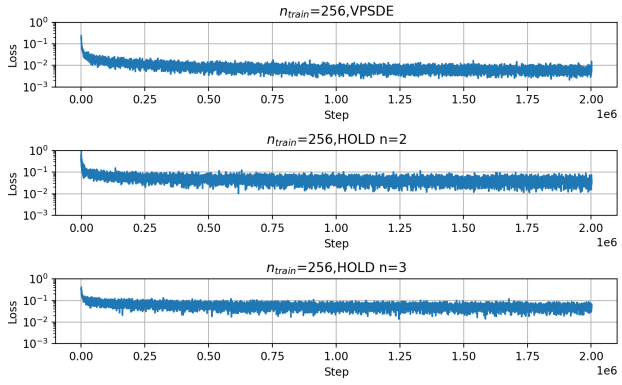


Figure 10: Training losses for $n_{\text{train}} = 256$ for the VPSDE and HOLD $n = 2, 3$ SDEs.



(a) VPSDE. Fmem: 99.807%, FID: 58.405



(b) HOLD $n = 2$. Fmem: 99.522%, FID: 58.421

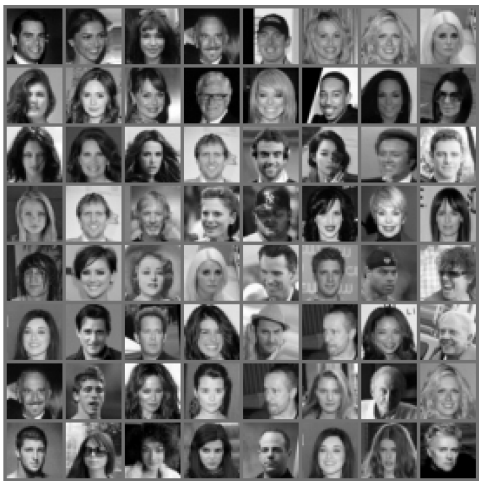


(c) HOLD $n = 3$. Fmem: 90.148%, FID: 53.066



(d) Samples from Training dataset

Figure 11: Celeba comparison for $n_{\text{train}} = 256$ training samples at $1{,}000{,}000$ training iterations.



(a) VPSDE. Fmem: 98.724%, FID: 51.945



(b) HOLD $n = 2$. Fmem: 89.363%, FID: 46.601



(c) HOLD $n = 3$. Fmem: 50.313%, FID: 47.817



(d) Samples from Training dataset

Figure 12: Celeba comparison for $n_{\text{train}} = 512$ training samples at $1{,}000{,}000$ training iterations.



(a) VPSDE. Fmem: 77.915%, FID: 47.012



(b) HOLD $n = 2$. Fmem: 47.024%, FID: 42.328



(c) HOLD $n = 3$. Fmem: 5.962%, FID: 60.380



(d) Samples from Training dataset

Figure 13: Celeba comparison for $n_{\text{train}} = 1024$ training samples at $1{,}000{,}000$ training iterations.



(a) VPSDE. Fmem: 18.579%, FID: 49.168



(b) HOLD $n = 2$. Fmem: 2.845%, FID: 54.236



(c) HOLD $n = 3$. Fmem: 0.000%, FID: 53.560



(d) Samples from Training dataset

Figure 14: Celeba comparison for $n_{\text{train}} = 2048$ training samples at $1{,}000{,}000$ training iterations.