Reducing Diffusion Model Memorization with Higher Order Langevin Dynamics

Benjamin Sterling
benjamin.sterling@stonybrook.edu
Stony Brook University

Mónica F. Bugallo
monica.bugallo@stonybrook.edu
Stony Brook University

Tom Tirer
tirer.tom@biu.ac.il
Bar-Ilan University

Abstract

Diffusion/score-based models have emerged as powerful generative models, capable of generating high-quality samples that mimic the training data distribution. However, it has been observed that they are prone to reproducing training samples—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 they regularize the trajectories of the data variable by implicitly imposing additional dynamical constraints. Our work provides, to our knowledge, the first theoretical characterization of the 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 and the possibility of distribution collapse. Together, our results explain the mitigation of memorization as the model order increases. Finally, we present an empirical study on real-world data that supports our theory and highlights this distinct advantage of HOLD over standard diffusion 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; Abu-Hussein et al., 2022; Garber and Tirer, 2024; 2025). Recently, however, 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 because of 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 memorization and generalization regimes depending on the dataset size. They argue that in the generalization

Accepted to FoGen 2026: Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning, an ICML 2026 workshop (non-archival).

regime, the learned distributions are largely independent of model architectures and are instead most heavily influenced by the training data. (Gu et al., 2025) empirically studies the relationship between dataset size, 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 one can avoid or mitigate memorization using early stopping within a range of time that depends on the dataset size. 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 besides early stopping.

In parallel to the works analyzing memorization in diffusion models, there have been separate methods 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 showed that it is possible to critically damp HOLD (Sterling and Bugallo, 2025; Sterling et al., 2025b), and that using HOLD helps to defend DDMs against membership inference attacks (Sterling et al., 2025a).

In this paper, we study the effect of higher-order diffusion modeling on memorization. First, we close a gap in the literature regarding the regularization effect of these models. These models were originally proposed based on 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 the regularization effect of HOLD. Specifically, we utilize the Laplace transform to demonstrate the 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 the impossibility of distribution collapse. Altogether, 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. Namely, we measure the generation quality and memorization rate of models of different orders along training for two different datasets: the CelebA dataset (extending the setup of (Bonnaire et al., 2025)) and the CIFAR-10 dataset. 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 works, 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 the 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 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 $\mathbf{s}_{\text{emp}}(\mathbf{x}, t) = \nabla_{\mathbf{x}} \log p_t^{\text{emp, OU}}(\mathbf{x})$, 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,

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

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., 2025b) following the variable conventions used in (Dockhorn et al., 2022). Suppose 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:

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

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 "velocity" variable $\mathbf{v}_t^{(1)}$ and the "acceleration" variable $\mathbf{v}_t^{(2)}$. 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),

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

It is proven in (Sterling et al., 2025b) that $\mathbf{u}_t$ must also possess 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 $\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2 \dots \boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_h)$, $\boldsymbol{\epsilon}_{\text{full}} = \text{vec}(\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2 \dots \boldsymbol{\epsilon}_n)$, 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 posses a single geometric eigenvalue $s_* = \sqrt{2n-3}$, as in (Sterling et al., 2025b), then it may be calculated by a finite Taylor Series, 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 under this forward SDE is the same score matching objective that (Dockhorn et al., 2022; Shi and Liu, 2024) use. Instead of modeling the score of the data variable $\mathbf{x}_t$ or the score of the full $\mathbf{u}_t$, 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 backwards 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 $\mathbf{S}_\theta(\mathbf{u}_t, t) = \text{vec}(\mathbf{0}_{(n-1)h}, \mathbf{s}_\theta(\mathbf{u}_t, t))$. It is further proven in (Sterling et al., 2025b) 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. While neither the reverse SDE nor 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 signals and systems perspective (Oppenheim et al., 1997).

We start with Lemma 3.1 below, which assists the proof of our key Theorem 3.2. Essentially, it is based on taking the Laplace transform of both sides of Equation (6), and identifying recursive patterns.

Lemma 3.1. *Consider the ODE (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 $\tilde{\mathbf{s}}(s)$ (accounting for t from both arguments), and $P(s)$ is the characteristic polynomial of $\mathbf{F}$, then*

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

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

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

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 3.1. 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 is also noteworthy that the score, $\mathbf{s}_\theta(\mathbf{u}_t, t)$ affects $\mathbf{x}_t$ only through convolution with $h_t^{(n)}$.

Theorem 3.2 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 for $\mathbf{x}_t$ 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. It was proven in (Sterling et al., 2025b) that the critically damped parameters are optimal for the following design objective involving the forward matrix $\mathbf{F}$:

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

With Lemma 3.1 and Theorem 3.2, 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 (7), not using the critically damped parameter choices necessitates 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 slower convergence along diffusion time, or from a frequency perspective: a mode that allows more high frequencies from $\mathbf{s}_\theta$ 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 that the solution of the backward ODE associated with the (first-order) Ornstein–Uhlenbeck process: $d\mathbf{x}_t = -\xi\mathbf{x}_t dt + \sqrt{2\xi L^{-1}}d\mathbf{w}_t$ may be expressed with $h_t^{\text{OU}} = -\xi L^{-1}\exp(-\xi t)$:

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

Contrasting this result with Theorem 3.2 sheds a new light on the inherent regularization of HOLD. It formally shows the smoothing effect of HOLD on the generated sample trajectories, where the "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 constraints 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 convolve 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 plugging in $s = i\omega$. 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

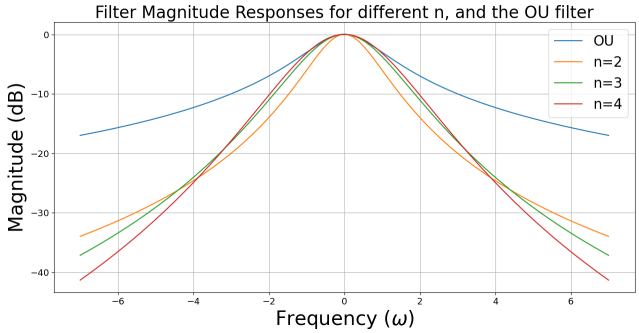


Figure 1. Magnitudes of the Fourier Transforms $|H(i\omega)|$ for different HOLD diffusion model orders n, and the Ornstein–Uhlenbeck filter with $\xi = 1$. One may observe that the HOLD filters are better at attenuating higher frequencies than OU, while still allowing a wide band of lower frequencies.

$h_t^{(n)}$, 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 for a wider range of low frequencies to pass while more sharply attenuating higher frequencies. This sharper attenuation is achieved by the term t^{n-1} in $h_t^{(n)}$, that is associated with a pole in Laplace space of multiplicity n. Clearly, the HOLD filters are stronger low-pass filters than the Ornstein–Uhlenbeck filter.

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 more complicated to learn than the optimal empirical score of a standard (first-order) DDM. Then, under certain approximations, we formally show that HOLD prevents the learned distribution from collapsing to training samples, contrary to the standard Ornstein–Uhlenbeck diffusion process.

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 empirical data distribution. The derivation makes use of a technical approximation that each training data point gets assigned a set of auxiliary variables once at the algorithm's initialization. A justification for this approximation is provided in subsection B.1.

Proposition 3.3. *For the empirical distribution* $p(\mathbf{u}_0) = \frac{1}{n_{train}} \sum_{k=1}^{n_{train}} \delta(\mathbf{u}_0 - \mathbf{u}_0^{(k)})$*, the distribution of* $\mathbf{u}_t$*, denoted by* p_t^{emp}*, obeys*

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

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

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

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 empirical distribution's score is a function of each $\mathbf{u}_0^{(k)}$, which contains initial auxiliary variables $(\mathbf{v}_0^{(1),(k)}, \ldots, \mathbf{v}_0^{(n-1),(k)})$ that are completely independent of the training data $\mathbf{x}_0^{(k)}$. HOLD thereby reduces the ability of the score network to fully memorize the true empirical distribution of the training data itself. It does so by forcing the score function to memorize the joint distribution containing the auxiliary variables instead.

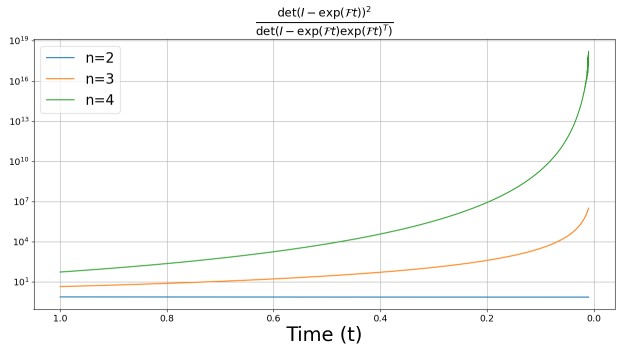


Figure 2. Determinant of the inverse covariance matrix, assuming $L^{-1} = 1$, used to calculate the Mahalanobis distance in Proposition 3.4. As one raises model order, the determinant only increases, thereby improving resistance to distribution collapse.

3.2.2. HOLD PREVENTS DISTRIBUTION COLLAPSE

Here, under certain simplifying assumptions, we formally show another distinction between HOLD and the standard first-order diffusion process. Specifically, we approximate the Mahalanobis distance between the kth training sample and the empirical diffusion distribution as time approaches zero. It is shown below that this distance approaches zero for the Ornstein–Uhlenbeck diffusion process as $t \to 0$, but approaches a finite limit for HOLD $n = 2$, and goes to infinity for HOLD $n = 3$.

Proposition 3.4. *Consider the kth training data sample $\mathbf{x}_t^{(k)}$, that including auxiliary variables becomes $\mathbf{u}_0^{(k)}$. Supposing the training data samples are far enough apart, one may approximate the empirical distributions of the HOLD process for $n = \{2, 3\}$ and the Ornstein–Uhlenbeck process as follows:*

$$p_t^{\text{emp,HOLD}} \approx \mathcal{N}(\boldsymbol{\mu}_t^{\text{HOLD}}, \boldsymbol{\Sigma}_t^{\text{HOLD}}) := \mathcal{N}(\exp(\mathcal{F}t)\mathbf{u}_0^{(k)}, L^{-1}(\mathbf{I} - \exp(\mathcal{F}t)\exp(\mathcal{F}t)^T)),$$
$$p_t^{\text{emp,OU}} \approx \mathcal{N}(\boldsymbol{\mu}_t^{\text{OU}}, \boldsymbol{\Sigma}_t^{\text{OU}}) := \mathcal{N}(\exp(-\xi t)\mathbf{x}_0^{(k)}, L^{-1}(1 - \exp(-2\xi t))\mathbf{I}).$$

Then, the following Mahalanobis distance limits apply:

$$\lim_{t\to 0^+} D_M(\mathbf{x}_0^{(k)}, p_t^{\text{emp,OU}}) = 0, \quad \lim_{t\to 0^+} D_M(\mathbf{u}_0^{(k)}, p_t^{\text{emp,HOLD}}) \gg 0.$$

See Appendix A.4 for the proof. This proposition is proven for the Ornstein–Uhlenbeck process and HOLD model orders $n = 2, 3$ through direct computation of limits. It suggests that for time $t \to 0$ the Ornstein–Uhlenbeck distribution collapses onto the data, but the HOLD distribution does not. Furthermore, Figure 2, numerically computes the determinants of the inverse covariance matrices used in Appendix A.4. This figure suggests that the Mahalanobis distances diverge for n larger than 2. Therefore, HOLD is able to avoid distribution collapse, unlike the Ornstein–Uhlenbeck process.

4. Experiments

In order to validate the theory that HOLD regularizes the training and sample generation processes, we perform experiments in a setup similar to (Bonnaire et al., 2025). We compare 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 on the sample level, as in (Bonnaire et al., 2025). The distances between every generated sample and every training sample are computed. For every generated sample, the gap ratio, the ratio of distances between the first closest and second closest training samples is computed. If it falls below a certain threshold, then that generated sample is declared to be memorized. Each reported memorization percentage is the ratio of memorized generated samples over 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

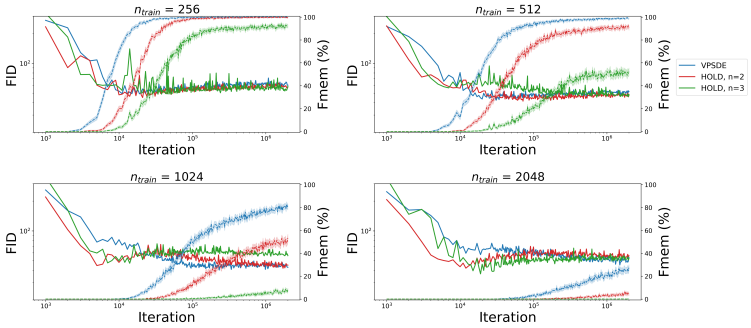


Figure 3. CelebA FIDs and fraction memorized (Fmem) percentages by the number of training samples. Memorization is presented with 95% confidence intervals. Using HOLD and increasing the model order helps to mitigate memorization for roughly the same FID levels.

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. More experimental details appear in Appendix D.

4.1. CelebA Dataset

The main experimental setup in our paper examines the memorization behavior of HOLD throughout training on the CelebA dataset (Liu et al., 2015) for different dataset sizes. To facilitate extensive examination, before training, the images are shrunk from the center to size 32×32, and are converted from RBG 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 inputs required for the auxiliary variables.

Figure 3 presents the FID and Fmem results during training 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 generated data samples in Figure 4 confirm these findings (See Appendix C for more visual results). Eight generated samples are drawn from each model and compared against their nearest neighbors in the training set. A majority the images generated by the VPSDE diffusion model either closely or identically resemble certain training data images. The HOLD $n = 2$ case significantly reduces the memorization percentage 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 fashion. Each diffusion model produces nearly equal FIDs at this number of training iterations, and visual image quality is similar overall.

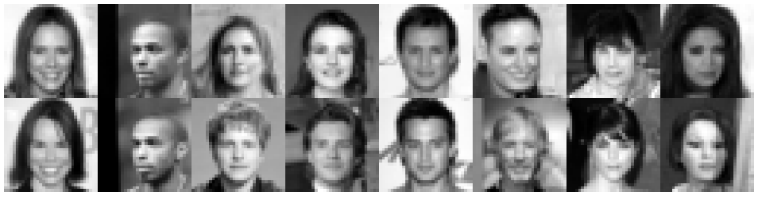
4.2. CIFAR-10 Dataset

We perform similar experiments with the same UNet architecture on the CIFAR-10 dataset (Krizhevsky et al., 2009) (converted to grayscale), and similar conclusions hold here. The key difference between CIFAR-10 and CelebA is that this dataset features ten different image categories, resulting in more dataset diversity and ultimately fewer images to learn from per category. However, this difference does not change the capabilities of HOLD that delay and attenuate memorization. Figure 5 demonstrates this, with memorization decreasing for higher orders, and plateauing to roughly the same FID. These curves do exhibit higher memorization variances and higher FIDs, but this is due to the fact that there are less images per category.

5. Conclusion

While 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

(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 4. 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.

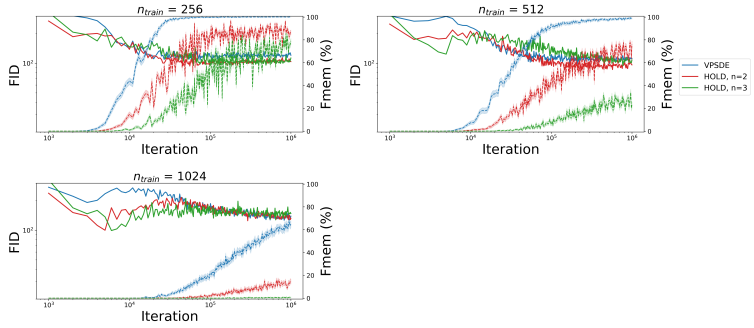


Figure 5. CIFAR-10 FIDs and fraction memorized (Fmem) percentages by the number of training samples. Memorization presented with 95% confidence intervals. Using HOLD and increasing the model order helps to mitigate memorization for similar FID levels.

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 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 do not increase by a factor of n (the model order); in our experiments, 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., 2025b). 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 may also make it harder to detect when users' data has been used for training without their consent.

References

Shady Abu-Hussein, Tom Tirer, and Raja Giryes. Adir: Adaptive diffusion for image reconstruction. *arXiv preprint arXiv:2212.03221*, 2022.

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.

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.

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.

Tomer Garber and Tom Tirer. Zero-shot image restoration using few-step guidance of consistency models (and beyond). In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2398–2407, 2025.

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.

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, Yousef El-Laham, and Mónica F. Bugallo. Defending diffusion models against membership inference attacks via higher-order Langevin dynamics, 2025a. URL https://arxiv.org/abs/2509.14225.

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

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

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 3.3

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

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

Minimizing $\mathcal{L}(s_\theta)$ may be achieved by minimizing $\mathcal{L}(s_\theta, \mathbf{u}_t)$ 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.

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

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

$$\begin{aligned} &\sum_{k=1}^{n_{\text{train}}} \left(\boldsymbol{\epsilon}_n^{(k)} + \mathbf{s}_\theta(\mathbf{u}_t, t)\left(\mathbf{L}_t[nh, nh]\right)\right) p(\boldsymbol{\epsilon}_{\text{full}}^{(k)}) = 0 \\ &\mathbf{s}_\theta(\mathbf{u}_t, t) = -\frac{\sum_{k=1}^{n_{\text{train}}} p(\boldsymbol{\epsilon}_{\text{full}}^{(k)})\boldsymbol{\epsilon}_n^{(k)}/\mathbf{L}_t[nh, nh]}{\sum_{k=1}^{n_{\text{train}}} p(\boldsymbol{\epsilon}_{\text{full}}^{(k)})} = \frac{\sum_{k=1}^{n_{\text{train}}} p(\boldsymbol{\epsilon}_{\text{full}}^{(k)}) \nabla_{\mathbf{v}_t^{(n-1)}} \log p(\mathbf{u}_t \mid \mathbf{u}_0^{(k)})}{\sum_{k=1}^{n_{\text{train}}} p(\boldsymbol{\epsilon}_{\text{full}}^{(k)})} \\ &= \left[\frac{\sum_{k=1}^{n_{\text{train}}} p(\boldsymbol{\epsilon}_{\text{full}}^{(k)}) \nabla_{\mathbf{u}_t} \log p(\mathbf{u}_t \mid \mathbf{u}_0^{(k)})}{\sum_{k=1}^{n_{\text{train}}} p(\boldsymbol{\epsilon}_{\text{full}}^{(k)})}\right]_{n(h-1):nh}. \end{aligned}$$

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

A.2. Proof of Lemma 3.1

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

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

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.

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

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 3.2

Proof. According to Lemma 3.1,

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

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:

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

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 3.4

Proof.

$$\begin{aligned}
p_t^{\text{emp, HOLD}} &\approx \mathcal{N}(\boldsymbol{\mu}_t^{\text{OU}}, \boldsymbol{\Sigma}_t^{\text{HOLD}}) = \mathcal{N}(\exp(\mathcal{F}t)\mathbf{u}_0^{(k)}, L^{-1}(\mathbf{I} - \exp(\mathcal{F}t)\exp(\mathcal{F}t)^T)), \\
p_t^{\text{emp, OU}} &\approx \mathcal{N}(\boldsymbol{\mu}_t^{\text{OU}}, \boldsymbol{\Sigma}_t^{\text{OU}}) = \mathcal{N}(\exp(-\xi t)\mathbf{x}_0^{(k)}, L^{-1}(1-\exp(-2\xi t))\mathbf{I}).
\end{aligned}$$

One may formally calculate that the Mahalanobis distance for the Ornstein–Uhlenbeck diffusion process is zero. Use L'Hôpital's rule to solve the final limit.

$$\begin{aligned}
D_M(\mathbf{x}_0^{(k)}, p_t^{\text{emp, OU}})^2 &= (\mathbf{x}_0^{(k)} - \exp(-\xi t)\mathbf{x}_0^{(k)})^T \frac{\mathbf{I}}{L^{-1}(1-\exp(-2\xi t))} (\mathbf{x}_0^{(k)} - \exp(-\xi t)\mathbf{x}_0^{(k)}) \\
&= \frac{(1-\exp(-\xi t))^2}{L^{-1}(1-\exp(-2\xi t))} ||\mathbf{x}_0^{(k)}||^2, \\
\lim_{t\to 0^+} D_M(\mathbf{x}_0^{(k)}, p_t^{\text{emp, OU}})^2 &= \lim_{t\to 0^+} \frac{(1-\exp(-\xi t))^2}{L^{-1}(1-\exp(-2\xi t))} ||\mathbf{x}_0^{(k)}||^2 = 0.
\end{aligned}$$

However, this limit is different for HOLD:

$$\begin{aligned}
D_M(\mathbf{u}_0^{(k)}, p_t^{\text{emp, HOLD}})^2 &= \frac{1}{L^{-1}} (\mathbf{u}_0^{(k)} - \exp(\mathcal{F}t)\mathbf{u}_0^{(k)})^T \left(\mathbf{I} - \exp(\mathcal{F}t)\exp(\mathcal{F}t)^T\right)^{-1} (\mathbf{u}_0^{(k)} - \exp(\mathcal{F}t)\mathbf{u}_0^{(k)}) \\
&= \frac{1}{L^{-1}} (\mathbf{u}_0^{(k)})^T (\mathbf{I} - \exp(\mathcal{F}t)^T) \left(\mathbf{I} - \exp(\mathcal{F}t)\exp(\mathcal{F}t)^T\right)^{-1} (\mathbf{I} - \exp(\mathcal{F}t))\mathbf{u}_0^{(k)}.
\end{aligned}$$

We are interested in the limit behavior as $t \to 0^+$. To analyze it, take the limit of the determinant of the inverse covariance matrix in the middle. For the purposes of this determinant, use $\mathbf{F}$ instead of $\mathcal{F} = \mathbf{F} \otimes \mathbf{I}_h$. The proof works out identically since for any matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\exp(\mathbf{A} \otimes \mathbf{I}_h) = \exp(\mathbf{A}) \otimes \mathbf{I}_h$, and $\det(\mathbf{A} \otimes \mathbf{I}_h) = \det(\mathbf{A})^h$; this does not change the determinant's behavior as we mainly care whether it goes to zero, remains finite, or diverges to infinity.

$$\begin{aligned}
&\lim_{t\to 0^+} \det\left((\mathbf{I} - \exp(\mathbf{F}t)^T) \left(\mathbf{I} - \exp(\mathbf{F}t)\exp(\mathbf{F}t)^T\right)^{-1} (\mathbf{I} - \exp(\mathbf{F}t)) \right) \\
&= \lim_{t\to 0^+} \frac{\det(\mathbf{I} - \exp(\mathbf{F}t))^2}{\det(\mathbf{I} - \exp(\mathbf{F}t)\exp(\mathbf{F}t)^T)}.
\end{aligned}$$

Firstly, a property of determinants is that it is the product of the eigenvalues of the argument. The eigenvalues of $\mathbf{I} - \exp(\mathbf{F}t)$ are all $1 - \exp(s_* t)$, therefore $\det(\mathbf{I} - \exp(\mathbf{F}t)) = (1 - \exp(s_* t))^n$. Furthermore, as $t \to 0^+$:

$$\begin{aligned}
(1-\exp(s_* t))^n &= (1 - (1 - s_* t + \mathcal{O}(t^2)))^n \\
&= (s_* t + \mathcal{O}(t^2))^n \\
&= (s_* t)^n + \mathcal{O}(t^{n+1}) \\
\det(\mathbf{I} - \exp(\mathbf{F}t))^2 &= (s_* t)^{2n} + \mathcal{O}(t^{2n+1}) \\
&= (2n-3)^n t^{2n} + \mathcal{O}(t^{2n+1}).
\end{aligned}$$

A.4.1. CASE $n = 2$

Specifically for $n = 2$, by Equation (4), $\exp(\mathbf{F}t) = \exp(-t)\,(\mathbf{I} + (\mathbf{F} - s_*\mathbf{I})t)$, and $s_* = -1, \xi = 2$. Therefore

$$\begin{aligned}
\det(\mathbf{I} - \exp(\mathbf{F}t)) &= (1 - \exp(-t))^2 \\
&= (1 - (1 - t + \mathcal{O}(t^2)))^2 \\
&= (t + \mathcal{O}(t^2))^2 \\
&= t^2 + \mathcal{O}(t^3) \\
\det(\mathbf{I} - \exp(\mathbf{F}t))^2 &= t^4 + \mathcal{O}(t^5).
\end{aligned}$$

Now moving onto the denominator.

$$\begin{aligned}
\exp(\mathbf{F}t) &= \exp(-t)(\mathbf{I}(1 + t) + \mathbf{F}t) \\
&= \exp(-t) \begin{pmatrix} 1 + t & -t \\ t & 1 - t \end{pmatrix} \\
\exp(\mathbf{F}t)\exp(\mathbf{F}t)^T &= \exp(-2t) \begin{pmatrix} 1 + t & -t \\ t & 1 - t \end{pmatrix} \\
&\quad \begin{pmatrix} 1 + t & t \\ -t & 1 - t \end{pmatrix} \\
&= \exp(-2t) \\
&\quad \begin{pmatrix} 2t^2 + 2t + 1 & 2t^2 \\ 2t^2 & 2t^2 - 2t + 1 \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
&\mathbf{I} - \exp(\mathbf{F}t)\exp(\mathbf{F}t)^T \\
&\det(\mathbf{I} - \exp(\mathbf{F}t)\exp(\mathbf{F}t)^T) \\
&= (1 - \exp(-2t)(2t^2 + 2t + 1)) \\
&(1 - \exp(-2t)(2t^2 - 2t + 1)) - 4t^4\exp(-4t).
\end{aligned}$$

Take the rest in pieces. $4t^4\exp(-4t) = 4t^4 + \mathcal{O}(t^5)$, and

$$\begin{aligned}
&1 - \exp(-2t)(2t^2 + 2t + 1) \\
&= 1 - (1 - 2t + 2t^2 - \frac{8t^3}{6} + \mathcal{O}(t^4))(2t^2 + 2t + 1) \\
&= 1 - (2t^2 + 2t + 1) + 2t(2t^2 + 2t + 1) \\
&- 2t^2(2t^2 + 2t + 1) + \frac{8t^3}{6} + \mathcal{O}(t^4) \\
&= \frac{4t^3}{3} + \mathcal{O}(t^4) \\
&1 - \exp(-2t)(2t^2 - 2t + 1) \\
&= 1 - (1 - 2t + 2t^2 + \mathcal{O}(t^3))(2t^2 - 2t + 1) \\
&= 1 - \left(2t^2 - 2t + 1 + 4t^2 - 2t + 2t^2 + \mathcal{O}(t^3)\right) \\
&= -2t^2 + 2t - 4t^2 + 2t - 2t^2 + \mathcal{O}(t^3) \\
&= 4t + \mathcal{O}(t^2).
\end{aligned}$$

Finally, one may derive:

$$\begin{aligned}
&\lim_{t\to 0^+} \frac{\det(\mathbf{I} - \exp(\mathbf{F}t))^2}{\det(\mathbf{I} - \exp(\mathbf{F}t)\exp(\mathbf{F}t)^T)} \\
&= \lim_{t\to 0^+} \frac{t^4}{\frac{4t^3}{3}4t - 4t^4} = \frac{3}{4}.
\end{aligned}$$

The inverse covariance matrix is therefore positive definite as $t \to 0^+$, so $\lim_{t\to 0^+} D_M(\mathbf{u}_0^{(k)}, p_t^{\text{emp, HOLD}}) \gg 0$. □

A.4.2. CASE $n = 3$

$$\det(\mathbf{I} - \exp(\mathbf{F}t))^2 = (1 - \exp(-\sqrt{3}t))^6 = (3^{3/2}t^3 + \mathcal{O}(t^4))^2 = 27t^6 + \mathcal{O}(t^7).$$

The following was computed with python's sympy library:

$$\begin{aligned}
&\det(\mathbf{I} - \exp(\mathbf{F}t)\exp(\mathbf{F}t)^T) \\
&= \left((-36t^4 + 24\sqrt{3}t^3 - 36t^2 - 3)\exp(4\sqrt{3}t) + (36t^4 + 24\sqrt{3}t^3 + 36t^2 + 3)\exp(2\sqrt{3}t) + \exp(6\sqrt{3}t) - 1\right)\exp(-6\sqrt{3}t) \\
&= \frac{24\sqrt{3}}{5}t^9 + \mathcal{O}(t^{10})
\end{aligned}$$

$$\lim_{t\to 0^+} \frac{\det(\mathbf{I} - \exp(\mathbf{F}t))^2}{\det(\mathbf{I} - \exp(\mathbf{F}t)\exp(\mathbf{F}t)^T)} = \lim_{t\to 0^+} \frac{27t^6}{\frac{24\sqrt{3}}{5}t^9} \to \infty.$$

The determinant of the inverse covariance matrix grows infinitely large as $t \to 0^+$, so $\lim_{t\to 0^+} D_M(\mathbf{u}_0^{(k)}, p_t^{\text{emp, HOLD}}) \to \infty$.

A.5. Derivation of Ornstein–Uhlenbeck Convolution

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

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

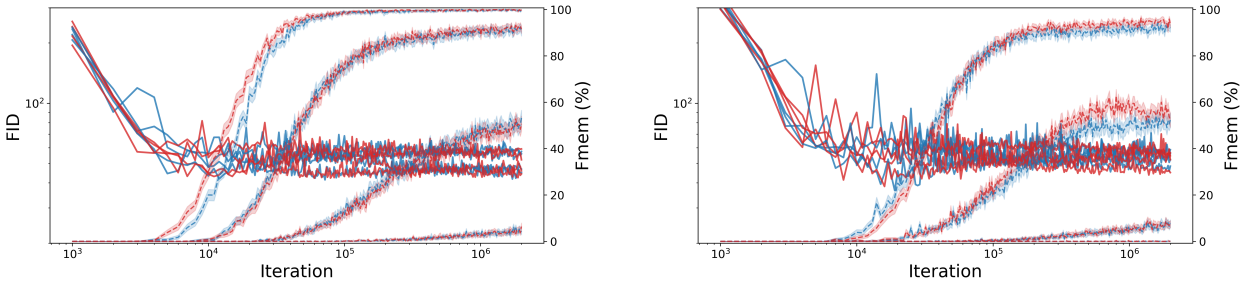
B. Mini-Experiments

B.1. Role of Auxiliary Variable Initialization on Memorization

Figure 6 compares memorization and image quality on the modified CelebA dataset for the case that initial auxiliary variables are drawn once and assigned to each data sample, and for the case that initial auxiliary variables are drawn for each run. The latter method models the joint diffusion process more closely as each initial auxiliary variable is drawn from the true distribution. Assigning each data point an auxiliary variable could also promote further memorization; analyzing this possibility is the purpose of this experiment. However, Figure 6 suggests that whether auxiliary variables are initialized once or constantly during training hardly makes a difference. The difference is slightly larger for model order $n = 3$, but still not significant.

B.2. CIFAR-10 Nearest Neighbors

Figure 7, similarly to Figure 4, demonstrates that higher order dynamics are less prone to memorization by plotting generated images on the top rows with each corresponding nearest neighbor on the bottom rows. The non-memorized samples are visually lower quality than those of CelebA because of the CIFAR-10 category issue, and the fact these images were taken after only $1{,}000{,}000$ training iterations; these experiments were shortened due to time constraints.



(a) Initial auxiliary variable comparison for $n = 2$.

(b) Initial auxiliary variable comparison for $n = 3$.

Figure 6. Initial auxiliary variable comparisons for $n = 2, 3$. Memorization is shown with 95% confidence intervals. The red curves represent initial auxiliary variables being assigned to each data sample, whereas the blue curves represent initial auxiliary variables being newly drawn for each training iteration. There are four pairs of memorization and FID that correspond to $n_{\text{train}} = 256, 512, 1024, 2048$.

B.3. Training Losses

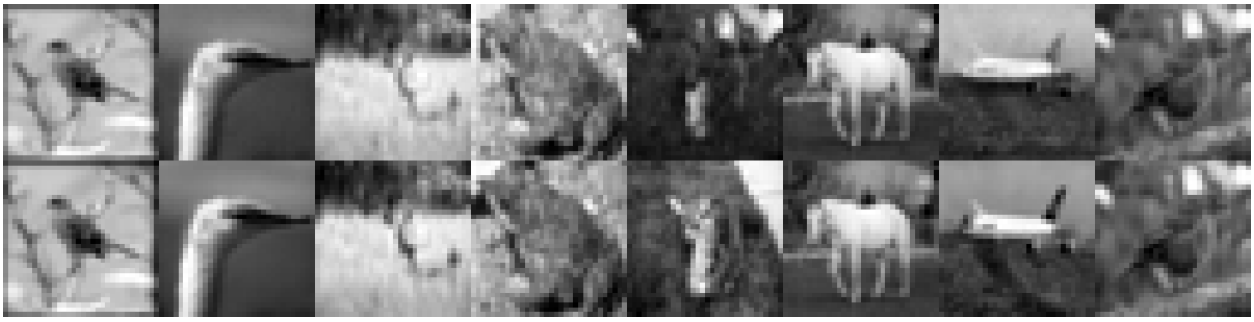
This section contains Figure 8, that presents the training losses for $n_{\text{train}} = 256$ for the VPSDE and HOLD $n = 2, 3$. The slightly higher training losses of HOLD $n = 2$ and $n = 3$ respectively may be explained by the difficulty of learning the optimal empirical score and statistical regularization that is proposed in this work.

C. Additional CelebA images with FIDs and Fmem

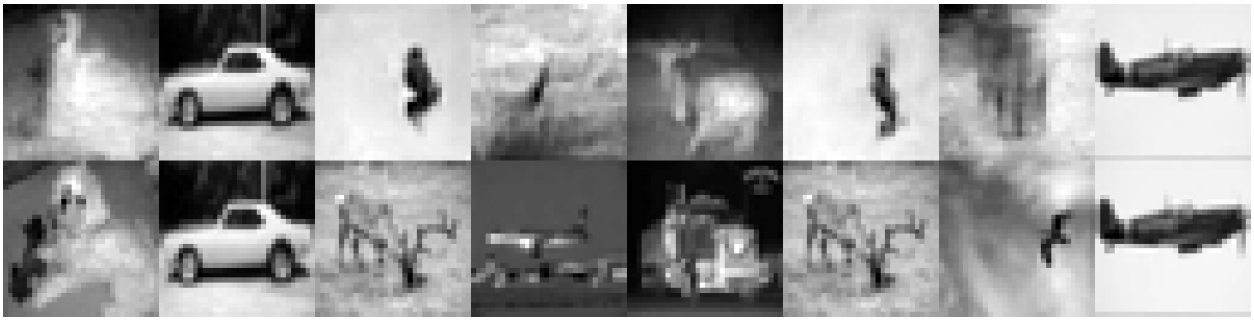
This appendix section presents more generated samples from the CelebA experiment for different number of training images n_{train}.

D. Miscellaneous details

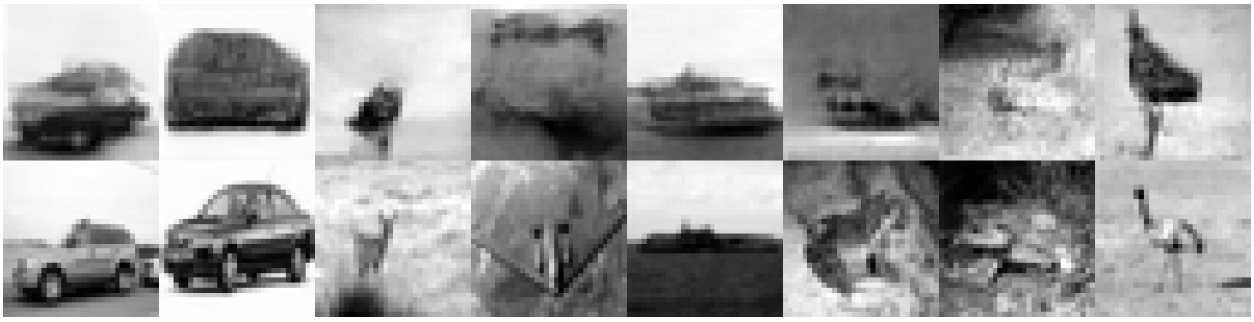
For all experiments, a learning rate of 1×10^{-4}, the Adam optimizer, 32 base channels, a dropout rate of 0.1, and 1000 diffusion model steps were used. For the VPSDE, a linear noise schedule with $\beta_0 = 1 \times 10^{-3}, \beta_1 = 10.0$ was used, and the HOLD runs used $L^{-1} = 1.0$. The UNet (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 3 days of wall clock time, and separate FID and Fmem calculations took negligible amounts of time.



(a) VPSDE. Fmem: 64.856%, FID: 138.768



(b) HOLD $n = 2$. Fmem: 14.410%, FID: 131.407



(c) HOLD $n = 3$. Fmem: 0.598%, FID: 147.492

Figure 7. Nearest training neighbors for different models at $1{,}000{,}000$ training iterations with 1024 training images on the CIFAR-10 dataset. Each first row contains the generated images, and each second row contains the corresponding nearest neighbors.

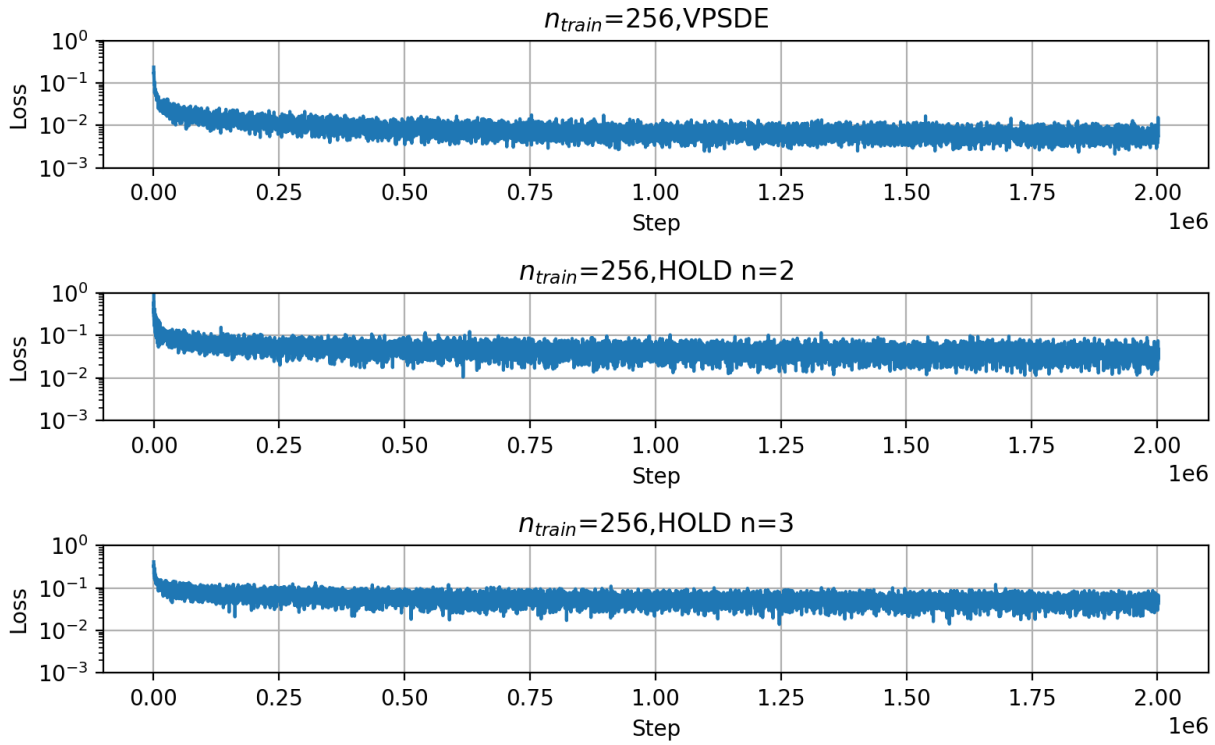


Figure 8. Training losses for $n_{\text{train}} = 256$ for the VPSDE and HOLD $n = 2, 3$ SDEs.

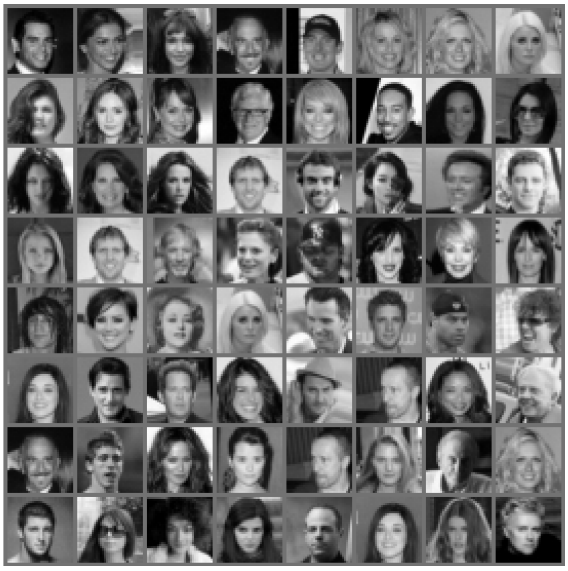
(a) VPSDE. Fmem: 99.807%, FID: 58.405

(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 9. 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

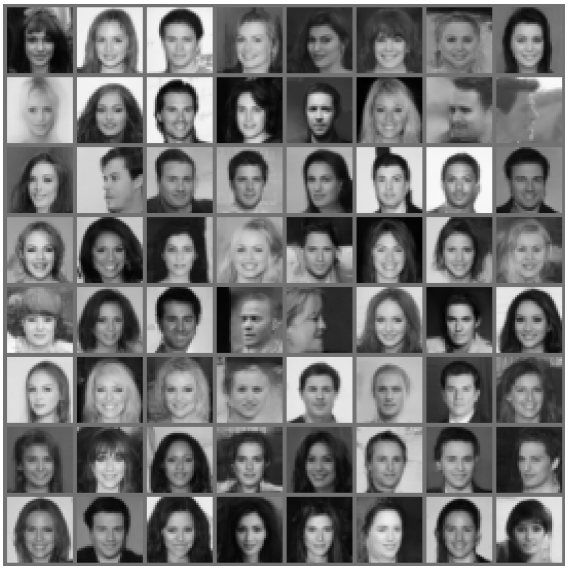
(c) HOLD $n = 3$. Fmem: 50.313%, FID: 47.817

(d) Samples from Training dataset

Figure 10. 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 11. 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 12. Celeba comparison for $n_{\text{train}} = 2048$ training samples at $1{,}000{,}000$ training iterations.