
Beyond Scores: Proximal Diffusion Models

Zhenghan Fang

Mathematical Institute for Data Science
Johns Hopkins University
zfang23@jhu.edu

Mateo Díaz

Mathematical Institute for Data Science
Johns Hopkins University
mateodd@jhu.edu

Sam Buchanan

Toyota Technological Institute at Chicago
sdbuchanan@berkeley.edu

Jeremias Sulam

Mathematical Institute for Data Science
Johns Hopkins University
jsulam1@jhu.edu

Abstract

Diffusion models have quickly become some of the most popular and powerful generative models for high-dimensional data. The key insight that enabled their development was the realization that access to the score—the gradient of the log-density at different noise levels—allows for sampling from data distributions by solving a reverse-time stochastic differential equation (SDE) via forward discretization, and that popular denoisers allow for unbiased estimators of this score. In this paper, we demonstrate that an alternative, backward discretization of these SDEs, using proximal maps in place of the score, leads to theoretical and practical benefits. We leverage recent results in *proximal matching* to learn proximal operators of the log-density and, with them, develop Proximal Diffusion Models (ProxDM). Theoretically, we prove that $\tilde{O}(d/\sqrt{\varepsilon})$ steps suffice for the resulting discretization to generate an ε -accurate distribution w.r.t. the KL divergence. Empirically, we show that two variants of ProxDM achieve significantly faster convergence within just a few sampling steps compared to conventional score-matching methods.

1 Introduction

Within a remarkably brief period, diffusion models [72, 27] have overtaken generative artificial intelligence, emerging as the *de facto* solution across diverse domains including medical imaging [13], video [36] and audio [28], drug discovery [15], and protein docking prediction [12]. Diffusion models combine a forward diffusion process that converts the data distribution we wish to sample into noise and a reverse-time process that reconstructs data from noise. The forward process is easy to simulate by progressively adding noise to data. The reverse process, which allows for sampling data from noise, involves the so-called score function—the gradient of the log-density of the noisy data at any time point. Though unknown in general, such a score can be estimated by common denoising algorithms that approximate minimum mean-squared-error solutions. Various diffusion algorithms have been recently proposed [75, 53, 81], all relying on forward discretizations of the reverse stochastic differential equation, and allowing for impressive generative modeling performance.

Despite these capabilities, current state-of-the-art diffusion-based methods face drawbacks: high-quality samples require a large number of model evaluations, which tends to be computationally expensive, and they are sensitive to hyperparameter choices and the lack of regularity of the data distribution [67, 35]. Naturally, computational cost, and sensitivity to hyperparameters and data regularity are not unique to diffusion models. Many numerical analysis and optimization algorithms grounded on forward discretizations of continuous-time processes share similar limitations. For

instance, when optimizing a smooth convex function f , the go-to algorithmic solution, gradient descent (GD), which updates $x_+ \leftarrow x - h\nabla f(x)$, represents a forward Euler discretization of gradient flow $\dot{x} = -\nabla f(x)$ [1]. GD is known to converge when the learning rate h is on the order of $1/L$ where L is the gradient Lipschitz constant. Consequently, when f approaches nonsmoothness (i.e., L is large), GD must proceed with very small steps, thus converging slowly.

Backward discretization schemes for continuous-time processes offer a potential remedy to these issues. For example, the backward Euler discretization for gradient flow updates iterates via the implicit equation $x_+ = x - h\nabla f(x_+)$. This algorithm is known as the Proximal Point Method, and, remarkably, it converges for any positive h [62]. The tradeoff is that x_+ is defined implicitly, requiring solving an auxiliary sub-problem that is potentially as expensive as the original one. However, in the context of generative models, where f represents the log-density of data distribution, one could learn such an update directly from data via neural networks (instead of learning the score), making the per-step complexity of a backward discretization comparable to that of a forward discretization. This motivates the central question of this work.

Can backward discretization improve diffusion models?

In this paper, we give a partial affirmative answer to this question. We develop a diffusion sampler based on an implicit, backward discretization scheme of the reverse-time stochastic differential equation (SDE). Our approach leverages recent results that allow us to train proximal operators from data via *proximal matching* [23]. We study two variants: one relying exclusively on backward discretization, and another hybrid approach that combines backward and forward terms. Our methods are competitive with score-based diffusion models—including those based on probability flow—when using a large number of discretization steps, while significantly outperforming them with fewer steps. Additionally, we provide supporting convergence theory that yields state-of-the-art rates under the idealized assumption of having access to the exact implicit update or proximal operator. Employing backward discretization of stochastic processes has been studied in both earlier [60, 5, 20, 79, 66, 42, 68, 29] as well as more recent [11, 48, 22, 49, 85, 26, 39, 50, 58, 80] works, mostly in the context of proximal version of Langevin dynamics. To our knowledge, no backward discretization has yet been explored for diffusion models. We defer a more thorough discussion of related works to after the introduction of the necessary background. Code for reproducing the experiments in this work is available at <https://github.com/ZhenganFang/ProxDM>.

2 Background

Diffusion and score-based generative models. Diffusion models perform generative modeling by posing a reversed stochastic differential equation (SDE) of a process that transforms a sample from the data distribution, $X_0 \sim p_0$, $X_0 \in \mathbb{R}^d$, into Gaussian noise [72, 27]. A widely adopted SDE for the forward process is given by

$$dX_t = -\frac{1}{2}\beta(t)X_t dt + \sqrt{\beta(t)}dW_t, \quad t \in [0, T]. \quad (1)$$

where W_t is a standard Wiener process in $\mathbb{R}^d$ (i.e. standard Brownian motion) [75]. At inference time, one can produce a new sample from p_0 by reversing the SDE above, resulting in the stochastic process

$$dX_t = \left[-\frac{1}{2}\beta(t)X_t - \beta(t)\nabla \ln p_t(X_t)\right] dt + \sqrt{\beta(t)}d\bar{W}_t, \quad (2)$$

where $\bar{W}_t$ is a reverse-time Brownian motion (with time flowing backwards from T to 0). This process can be simulated as long as the score function, $s_t(x) = \nabla \ln p_t(x)$, can be computed (or approximated for unknown real-world distributions).

The continuous-time process in (2) must be discretized to be implemented. By far, the most common strategy is applying the forward Euler-Maruyama discretization, leading to the score-based sampling algorithm:

$$X_{k-1} = X_k + \gamma_k \left[\frac{1}{2}X_k + \nabla \ln p_{t_k}(X_k)\right] + \sqrt{\gamma_k}z_k, \quad (\text{Euler-Maruyama})$$

where $\gamma_k = \beta(t_k)(t_k - t_{k-1})$ and $z_k \sim \mathcal{N}(0, I)$. Many variations of these processes have been proposed. For instance, instead of the variance-preserving formulation of the process in (1), one

could instead consider a variance-exploding alternative (by removing the drift) [72, 75]. Likewise, instead of considering stochastic processes, one can find deterministic ordinary differential equations (ODEs) that reverse the forward process and match the marginals of the reversed SDE, resulting in a probability-flow ODE [75, 70].

In order to apply these sampling algorithms for generative modeling on real-world data, corresponding score functions must be obtained for every distribution p_t . The popular *score matching* approach [32, 72, 74, 59] learns parametric regression functions $s_\theta(x, t)$ (with parameters θ) by minimizing the objective

$$\mathbb{E}_{t, X_0, \varepsilon} \left\{ \lambda(t) \left\| s_\theta(X_t, t) - \frac{-\varepsilon}{\sqrt{1 - \alpha_t}} \right\|_2^2 \right\}, \quad (3)$$

where $X_0 \sim p_0$, $\varepsilon \sim \mathcal{N}(0, I_d)$, $\alpha_t = \exp\left(-\int_0^t \beta(s) ds\right)$, $X_t = \sqrt{\alpha_t} X_0 + \sqrt{1 - \alpha_t} \varepsilon$, and $\lambda(t)$ is a function that weighs different terms (for different t) appropriately. It can be shown [75] that the solution to such an optimization problem results in the desired score, $s^*(x, t) = \nabla \ln p_t(x)$. There are two important remarks about this approach: (i) such a function amounts to a *denoiser*—that is, it seeks to recover a signal corrupted by (Gaussian) noise¹—and (ii) such a denoiser is a minimum mean-squared error (MMSE) estimate, as it minimizes the reconstruction error with respect to a squared Euclidean norm. This can also be seen alternatively by Tweedie’s formula [21, 61], which relates the score to the conditional expectation of the unknown, clean sample $\sqrt{\alpha_t} X_0$; namely $\mathbb{E}[\sqrt{\alpha_t} X_0 \mid X_t] = X_t + \sigma^2 \nabla \ln p_t(X_t)$, where $\sigma = \sqrt{1 - \alpha_t}$. Such a conditional mean is the minimizer of the MSE regression problem [57], showing that MMSE denoisers provide direct access to the score of the distribution p_t . Fortunately, the design and development of denoising algorithms is very mature, and strong algorithms exist [57]. As an example, it is standard to employ convolutional networks with U-net architectures [65] for score-matching in image generation [27, 75].

Proximal operators. Proximal operators are ubiquitous in optimization, signal and image processing, and machine learning. For a functional $f: \mathbb{R}^d \rightarrow \mathbb{R}$, the proximal operator of f is a mapping from $\mathbb{R}^d$ to $\mathbb{R}^d$ defined by

$$\text{prox}_f(x) = \arg \min_u f(u) + \frac{1}{2} \|u - x\|_2^2. \quad (4)$$

Intuitively, the proximal operator finds a point that is close to the anchor, x , and has low function value in f . It is convenient to introduce an explicit parameter $\lambda \in \mathbb{R}^+$ to control the relative strength of the functional f , and consider $\text{prox}_{\lambda f}(x)$ instead. In this way, $\text{prox}_{\lambda f}(x) \rightarrow x$ when $\lambda \rightarrow 0$, whereas $\text{prox}_{\lambda f}(x) \rightarrow \arg \min_u f(u)$ when $\lambda \rightarrow \infty$. Further, when f is L -smooth and $\lambda \leq \frac{1}{L}$, first-order optimality conditions yield that $x^+ = \text{prox}_{\lambda f}(x)$ if, and only if, $x^+ = x - \lambda \nabla f(x^+)$, which illustrates its relation to backward discretizations.

Importantly, proximal operators can be equivalently described as *maximum-a-posteriori* (MAP) denoising algorithms under a Gaussian corruption model. Indeed, it is easy to verify² that when $y = x + \varepsilon$, for $x \sim p_0$, Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_d)$, and $f = -\ln p_0$, then $\text{prox}_{\sigma^2 f}(y)$ provides the mode of the conditional distribution of the unknown x given the measurements y ; i.e.,

$$\text{prox}_{\sigma^2 f}(y) = \arg \max_X p_0(X|y). \quad (5)$$

This view of proximal maps in terms of MAP denoising algorithms should be contrasted with that of the score, which relies on MMSE denoising algorithms: unlike the latter, the former does not require the function $f = -\ln p_0$ to be differentiable nor continuous.³ This has been extensively exploited in nonsmooth optimization [3, 14, 18, 19], leading to speedups in convergence. The advantages of employing proximal operators (instead of gradients) for sampling have also been recognized in earlier works [60, 5, 20, 79, 26, 39], leading to faster convergence or less stringent conditions on the parameters of the problem, like step sizes. More generally, MAP estimates are known to provide samples that are closer to the support of the data distribution than MMSE denoisers [16]. In the following section, we will harness proximal maps to develop a new discretization of denoising diffusion models, leveraging MAP denoisers instead of MMSE ones.

¹More precisely, (3) seeks to equivalently estimate the *noise* that has been added to the ground truth X_t .

²(5) follows since $p_0(X|y) \propto p(y|X)p_0(X) \propto \exp(-\|y - X\|^2 / (2\sigma^2)) \exp(-f(X))$ and $\ln$ is monotonic.

³It simply requires f to be weakly-convex or prox-regular [63].

Algorithm 1 Proximal Diffusion Model (ProxDM)

Input: Map $\text{prox}_{-\lambda \ln p_t}(\cdot)$, function $\beta(\cdot)$, grid $\{0 = t_0 < t_1 < \dots < t_N = T\}$, boolean `hybrid`.

Output: Sample X_0 .

Generate $X_N \sim \mathcal{N}(0, I)$

for $k = N$ **to** 1

Generate $z_k \sim \mathcal{N}(0, I)$ and compute $\gamma_k \leftarrow \int_{t_{k-1}}^{t_k} \beta(s) ds$

if `hybrid` **then**

Update $X_{k-1} \leftarrow \text{prox}_{-\gamma_k \ln p_{t_{k-1}}} \left[\left(1 + \frac{1}{2}\gamma_k\right) X_k + \sqrt{\gamma_k} z_k \right]$

else

Update $X_{k-1} \leftarrow \text{prox}_{-\frac{2\gamma_k}{2-\gamma_k} \ln p_{t_{k-1}}} \left[\frac{2}{2-\gamma_k} (X_k + \sqrt{\gamma_k} z_k) \right]$

return X_0

3 Proximal-based generative diffusion modeling

Recall from Section 2 that all score-based generative models apply forward discretization to the reverse-time SDE, resulting in updates of the general form in (Euler-Maruyama), with the score playing a central role. We now propose a different avenue to turn this SDE into a sampling algorithm.

3.1 From backward discretization to proximal-based sampling

Consider now the *backward* discretization of the reversed-time SDE from (2), that is:

$$X_{k-1} = X_k + \gamma_k \left[\frac{1}{2} X_{k-1} + \nabla \ln p_{t_{k-1}}(X_{k-1}) \right] + \sqrt{\gamma_k} z_k. \quad (6)$$

Unlike the forward (Euler-Maruyama) discretization, this is an implicit equation of X_{k-1} . It is easy to verify that this update corresponds to the first-order optimality conditions of the problem $\min_u g(u)$, where $g(u) := -\frac{2\gamma_k}{2-\gamma_k} \ln p_{t_{k-1}}(u) + \frac{1}{2} \left\| u - \frac{2}{2-\gamma_k} (X_k + \sqrt{\gamma_k} z_k) \right\|_2^2$, precisely a proximal step. Such an update can be written succinctly as

$$X_{k-1} = \text{prox}_{-\frac{2\gamma_k}{2-\gamma_k} \ln p_{t_{k-1}}} \left[\frac{2}{2-\gamma_k} (X_k + \sqrt{\gamma_k} z_k) \right]. \quad (\text{PDA})$$

We refer to this step as the proximal diffusion algorithm (PDA). It remains to define the discretization of the step γ_k : recall that for forward discretizations, one can take $\gamma_k = \beta(t_k) \Delta t_k$. Here, we instead make the choice of setting $\gamma_k = \int_{t_{k-1}}^{t_k} \beta(s) ds$. Note that both are equivalent when $\Delta t_k \rightarrow 0$.

It is worth remarking that the discretization in (6) is a *fully* backward discretization: one where both terms depending on the time-continuous $X(t)$ are discretized as X_{k-1} . Yet, one can also consider a *hybrid* approach by considering the updates given by $X_{k-1} = X_k + \gamma_k \left[\frac{1}{2} X_k + \nabla \ln p_{t_{k-1}}(X_{k-1}) \right] + \sqrt{\gamma_k} z_k$, where both X_k and X_{k-1} are employed in the discretization of the drift term. In this case, the resulting update can be concisely written as

$$X_{k-1} = \text{prox}_{-\gamma_k \ln p_{t_{k-1}}} \left[\left(1 + \frac{1}{2}\gamma_k\right) X_k + \sqrt{\gamma_k} z_k \right]. \quad (\text{PDA-hybrid})$$

The resulting hybrid approach is akin to a forward-backward method in nonsmooth optimization [3], is complementary to the update in (PDA), and will have advantages and disadvantages, on which we will expand shortly. For now, we have everything we need to formally define our sampling algorithm for Proximal Diffusion Models (ProxDM), as presented in Algorithm 1.

Remarks. First, we note that this technique for deriving proximal versions of diffusion models via backward discretization is quite general and widely applicable to other variants of diffusion sampling schemes. We have shown this idea for a variance preserving (VP) SDE and the Euler-Maruyama solver, but the same can be done for the reverse diffusion sampler and the ancestral sampler (see [75, Appx. E]), as well as variance exploding (VE) SDE and probability flow ODE [75]. In each case, one would obtain a new proximal-based sampling algorithm that mirrors the diversity of score-based sampling algorithms (see further discussions in Appendix F).

Compare now the update in (PDA) to the score (gradient)-based update in (Euler-Maruyama). First, both algorithms call a denoiser of the data distribution at each step; the score-based method uses an MMSE denoiser based on gradients, whereas PDA uses a MAP denoiser via proximal operators. Second, both updates consist of a denoising step and a noise addition step, but in switched order: the forward version applies denoising first, followed by noise addition. As a result, the final samples from score-based models do contain small amounts of noise, requiring an additional, ad-hoc denoising step at the end to mitigate this and improve sampling performance (see e.g., [75, Appx. G]). In contrast, PDA performs denoising after noise injection. This avoids the need for extra modifications and yields a more principled and general scheme.

Given the benefits of backward discretization, why should one bother with a hybrid approach (PDA-hybrid)? As we will see in the next section, (PDA) enjoys faster convergence rates. However, this comes at the cost of a limitation in the minimal number of sampling steps: one can verify from (PDA) that we must ensure $\gamma_k < 2$ (as otherwise the regularization parameter, $\frac{2\gamma_k}{2-\gamma_k}$, would be negative), resulting in an upper-bound on the step size and thus a lower-bound to the number of sampling steps. The hybrid approach in (PDA-hybrid) does not suffer from this hard constraint, resulting in practical benefits, albeit at the expense of a slower theoretical rate.

The reader should note that the connection between backward discretization and proximal-based algorithms is not new and has been explored in different contexts across optimization and sampling. For example, while forward discretization of the gradient flow leads to the gradient descent algorithm, backward discretization corresponds to the proximal point method [62]. In the context of sampling, forward discretization of the Langevin dynamics yields the (unadjusted) Langevin algorithm (ULA), whereas backward discretization leads to the proximal Langevin algorithm (PLA) [60, 5, 20, 79]. It is interesting to note that, in this context, proximal algorithms that leverage backward discretizations often provide benefits that the simple forward ones do not have: they are often more stable to the choice of hyper-parameters and can do away with requiring differentiability of the potential function [60, 79]. A less related but interesting connection between forward-backward methods (in the space of measures) and sampling has also been studied in recent works [77, 42, 68, 11, 80], and can provide unbiased asymptotic samples (which ULA does not provide) [80]. These advances, however, have all been provided in the context of Langevin dynamics and, to the best of our knowledge, our PDA approach in Algorithm 1 is the first instantiation of these ideas in the context of diffusion processes.

Lastly, score-based methods need the gradient of the log-density at time t_k , $\nabla \ln p_{t_k}$, which can be easily estimated via score-matching. On the other hand, both proximal updates in Algorithm 1 require $\text{prox}_{-\lambda_k \ln p_{t_{k-1}}}$, i.e. the proximal operator for the log-density at time t_{k-1} . Next, we show how to estimate such proximal operators.

3.2 Proximal matching for proximal operator estimation

Learning data-dependent proximal operators has only recently begun to receive attention [30, 56, 47, 23]. We will make use of recent work by Fang et al. [23], who presented a general approach to obtaining learned proximal networks (LPNs). They present a *proximal-matching* loss function that ensures the recovery of the proximal of the log-density⁴ of continuous distributions at arbitrary regularization strengths from i.i.d. samples from that distribution. In a nutshell, proximal-matching augments the random samples $X \sim p_0$ with Gaussian noise, $Y = X + \sigma\epsilon$, for $\epsilon \sim \mathcal{N}(0, I_d)$. Then, it performs a regression using a special loss function and schedule to learn a MAP denoiser for the data distribution, with the noise level dependent on the desired regularization strength λ in the proximal. To learn $\text{prox}_{-\lambda \ln p_0}$, one minimizes the problem $\mathbb{E}_X \mathbb{E}_{Y|X} [\ell_{\text{PM}}(f_\theta(Y), X; \zeta)]$ with the proximal-matching loss

$$\ell_{\text{PM}}(x, y; \zeta) = 1 - \exp\left(-\frac{\|x - y\|_2^2}{d\zeta^2}\right).$$

Fang et al. [23, Theorem 3.2] showed that, in the limit of $\zeta \rightarrow 0$, the problem above⁵ recovers the proximal operator of the log density of the distribution with regularization strength σ^2 ; i.e.

⁴The work in [23] also proposes a specific parametrization of neural networks based on gradients of convex functions (such as those arising from ICNNs [2]) that guarantees exact proximal operators for any obtained network.

⁵We remark that the form of ℓ_{PM} used here differs slightly from the definition in [23], but it does not change the optimization problem nor its guarantees.

$f_{\theta^*} = \text{prox}_{-\sigma^2 \ln p_0}$. Intuitively, the proximal matching loss can be interpreted as a smoothed version of the ℓ_0 pseudo-norm. In one dimension, minimizing the (non-differentiable) ℓ_0 loss recovers the mode of a distribution. The proximal matching loss generalizes this idea to higher dimensions and differentiable functions, making it amenable to first-order optimization methods. Fig. 6 visualizes the proximal matching loss along with the mean-squared error and ℓ_1 losses to illustrate their difference.

Here, we extend proximal-matching to a setting that will enable us to learn a set of proximals for varying densities and regularization strengths, as needed for ProxDM: the set $\{\text{prox}_{-\lambda \ln p_t}\}$ for a range of t and λ 's. Our generalization of the proximal matching objective for proximal diffusion model is

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{t,\lambda} \left\{ \mathbb{E}_{X_t} \mathbb{E}_{Y|X_t} [\ell_{PM}(f_{\theta}(Y; t, \lambda), X_t; \zeta)] \right\} \quad (7)$$

where $X_t \sim p_t$ and $Y|X_t \sim \mathcal{N}(X_t, \lambda I_d)$ are clean and noisy samples for the forward process, respectively. Intuitively, with sufficient data and model capacity (just as in score-matching networks), the optimal solution to (7), denoted by $f_{\theta^*}(x; t, \lambda)$, approximates $\text{prox}_{-\lambda \ln p_t}(x)$ as $\zeta \rightarrow 0$.

3.3 Practical considerations and implementation details

Although this proximal matching framework is conceptually simple, translating it into practice requires careful implementation. Herein, we present a few important methodological details.

Sampling of (t, λ) pair. Optimizing (7) requires sampling (t, λ) pairs, and the sampling distribution should ideally match the distribution to be used after training. As in score-based models, a reasonable choice for the time step t is uniform in $[0, T]$. The values of λ is less obvious, however. Recall that this regularization strength depends on the step size (γ_k) and, thus, also on time t_{k-1} and t_k . We adopt a heuristic for sampling (t, λ) according to the candidate number of steps to be used at inference time (see Appendix E.2 for further details and motivation), which in practice enables the trained model to be applied for a range of different numbers of steps. We note that this sampling scheme is not universally optimal, and we leave the optimization of sampling schemes to future work.

Parameterization of the proximal network $f_{\theta}(x; t, \lambda)$. We parameterize the proximal operator $\text{prox}_{-\lambda \ln p_t}$ using a neural network denoted by $f_{\theta}(\cdot; t, \lambda)$. Notably, while the score network is conditioned on a single time scalar t [27, 75], the proximal network is conditioned on two scalars, t and λ . Similar to score networks, conditioning is implemented by adding learned embeddings for t and λ to the intermediate feature maps. Moreover, the score network indirectly parameterizes the score via $s_{\theta}(x, t) = \frac{-\epsilon_{\theta}(x, t)}{\sqrt{1-\alpha_t}}$ [27], where $s_{\theta}(x, t)$ is the (learned) score and $\epsilon_{\theta}(x, t)$ is a neural network. Analogously, we parameterize the proximal indirectly by letting $f_{\theta}(x; t, \lambda) = x - \sqrt{\lambda} \epsilon_{\theta}(x; t, \lambda)$, where $\epsilon_{\theta}(x; t, \lambda)$ is a neural network that predicts the residual of the proximal operator: $\epsilon_{\theta}(x; t, \lambda) = [x - f_{\theta}(x; t, \lambda)]/\sqrt{\lambda}$. This parameterization led to better empirical results at large step numbers in our experiments and coincides naturally with the balancing of objective across different choices of (t, λ) 's, which we discuss next.

Balancing contributions of different (t, λ) in the objective. Score matching uses a weighing function to balance the loss at different times (see [75, $\lambda(t)$ in Eq. (7)]). For proximal matching, a proper weighing should also be used for balanced learning, and the weighing should be based on both t and λ (c.f. (7)). A general rule of thumb would choose these weights such that the objective of the inner expectation in (7) has similar magnitude across different t and λ . Score model achieves this by weighing according to the square of the magnitude of the target in the objective [75, Sec. 3.3], which is reasonable when the magnitude of the target can be estimated based on t and when the loss function is (2-)homogeneous. Unfortunately, none of these are true for proximal-matching. Thus, instead of weighing the objective, we transform the target to align its magnitude across different (t, λ) 's. Specifically, we rewrite the problem in (7) as follows (but without changing the objective):

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{t,\lambda} \left\{ \mathbb{E}_{X_t} \mathbb{E}_{Y|X_t} \left[\ell_{PM} \left(\frac{Y - f_{\theta}(Y; t, \lambda)}{\sqrt{\lambda}}, \frac{Y - X_t}{\sqrt{\lambda}}; \zeta \right) \right] \right\}, \quad (8)$$

where $\epsilon := \frac{Y - X_t}{\sqrt{\lambda}}$ is the new target. Since $Y \sim \mathcal{N}(X_t, \lambda I)$, we have $\epsilon \sim \mathcal{N}(0, I)$. Notably, the magnitude of ϵ is *independent* of (t, λ) , allowing for a natural balance. Plugging in the parameterization

of $f_\theta(x; t, \lambda) = x - \sqrt{\lambda}\epsilon_\theta(x; t, \lambda)$, the final objective reduces to:

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{t, \lambda} \left\{ \mathbb{E}_{X_t} \mathbb{E}_{\epsilon} \left[\ell_{\text{PM}} \left(\epsilon_\theta \left(X_t + \sqrt{\lambda}\epsilon; t, \lambda \right), \epsilon; \zeta \right) \right] \right\}. \quad (9)$$

Recall that ϵ_θ is a neural network, $f_\theta(x; t, \lambda)$ is the learned approximation of $\text{prox}_{-\lambda \ln p_t}(x)$, (t, λ) are sampled according to a predefined distribution, and ζ is a hyperparameter that should decrease through training. Note that this objective is similar to that of diffusion models with analogous parameterization [27, Eq. (12)]. We include pseudo-code for proximal-matching training in Algorithm 2, noting that it simply boils down to optimizing (9) in a stochastic manner, allowing for any solvers of choice (stochastic gradient descent, Adam, etc).

It is worth comparing the resulting method with score matching. First, score matching samples Gaussian noise only once (to compute X_t). In contrast, our approach samples noise *twice*: once to construct X_t and once for proximal matching training. Second, score matching aims to learn the MMSE denoiser for X_0 (more precisely, $\sqrt{\alpha_t}X_0$) with noise level $\sqrt{1 - \alpha_t}$, whereas proximal matching searches for the MAP denoiser for X_t with noise level $\sqrt{\lambda}$. Lastly, the loss in proximal matching is parameterized by ζ , which decreases via a scheduler, whereas score matching uses a fixed MSE loss throughout.

4 Convergence theory

In this section, we establish a convergence guarantee for Algorithm 1 under the idealized assumption that the proximal operator we use is exact instead of learned. Our results provide a bound on the number of steps needed by the two discretizations we proposed—backward and hybrid—to achieve a KL-divergence of less than ε . We consider the canonical Ornstein–Uhlenbeck (OU) process as the forward process, that is $\beta(t) \equiv 2$ in (1), as is standard in both practical implementations of diffusion models and theoretical analyses of convergence. Further, we consider a uniform time grid $t_k = kT/N$. With this choice, letting $h := T/N$, the step sizes γ_k in Algorithm 1 reduce to $\gamma_k = 2h$.

We now state an informal version of our main result. The formal statement requires additional technical details that we defer to Appendix C. Recall that p_0 is the target distribution.

Theorem 1 (Informal). *Suppose that $\mathbb{E}_{p_0} \|X\|^2$ is bounded above by $M_2 < \infty$. Further, assume for all $t \geq 0$, the potential $\ln p_t$ is a three-times differentiable function with L -Lipschitz gradient and H -Lipschitz Hessian, the moment $\mathbb{E}_{p_t} \|\nabla \ln p_t(X)\|^2$ is bounded above by $C_p < \infty$, and certain regularity conditions hold. Set the step size and time horizon to satisfy $h \leq \frac{1}{8L+4}$ and $T \geq 0.25$. Let q_T be the distribution of the output of Algorithm 1. The following two hold true.*

(Hybrid) *If the `hybrid` flag is set to true, then*

$$\text{KL}(p_0 \| q_T) \lesssim (d + M_2)e^{-T} + hTdL^2 + h^2T[(d + M_2 + C_p)L^2 + d^2H^2] + h^4Td^3L^2H^2.$$

(Backward) *If the `hybrid` flag is set to false and further $h \leq 1$, then*

$$\text{KL}(p_0 \| q_T) \lesssim (d + M_2)e^{-T} + h^2T[(d + M_2 + C_p)L^2 + d^2H^2] + h^4Td^3L^2H^2.$$

A few remarks are in order. First, the uniform bounds on H and C_p can be relaxed to a much weaker bound on an appropriate running average on t —we elaborate on the more general assumptions in the formal statement of the result. Second, the regularity conditions mentioned above—made precise in the appendix—are necessary to guarantee that certain boundary conditions are met. Interestingly, similar conditions seem to be implicitly assumed in many existing results [6, 79, 41, 78], and we defer a longer discussion to the appendix. Third, note that we do not require the distribution to be log-concave, nor do we impose that it satisfies the log-Sobolev inequality. Finally, the proof strategy is inspired by the analysis of the Proximal Langevin algorithm from [79], in combination with techniques from [6, 7, 40, 41, 78]. To better illustrate this result, the following corollary bounds the number of steps necessary to achieve an ε -accurate distribution in KL divergence.

Corollary 1. *Consider the setting of Theorem 1. Moreover, assume that $M_2 \lesssim d, C_p \lesssim dL^2$. The following two hold true.*

(Hybrid) *If the `hybrid` flag is set to true, then $\text{KL}(p_0 \| q_T) \leq \varepsilon$ provided that*

$$T \gtrsim \log \left(\frac{d}{\varepsilon} \right) \quad \text{and} \quad N \gtrsim \frac{T^2 d L^2}{\varepsilon} + \left[\frac{T^3 (d L^4 + d^2 H^2)}{\varepsilon} \right]^{1/2} + \left(\frac{T^5 d^3 L^2 H^2}{\varepsilon} \right)^{1/4}.$$

(**Backward**) If the *hybrid* flag is set to false and further $h \leq 1$, then $\text{KL}(p_0 \| q_T) \leq \varepsilon$ provided that

$$T \gtrsim \log\left(\frac{d}{\varepsilon}\right) \quad \text{and} \quad N \gtrsim \left\lceil \frac{T^3(dL^4 + d^2H^2)}{\varepsilon} \right\rceil^{1/2} + \left(\frac{T^5 d^3 L^2 H^2}{\varepsilon} \right)^{1/4}.$$

The dimension dependence on M_2 and C_p is standard and is, e.g., satisfied by Gaussian distributions. With this result, the benefits of the full backward discretization become clear: the hybrid discretization requires $\tilde{\mathcal{O}}(d\varepsilon^{-1})$ steps, while the backward discretization requires only $\tilde{\mathcal{O}}(d\varepsilon^{-\frac{1}{2}})$ steps. This benefit comes at the expense of the additional condition that $h \leq 1$, since otherwise the regularization coefficient of the proximal operator would become negative. This translates into a hard lower bound on the number of steps one can take, as we discussed above and will see shortly again in the experiments section. In comparison, the most competitive rates for score-based samplers in terms of convergence in KL divergence are $\tilde{\mathcal{O}}(d\varepsilon^{-1})$ for a vanilla method [6, 4] and $\tilde{\mathcal{O}}(d^{\frac{3}{4}}\varepsilon^{-\frac{1}{2}})$ for an accelerated strategy [81]. Note that these rates are not directly comparable, as the underlying assumptions and algorithmic setups (e.g., smoothness, boundedness, or early stopping) differ across works. In particular, [6] assumed L -Lipschitzness for the score function, similar to our work but without the additional smoothness assumption on the Hessian as used here. On the other hand, [4] lifted the smoothness assumptions entirely but instead analyzed the KL with respect to a slightly noise-perturbed version of the target distribution. We point interested readers to the broader body of work on the convergence theory of diffusion models, including both SDE-based samplers [7, 40, 41, 6, 4, 44, 45, 43] and ODE-based samplers [9, 8, 46]; see [44, Section 4] and [43] for a comprehensive review.

5 Experiments

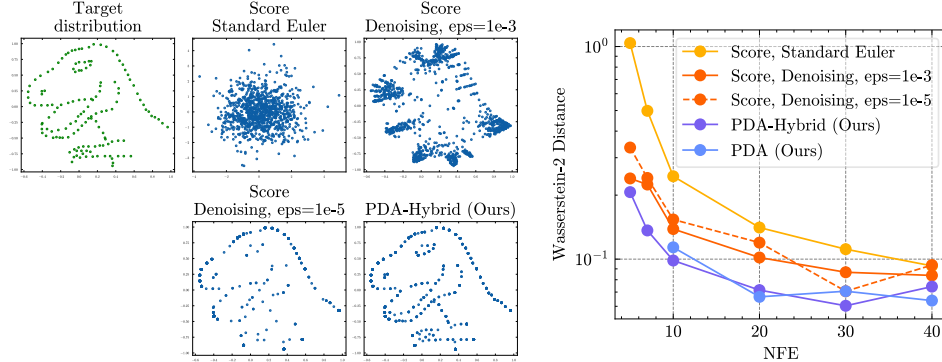


Figure 1: Sampling from a distribution supported on a discrete set of points in 2D using exact score and proximal operators. LEFT: the target distribution and samples generated by various samplers using 5 sampling steps; RIGHT: Wasserstein-2 distance between sample and target across varying step numbers (NFE, number of function evaluations). Standard Euler-Maruyama without varying steps fails when using only 5 steps. Score-based sampler with a denoising step at the end, as is common [75], requires tuning ϵ : the step size for the last denoising step. Both PDA variants perform well without additional hyperparameters.

Synthetic example. We begin with an example in cases where the scores and proximals can be computed to arbitrary precision. We set p_0 as a uniform distribution over a discrete set of points (dataset from [55]), and run score-based diffusion sampling, as well as (PDA) and (PDA-hybrid) (see Algorithm 1). We include both the standard Euler-Maruyama method (“Standard Euler”), as well as the popular modification that includes an ad-hoc final denoising step after the last iterate [75]. As shown by the results in Fig. 1, both variants of ProxDM provide faster convergence with a closer final sample to the ground truth p_0 , and require less hyperparameters to tune and no ad hoc modification. To further study the impact of the smoothness of the distribution on sampling performance, we perform an experiment to sample from 1D distributions with different levels of smoothness at the boundary of their support (see details and results in Appendix H.1). Although the relation between sample quality and smoothness of the distribution is relatively subtle—likely due to smoothing effect inhere in the forward diffusion process—it is clear that ProxDM recovers the support of the distribution more accurately than score-based samplers.

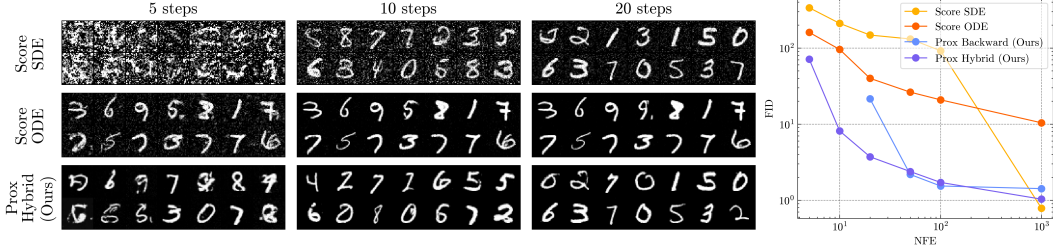


Figure 2: Left: MNIST samples generated by the score SDE sampler, score ODE sampler, and our hybrid ProxDM. Right: the resulting FID score as a function of the number of sampling steps.

MNIST. We now move to a real, simple case for digits [17], where we must train score-networks (through score matching) as well as our learned proximal networks for ProxDM. The $\beta(t)$ in forward process is set as linear, $\beta(t) = \beta_{\min} + (\beta_{\max} - \beta_{\min})t$, for $t \in [0, 1]$, with $\beta_{\min} = 0.1, \beta_{\max} = 20$, as is standard [75]. ProxDM is pre-trained with an ℓ_1 loss, followed by proximal matching loss with $\zeta = 1$ in the first half and decayed to 0.5 in the second half. Both networks are trained for the same number of epochs (see more details in Appendix E.3).

We compare with two score-based sampler: Euler-Maruyama for the reverse-time SDE (denoted as “SDE”) and the Euler sampler for the probability flow ODE (resp. “ODE”) [75]. We show the FID score by NFE (number of function evaluations, i.e. sampling steps) and uncured samples in Fig. 2 (right and left, respectively). ProxDM provides significant speedup compared to both the score samplers based on the reverse-time SDE and probability-flow ODE.

CIFAR10. We perform a similar experiment on the CIFAR10 dataset [37]. The score model uses the same U-Net as in [27]. The network for ProxDM is the same as the score network, except that it has two parallel conditioning branches for t and λ respectively, with a comparable total number of parameters. Score matching follows the standard setup in [27] with learning rate warm-up, gradient clipping, and uniform t sampling as introduced in [75]. For proximal matching, the proximal network is pretrained with ℓ_1 loss, followed by proximal matching loss with $\zeta = 2$ in the first half and $\zeta = 1$ in the second half (see more details in Appendix E.3). Fig. 3 shows the FID score vs. NFE curves. It is worth noting that the score SDE sampler’s FID at 1000 steps matches the value reported in the original paper [27]. It is clear that our ProxDM outperforms score-based samplers and enables faster sampling. Notably, the FID of ProxDM at 10 steps matches that of the score ODE sampler at 20 steps and surpasses the score SDE sampler even at 100 steps. Uncured samples are shown in Fig. 4, demonstrating sharper and cleaner samples by ProxDM with fewer sampling steps. Furthermore, when trained specifically for small step numbers (5, 10 and 20), the advantage of ProxDM over score-based samplers becomes more significant (see Fig. 3, dashed lines). If one is specifically interested in a very small number of steps, disregarding some of the heuristics described in Section 3.3 can lead to even better results (see Fig. 9). Additionally, we report the precision and recall metrics [38], which measure the quality of generated samples and coverage of the real data distribution, respectively (see detailed definitions in Appendix E.3). The results are presented in Fig. 10, further illustrating the improved sample quality and coverage achieved by ProxDM. Finally, the FID of ProxDM at 1000 steps appears worse than that at 100 steps—we discuss on this in Appendix A.3.

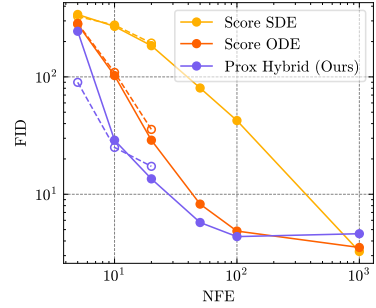


Figure 3: FID vs. number of sampling steps (NFE) on CIFAR10. The dashed lines correspond to models trained specifically for 5, 10 and 20 steps, as opposed to the full-range models in solid lines.

CelebA-HQ (256 × 256). We further evaluate our approach for high-resolution image synthesis using the CelebA-HQ dataset at 256 × 256 resolution [34]. Because of the higher dimensionality of this data, both the score and proximal approaches operate in the latent space of a pretrained VAE, following the standard latent-diffusion setup [64]. As shown in Fig. 11, the hybrid ProxDM achieves improved FID compared to the score SDE sampler and performs competitively with the score ODE



Figure 4: CIFAR10 samples from score SDE, score ODE, and hybrid ProxDM samplers.

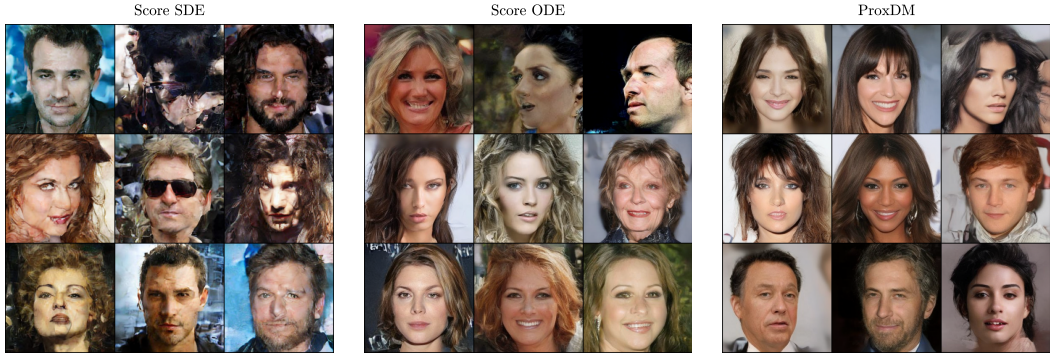


Figure 5: CelebA-HQ (256×256) samples from score SDE, score ODE, and hybrid ProxDM samplers using 20 sampling steps.

sampler across all step counts. Uncurated visual samples using 20 sampling steps are provided in Fig. 5; samples at other step numbers can be found in Fig. 12.

6 Conclusions and limitations

Diffusion models represent a paradigm shift in generative modeling. Since a reversed-time SDE drives sampling, discretization strategies are needed to turn these into practical algorithms. To our knowledge, all diffusion models have so far relied on forward discretizations of these SDEs, relying on the distribution score at each step. Here, we have shown that backward discretizations can be greatly beneficial, both in theory and practice. For the former, our full backward ProxDM requires only $\tilde{O}(d/\sqrt{\epsilon})$ steps to produce a sampling distribution ϵ -away from the target in KL divergence (assuming perfect proximal steps). In practice, we show that neural networks trained with proximal matching allow for practical methods that generate accurate samples far faster than alternative score-based algorithms, and even faster than score-based ODE samplers.

Our work has several limitations. First, our theoretical analysis assumes access to an oracle proximal operator. While we demonstrated that such models can be approximately obtained in practice, incorporating this discrepancy (say, at an α -accuracy level) as done in other works [7, 6, 31] remains open. Second, our convergence results also require additional regularity conditions. While these are arguably mild (see discussion in the appendix), we conjecture that these could be removed with a more careful analysis. On the practical side, we provided a first instantiation of learned proximal networks for ProxDM, but many open questions remain, including how to choose the optimal weighing strategy for different sampling points across training, optimal ways to schedule the ζ hyper-parameter in the proximal-matching loss, and employing networks that provide exact proximal operators for any sampling time.

Acknowledgment

ZF acknowledges support from the Johns Hopkins University Mathematical Institute for Data Science (MINDS) Fellowship. MD was partially supported by NSF awards CCF 2442615. MD and JS were partially supported by DMS 2502377. ZF and JS have been partially supported by NIH Grant P41EB031771.

References

- [1] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savare. *Gradient flows: In metric spaces and in the space of probability measures*. Lectures in Mathematics. ETH Zürich. Birkhauser Verlag AG, Basel, Switzerland, 2 edition, December 2008.
- [2] Brandon Amos, Lei Xu, and J. Zico Kolter. Input convex neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 146–155, 2017.
- [3] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.
- [4] Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d -linear convergence bounds for diffusion models via stochastic localization. *arXiv [stat.ML]*, August 2023.
- [5] Espen Bernton. Langevin monte carlo and JKO splitting. *arXiv [stat.CO]*, February 2018.
- [6] Hongrui Chen, Holden Lee, and Jianfeng Lu. Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. *arXiv [cs.LG]*, November 2022.
- [7] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv [cs.LG]*, September 2022.
- [8] Sitan Chen, Sinho Chewi, Holden Lee, Yuanzhi Li, Jianfeng Lu, and Adil Salim. The probability flow ode is provably fast. *Advances in Neural Information Processing Systems*, 36:68552–68575, 2023.
- [9] Sitan Chen, Giannis Daras, and Alex Dimakis. Restoration-degradation beyond linear diffusions: A non-asymptotic analysis for ddim-type samplers. In *International Conference on Machine Learning*, pages 4462–4484. PMLR, 2023.
- [10] Tianyu Chen, Yasi Zhang, Zhendong Wang, Ying Nian Wu, Oscar Leong, and Mingyuan Zhou. Denoising score distillation: From noisy diffusion pretraining to one-step high-quality generation, 2025. URL <https://arxiv.org/abs/2503.07578>.
- [11] Yongxin Chen, Sinho Chewi, Adil Salim, and Andre Wibisono. Improved analysis for a proximal algorithm for sampling. In *Conference on Learning Theory*, pages 2984–3014. PMLR, 2022.
- [12] Lee-Shin Chu, Sudeep Sarma, and Jeffrey J Gray. Unified sampling and ranking for protein docking with DFMDock. *bioRxiv.org*, page 2024.09.27.615401, September 2024.
- [13] Hyungjin Chung and Jong Chul Ye. Score-based diffusion models for accelerated MRI. *Med. Image Anal.*, 80:102479, August 2022.
- [14] Patrick L Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212, 2011.
- [15] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. DiffDock: Diffusion steps, twists, and turns for molecular docking. *arXiv [q-bio.BM]*, October 2022.

- [16] Mauricio Delbracio and Peyman Milanfar. Inversion by direct iteration: An alternative to denoising diffusion for image restoration. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL <https://openreview.net/forum?id=VmyFF51L3F>. Featured Certification, Outstanding Certification.
- [17] Li Deng. The MNIST Database of Handwritten Digits. <http://yann.lecun.com/exdb/mnist/>, 2012.
- [18] Dmitriy Drusvyatskiy and Adrian S Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research*, 43(3):919–948, 2018.
- [19] Dmitriy Drusvyatskiy and Courtney Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. *Mathematical Programming*, 178:503–558, 2019.
- [20] Alain Durmus, Eric Moulines, and Marcelo Pereyra. Efficient bayesian computation by proximal markov chain monte carlo: When langevin meets moreau. *SIAM J. Imaging Sci.*, 11(1):473–506, January 2018.
- [21] B Efron. Tweedie’s formula and selection bias. *J. Am. Stat. Assoc.*, 106(496):1602–1614, December 2011.
- [22] Jiaojiao Fan, Bo Yuan, and Yongxin Chen. Improved dimension dependence of a proximal algorithm for sampling. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 1473–1521. PMLR, 2023.
- [23] Zhenghan Fang, Sam Buchanan, and Jeremias Sulam. What’s in a prior? learned proximal networks for inverse problems. In *International Conference on Learning Representations*, 2024.
- [24] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models, 2025. URL <https://arxiv.org/abs/2410.12557>.
- [25] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling, 2025. URL <https://arxiv.org/abs/2505.13447>.
- [26] Andreas Habring, Martin Holler, and Thomas Pock. Subgradient langevin methods for sampling from nonsmooth potentials. *SIAM Journal on Mathematics of Data Science*, 6(4):897–925, 2024.
- [27] J Ho, A Jain, and P Abbeel. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.*, 2020.
- [28] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Neural Inf Process Syst*, abs/2204.03458:8633–8646, April 2022.
- [29] Liam Hodgkinson, Robert Salomone, and Fred Roosta. Implicit langevin algorithms for sampling from log-concave densities. *Journal of Machine Learning Research*, 22(136):1–30, 2021.
- [30] Samuel Hurault, Arthur Leclaire, and Nicolas Papadakis. Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization. In *International Conference on Machine Learning*, pages 9483–9505. PMLR, 2022.
- [31] Samuel Hurault, Matthieu Terris, Thomas Moreau, and Gabriel Peyré. From score matching to diffusion: A fine-grained error analysis in the gaussian setting. *arXiv preprint arXiv:2503.11615*, 2025.
- [32] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.*, 6(24):695–709, December 2005.
- [33] James M Joyce. Kullback-leibler divergence. In *International Encyclopedia of Statistical Science*, pages 720–722. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

- [34] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=Hk99zCeAb>.
- [35] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. 2022. URL <https://arxiv.org/abs/2206.00364>.
- [36] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. *arXiv [eess.AS]*, September 2020.
- [37] Alex Krizhevsky. CIFAR-10 and CIFAR-100 datasets. <https://www.cs.toronto.edu/~kriz/cifar.html>, 2009.
- [38] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.
- [39] Tim Tsz-Kit Lau, Han Liu, and Thomas Pock. Non-log-concave and nonsmooth sampling via langevin monte carlo. *Advanced Techniques in Optimization for Machine Learning and Imaging*, 61:83, 2024.
- [40] Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence for score-based generative modeling with polynomial complexity. *arXiv [cs.LG]*, June 2022.
- [41] Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence of score-based generative modeling for general data distributions. *arXiv [cs.LG]*, September 2022.
- [42] Yin Tat Lee, Ruoqi Shen, and Kevin Tian. Structured logconcave sampling with a restricted gaussian oracle. *arXiv [cs.DS]*, October 2020.
- [43] Gen Li and Yuling Yan. $O(d/t)$ convergence theory for diffusion probabilistic models under minimal assumptions. *arXiv preprint arXiv:2409.18959*, 2024.
- [44] Gen Li, Yuting Wei, Yuxin Chen, and Yuejie Chi. Towards faster non-asymptotic convergence for diffusion-based generative models. *arXiv [stat.ML]*, June 2023.
- [45] Gen Li, Yu Huang, Timofey Efimov, Yuting Wei, Yuejie Chi, and Yuxin Chen. Accelerating convergence of score-based diffusion models, provably. *arXiv preprint arXiv:2403.03852*, 2024.
- [46] Gen Li, Yuting Wei, Yuejie Chi, and Yuxin Chen. A sharp convergence theory for the probability flow odes of diffusion models. *arXiv preprint arXiv:2408.02320*, 2024.
- [47] Lingxiao Li, Noam Aigerman, Vladimir G Kim, Jiajin Li, Kristjan Greenewald, Mikhail Yurochkin, and Justin Solomon. Learning proximal operators to discover multiple optima. *arXiv preprint arXiv:2201.11945*, 2022.
- [48] Jiaming Liang and Yongxin Chen. A proximal algorithm for sampling. *arXiv preprint arXiv:2202.13975*, 2022.
- [49] Jiaming Liang and Yongxin Chen. Proximal oracles for optimization and sampling. *arXiv preprint arXiv:2404.02239*, 2024.
- [50] Jiaming Liang, Siddharth Mitra, and Andre Wibisono. Characterizing dependence of samples along the langevin dynamics and algorithms via contraction of ϕ -mutual information. *arXiv preprint arXiv:2402.17067*, 2025. URL <https://arxiv.org/abs/2402.17067v2>.
- [51] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=PqvMRDCJT9t>.
- [52] Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=XVjTT1nw5z>.

- [53] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. *arXiv [cs.LG]*, June 2022.
- [54] Yi-an Ma, Niladri S Chatterji, Xiang Cheng, Nicolas Flammarion, Peter L Bartlett, and Michael I Jordan. Is there an analog of nesterov acceleration for gradient-based mcmc? *Bernoulli*, 27(3): 1942–1992, 2021.
- [55] Justin Matejka and George Fitzmaurice. Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 1290–1294, 2017.
- [56] Tim Meinhardt, Michael Moller, Caner Hazirbas, and Daniel Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1781–1790, 2017.
- [57] Peyman Milanfar and Mauricio Delbracio. Denoising: A powerful building-block for imaging, inverse problems, and machine learning. *arXiv preprint arXiv:2409.06219*, 2024.
- [58] Siddharth Mitra and Andre Wibisono. Fast convergence of ϕ -divergence along the unadjusted langevin algorithm and proximal sampler. In *36th International Conference on Algorithmic Learning Theory*, 2025.
- [59] Tianyu Pang, Kun Xu, Chongxuan Li, Yang Song, Stefano Ermon, and Jun Zhu. Efficient learning of generative models via finite-difference score matching. *arXiv [cs.LG]*, July 2020.
- [60] Marcelo Pereyra. Proximal markov chain monte carlo algorithms. *Statistics and Computing*, 26: 745–760, 2016.
- [61] Herbert Robbins. An empirical bayes approach to statistics. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Vol. I*, pages 157–163. University of California Press, 1956.
- [62] R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.
- [63] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.
- [64] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.
- [65] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [66] Adil Salim, Dmitry Kovalev, and Peter Richtárik. Stochastic proximal langevin algorithm: Potential splitting and nonasymptotic rates. *arXiv [stat.ML]*, May 2019.
- [67] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=TIIdIXIpzhoI>.
- [68] Ruoqi Shen, Kevin Tian, and Yin Tat Lee. Composite logconcave sampling with a restricted gaussian oracle. *arXiv preprint arXiv:2006.05976*, 2020.
- [69] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations*. Association for Computing Machinery, 2015.
- [70] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv [cs.LG]*, October 2020.

- [71] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=WNzy9bRDvG>.
- [72] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *arXiv [cs.LG]*, July 2019.
- [73] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *arXiv [cs.LG]*, June 2020.
- [74] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. *arXiv [cs.LG]*, May 2019.
- [75] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv [cs.LG]*, November 2020.
- [76] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org, 2023.
- [77] Michalis K Titsias and Omiros Papaspiliopoulos. Auxiliary gradient-based sampling algorithms. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80(4):749–767, 2018.
- [78] Santosh S Vempala and Andre Wibisono. Rapid convergence of the unadjusted langevin algorithm: Isoperimetry suffices. *NeurIPS 2019*, March 2019.
- [79] Andre Wibisono. Proximal langevin algorithm: Rapid convergence under isoperimetry. *arXiv preprint arXiv:1911.01469*, 2019.
- [80] Andre Wibisono. Mixing time of the proximal sampler in relative fisher information via strong data processing inequality. *arXiv preprint arXiv:2502.05623*, 2025.
- [81] Yuchen Wu, Yuxin Chen, and Yuting Wei. Stochastic runge-kutta methods: Provable acceleration of diffusion models. *arXiv preprint arXiv:2410.04760*, 2024.
- [82] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv [cs.LG]*, April 2022.
- [83] Yasi Zhang, Tianyu Chen, Zhendong Wang, Ying Nian Wu, Mingyuan Zhou, , and Oscar Leong. Restoration score distillation: From corrupted diffusion pretraining to one-step high-quality generation. *arXiv 2505.13377*, 2025. URL <https://arxiv.org/abs/2505.13377>. (the first two authors contributed equally).
- [84] Mingyuan Zhou, Yi Gu, and Zhendong Wang. Few-step diffusion via score identity distillation. *arXiv 2505.12674*, 2025. URL <https://arxiv.org/abs/2505.12674>.
- [85] Xinkai Zhou, Qiang Heng, Eric C Chi, and Hua Zhou. Proximal mcmc for bayesian inference of constrained and regularized estimation. *The American Statistician*, 78(4):379–390, 2024.
- [86] Bernt Øksendal. *Stochastic differential equations: An introduction with applications*. Universitext. Springer, Berlin, Germany, July 2003.

Contents

1	Introduction	1
2	Background	2
3	Proximal-based generative diffusion modeling	4
3.1	From backward discretization to proximal-based sampling	4
3.2	Proximal matching for proximal operator estimation	5
3.3	Practical considerations and implementation details	6
4	Convergence theory	7
5	Experiments	8
6	Conclusions and limitations	10
A	Discussions	17
A.1	Comparison to Existing Accelerated Methods of Diffusion Models	17
A.2	Connection to Maximum Likelihood Training	17
A.3	Bottleneck of Convergence at Large Step Numbers	18
B	Notation	18
C	Formal statement of the convergence guarantee	19
C.1	Proof of Proposition 1	23
C.2	Proof of Lemma 5	26
D	Proofs of auxiliary lemmata	27
D.1	Proof of Lemma 1 – Interpolation SDE for the Backward Algorithm	27
D.2	Proof of Lemma 2 – Interpolation SDE for the Hybrid Algorithm	28
D.3	Proof of Lemma 3 – Second Moment of OU Process	29
D.4	Proof of Lemma 4 – Bound on Initialization Mismatch	29
D.5	Proof of Lemma 6 – Bounds on the Diffusion Matrix $\widehat{G}_k$ for Backward Algorithm .	30
D.6	Proof of Lemma 9 – Bounds on the Diffusion Matrix $\widehat{G}_k$ for Hybrid Algorithm . .	30
D.7	Proof of Lemma 7 – Bound on the Drift Difference $\mu_k - \widehat{\mu}_k$ for Backward Algorithm	30
D.8	Proof of Lemma 10 – Bound on the Drift Difference $\mu_k - \widehat{\mu}_k$ for Hybrid Algorithm	32
D.9	Proof of Lemma 8 – Bound on the Divergence of Diffusion Matrix $\nabla \cdot \widehat{G}_k$ for Backward Algorithm	32
D.10	Proof of Lemma 11 – Bound on the Divergence of Diffusion Matrix $\nabla \cdot \widehat{G}_k$ for Hybrid Algorithm	34
D.11	Proof of Lemma 13	34
D.12	A General Lemma for Time Derivative of KL	35

E Method and Experimental Details	36
E.1 Pseudo-code for Proximal Matching Training	37
E.2 Sampling t and λ	37
E.3 Image Generation	38
F Extension of ProxDM to Other SDEs and ODEs	38
G Connection between Backward Discretization and Proximal Algorithms	39
H Additional Results	39
H.1 Ablation on Smoothness of Target Distribution in 1D	39
H.2 Additional Results	41

A Discussions

A.1 Comparison to Existing Accelerated Methods of Diffusion Models

Since the introduction of diffusion models, numerous attempts have been made to developing accelerated methods, including those based on deterministic samplers [70, 8], higher-order solvers [53, 82, 45, 81], consistency models [76, 71], distillation methods [67, 10, 84, 83], flow matching [51, 52], and shortcut models [24]. Our aim was not merely to accelerate inference within score-based models, however, but to propose a new and principled framework for diffusion models based on new discretization schemes with convergence guarantees.

In particular, we demonstrate that backward discretization schemes lead to a new class of diffusion models based on proximal operators (instead of scores), and show that this formulation leads to immediate gains in sampling efficiency compared to vanilla score-based SDE samplers. Given its simplicity and generality, the same principle can be naturally extended to existing accelerated methods—for example, by applying backward discretization to probability flow ODEs and flow matching. Our approach thus represents an orthogonal direction to prior advancements, offering a general pathway to improve existing methods. Notably, our approach requires no distillation and is trained directly from data. Moreover, our method enjoys provably favorable convergence properties over score-based counterparts.

Finally, we believe the findings of ProxDM could lead to novel understandings of recent empirical heuristics. For example, recent works have shown the benefits of training generative models using MAP-promoting losses rather than the conventional MSE loss [71, 25]. Our framework employs the proximal matching loss, which also seeks a MAP estimate, but arises naturally from the perspective of learning proximal operators. This connection offers new theoretical insights for understanding the effectiveness of such losses and may inspire future methodological improvements.

A.2 Connection to Maximum Likelihood Training

It has been shown that minimizing the denoising score matching objective is equivalent to maximizing a lower bound on the likelihood of the observed data [27]. In contrast, our proximal matching loss does not naturally admit such a Maximum-likelihood (MLE) interpretation, since it is derived from a continuous-time SDE perspective rather than the discrete Markov chain formulation. It remains an open question whether such an equivalence exists between proximal matching and MLE training. Notably, since our ProxDM differs fundamentally from score-based diffusion models in both the transition kernel (non-Gaussian vs. Gaussian) and training objective (non-MSE vs. MSE), we anticipate that establishing such an equivalence is nontrivial—if at all possible.

Despite this, the samples generated by our method enjoy strong theoretical guarantees. In this work, we bound the KL divergence between the samples generated by our proximal diffusion algorithm and the true data distribution. Combined with existing theoretical results for proximal matching [23], our ProxDM provides provable guarantees for its generated samples.

A.3 Bottleneck of Convergence at Large Step Numbers

Our results showed that the FID score of ProxDM deteriorates at large step numbers (see Fig. 3). We conjecture that this behavior arises because ProxDM, unlike score-based methods, relies on solving an implicit equation that in turn defines a proximal map. In practice, we use proximal matching to train networks to approximate the corresponding proximal operators, rather than implementing exact proximals at each step. As a result, the computed updates may violate the optimality conditions that define such updates, introducing inaccuracies that accumulate. Empirically, we observe that these inaccuracies appear more problematic than the approximation errors of scores in score-based samplers. Better network architectures and training strategies will provide more accurate and stable approximation of the proximal operators, leading to even higher empirical accuracy at high step counts.

B Notation

In this section, we introduce the notation we use and some necessary technical background.

Glossary. Let us summarize the symbols used throughout the appendix:

- h : Discretization step size.
- T : Time horizon of forward process.
- N : Number of discretization steps $N = T/h$.
- p_t : Distribution of the forward process at time t .
- $\bar{p}_t$: Distribution of the reverse-time process at time t , i.e., $\bar{p}_t = p_{T-t}$.
- q_t : Distribution of samples from the algorithm at time t . The density of the samples after k steps is q_{kh} . The density of the final samples is q_T . The density of the initial samples q_0 is a standard Gaussian.
- $\bar{p}_t^{(k)}$: Distribution of the reverse-time process within the k -th step of the algorithm, where $k \in \{0, 1, \dots, N-1\}$ and $t \in [0, h]$. That is $\bar{p}_t^{(k)} = \bar{p}_{kh+t} = p_{T-kh-t}$.
- $q_t^{(k)}$: Distribution of the interpolating process of the sampling algorithm within the k -th step (see Lemmas 1 and 2 and surrounding discussion). Note that $q_t^{(k)} = q_{kh+t}$.

Remarks on terminology. We abuse notation by using the same symbols to denote distributions and their probability density function. We call the distribution we wish to sample from the target or data distribution. We use the word “process” to refer to a stochastic process.

Linear algebra. The symbols $\mathbb{R}^d$, $\mathbb{R}^{d \times d}$, $\mathcal{S}^d$, and $\mathbb{R}^{d \times d \times d}$ denote the set of real d -dimensional vectors, $d \times d$ matrices, and $d \times d$ symmetric matrices, and $d \times d \times d$ tensors. We endow $\mathbb{R}^d$ with the standard inner product $\langle x, y \rangle = x^\top y$, which induces the ℓ_2 norm via $\|x\| = \sqrt{\langle x, x \rangle}$. For a matrix $A \in \mathbb{R}^{d \times d}$ we let $\|A\|_{\text{op}}$ denote the operator norm induced by the ℓ_2 norm and $\|A\|_F$ the Frobenius norm. In particular, if A is symmetric with eigenvalues $\lambda(A) \in \mathbb{R}^d$, then $\|A\|_{\text{op}} = \max_i |\lambda_i|$ and $\|A\|_F = \|\lambda(A)\|$. Thus, $\|A\|_F \leq \sqrt{d} \|A\|_{\text{op}}$. For a symmetric matrix $Q \in \mathcal{S}^d$ with positive eigenvalues (i.e., a positive definite matrix), we write $\|x\|_Q = \sqrt{\langle x, Qx \rangle}$ for the associated norm.

Calculus. Given a differentiable map $\phi: \mathbb{R}^d \rightarrow \mathbb{R}^k$, we use $\nabla \phi(x)$ to denote its gradient or Jacobian at x if $k = 1$ or $k > 1$, respectively. Similarly, given a C^3 function $\varphi: \mathbb{R}^d \rightarrow \mathbb{R}$, we use the symbols $\nabla^2 \varphi(x) \in \mathbb{R}^{d \times d}$ and $\nabla^3 \varphi(x)$ to denote its Hessian and tensor of third-order derivatives at x whose components are given by

$$\nabla^2 \varphi(x)_{ij} = \frac{\partial^2 \varphi(x)}{\partial x_i \partial x_j} \quad \text{and} \quad \nabla^3 \varphi(x)_{ijk} = \frac{\partial^3 \varphi(x)}{\partial x_i \partial x_j \partial x_k},$$

respectively. We denote the Laplacian of φ via

$$\Delta \varphi(x) = \text{Tr}(\nabla^2 \varphi(x)) = \sum_{i=1}^d \frac{\partial^2 \varphi(x)}{\partial x_i^2} \in \mathbb{R}.$$

The symbol $\nabla \cdot$ denotes the divergence operator, which acts on a vector field $v: \mathbb{R}^d \rightarrow \mathbb{R}^d$ via

$$\nabla \cdot v(x) = \sum_{i=1}^d \frac{\partial v_i(x)}{\partial x_i} \in \mathbb{R}.$$

Note that with this notation in place we have the identity $\nabla \cdot (\nabla \varphi) = \Delta \varphi$. For a matrix-valued function $G: \mathbb{R}^d \rightarrow \mathbb{R}^{d \times d}$ and a differentiable function $\phi: \mathbb{R}^d \rightarrow \mathbb{R}$, let $\text{Tr}(\nabla^3 \varphi(x) G(x)) \in \mathbb{R}^d$ denote the vector whose i -th component is $\text{Tr}(\nabla_i \nabla^2 \varphi(x) G(x))$ where

$$\nabla_i \nabla^2 \varphi(x) = \frac{\partial}{\partial x_i} \nabla^2 \varphi(x) = \left(\frac{\partial^3 \varphi(x)}{\partial x_i \partial x_j \partial x_k} \right)_{jk} \in \mathbb{R}^{d \times d}.$$

We use the symbol $\nabla \cdot G$ to denote the divergence operator broadcast to the rows of G , i.e.,

$$\nabla \cdot G(x) := [\nabla \cdot G_1(x), \dots, \nabla \cdot G_d(x)]^\top \in \mathbb{R}^d$$

where $G_i(x) \in \mathbb{R}^d$ is the i -th row of $G(x)$. For brevity, we write G in place of $G(x)$ when the argument is clear from context.

Finally, given a function $\varphi: \mathcal{X} \rightarrow \mathcal{Y}$ between finite dimensional vector spaces and a norm-one vector v we define the directional derivative of φ with respect to v at the point x as

$$\nabla_v \varphi(x) = \lim_{s \downarrow 0} \frac{\varphi(x + sv) - \varphi(x)}{s}.$$

Further, given a canonical basis $\{e_1, \dots, e_{\dim \mathcal{X}}\}$ of $\mathcal{X}$ we let $\nabla_k \varphi$ be a shorthand for $\nabla_{e_k} \varphi$ for $k \in \{1, 2, \dots, \dim \mathcal{X}\}$.

Lipschitzness of the Hessian. For a function $f: \mathbb{R}^d \rightarrow \mathbb{R}$, we define

$$H_f(x) := \max \left\{ \|\nabla_i \nabla^2 f(x)\|_{\text{op}} \mid i = 1, 2, \dots, d \right\} \in \mathbb{R}. \quad (10)$$

The condition $H_f(x) < \infty$ is weaker than a global Lipschitzness for the Hessian.⁶ Indeed, if

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_{\text{op}} \leq H \|x - y\| \quad \text{for all } x, y \in \mathbb{R}^d,$$

then for all $x \in \mathbb{R}^d$,

$$\|\nabla_v \nabla^2 f(x)\|_{\text{op}} \leq H \quad \text{for all } v \in \mathbb{R}^d, \|v\| = 1,$$

which implies

$$\|\nabla_i \nabla^2 f(x)\|_{\text{op}} \leq H \quad \text{for } i = 1, 2, \dots, d.$$

Comparing the above with (10), we can see that $H_f(x)$ is weaker than H in two aspects: (i) the quantity $H_f(x)$ is local, and (ii) it only concerns the directional derivative with respect to canonical basis vectors e_i , not all directions. We use this definition because it suffices for our convergence analysis. We define the shorthand

$$H_t(x) := H_{\ln p_t}(x) \quad \text{for } t \in \mathbb{R}. \quad (11)$$

where p_t is the density along the forward process at time t .

C Formal statement of the convergence guarantee

Before introducing the formal statement, we sketch the proof strategy, which motivates some of the technical assumptions. We present the sketch here regarding the fully backward discretization, but the strategy for the hybrid method is analogous. This proof strategy is based on the analysis of Proximal Langevin Algorithm [79].

⁶[79] assumes that the Hessian of the target potential has a global Lipschitz constant.

We consider the Ornstein–Uhlenbeck (OU) process as the forward process:

$$dX_t = -X_t dt + \sqrt{2} dW_t, \quad X_0 \sim p_0, \quad t \in [0, T], \quad (12)$$

and use p_t to denote the distribution of X_t . For simplicity, we will consider reverse-time processes with time flowing forward. That is, although the original reverse time process evolves via

$$dX_t = [-X_t - 2\nabla \ln p_t(X_t)] dt + \sqrt{2} d\bar{W}_t \quad \text{where } t \text{ flows from } T \text{ to } 0 \quad (13)$$

and $\bar{W}_t$ is the backward Brownian motion, we consider the more intuitive

$$dX_t = [X_t + 2\nabla \ln p_{T-t}(X_t)] dt + \sqrt{2} dW_t \quad (14)$$

where t flows from 0 to T . We use $\tilde{p}_t = p_{T-t}$ for the distribution of the process in (14) to distinguish it from that of the process in (13). We consider the backward discretization $\hat{X}_{kh}$ evolving via the recursion

$$\hat{X}_{(k+1)h} - \hat{X}_{kh} = h\hat{X}_{(k+1)h} + 2h\nabla \ln p_{T-(k+1)h}(\hat{X}_{(k+1)h}) + \sqrt{2h}z_k, \quad (15)$$

here k goes from 0 to $N-1$, $z_k \sim \mathcal{N}(0, I_d)$ and $\hat{X}_0 \sim \mathcal{N}(0, I_d)$. This is equivalent to ProxDM with the hybrid flag set to false (with the only difference that k traverses from 0 to $N-1$ as opposed to the reverse). When the hybrid flag is set to true, the discretization reads

$$\hat{X}_{(k+1)h} - \hat{X}_{kh} = h\hat{X}_{kh} + 2h\nabla \ln p_{T-(k+1)h}(\hat{X}_{(k+1)h}) + \sqrt{2h}z_k. \quad (16)$$

Our goal is to bound the KL divergence $\text{KL}(\tilde{p}_T \| q_T)$ between $\tilde{p}_T$, the target distribution (note that $\tilde{p}_T = p_0$), and q_T , the distribution of $\hat{X}_T$. If $\hat{X}_T$ was defined by a continuous time process with sufficient regularity, then one could apply the fundamental theorem of calculus to get

$$\text{KL}(\tilde{p}_T \| q_T) = \text{KL}(\tilde{p}_0 \| q_0) + \int_0^T \frac{\partial \text{KL}(\tilde{p}_t \| q_t)}{\partial t} dt. \quad (17)$$

In turn, bounding the derivative of the KL divergence as time evolves would yield a bound on the quantity of interest. To execute this plan in our discretized setting, we construct interpolating stochastic processes flowing from $\hat{X}_{kh}$ to $\hat{X}_{(k+1)h}$. Define the stochastic processes $(\hat{X}_t^{(k)})_{t \in [0, h]}$ given by

$$\hat{X}_t^{(k)} = \hat{X}_0^{(k)} + t\hat{X}_t^{(k)} + 2t\nabla \ln p_{T-kh-t}(\hat{X}_t^{(k)}) + \sqrt{2}W_t \quad (18)$$

starting at $\hat{X}_0^{(k)} = \hat{X}_{kh}$ for all k . The next lemma gives an explicit formula for the SDE governing the dynamics of this stochastic process; we defer its proof to Appendix D.1.

Lemma 1 (Interpolating Process for Backward Algorithm). *Suppose that $h \leq \frac{1}{2L+1}$. Then, the discrete algorithm in (15) has a continuous interpolating process evolving according to the SDE*

$$d\hat{X}_t^{(k)} = \hat{\mu}_k(\hat{X}_t^{(k)}, t) dt + \sqrt{2\hat{G}_k(\hat{X}_t^{(k)}, t)} dW_t \quad t \in [0, h] \quad (19)$$

where

$$\begin{aligned} \hat{G}_k(x, t) &= [(1-t)I_d - 2t\nabla^2 \ln p_{T-kh-t}(x)]^{-2}, \quad \text{and} \\ \hat{\mu}_k(x, t) &= \sqrt{\hat{G}_k(x, t)} \left(x + 2\nabla \ln p_{T-kh-t}(x) + 2t \frac{\partial}{\partial t} \nabla \ln p_{T-kh-t}(x) \right. \\ &\quad \left. + 2t \text{Tr} \left[\nabla^3 \ln p_{T-kh-t}(x) \hat{G}_k(x, t) \right] \right). \end{aligned}$$

We shall compare the distributions of these processes with the equivalent pieces of (14), i.e.,

$$dX_t^{(k)} = \underbrace{\left[X_t^{(k)} + 2\nabla \ln p_{T-kh-t}(X_t^{(k)}) \right]}_{:=\mu_k(X_t^{(k)}, t)} dt + \underbrace{\sqrt{2}}_{:=G_k(X_t^{(k)}, t)} dW_t \quad t \in [0, h] \quad (20)$$

To derive a bound on the time derivative of KL divergence in (17), we leverage the Fokker-Planck equation in tandem with a boundary condition that allows us to apply an integral by parts. This motivates the assumptions we impose next.

Assumption 1 (Regularity of the potential). *The following hold.*

(Smoothness) *There exists $L > 1$, such that for all $t \geq 0$, $\ln p_t$ is three-times differentiable and L -smooth, i.e. the gradient $\nabla \ln p_t$ is L -Lipschitz.*

(Data second moment) *The expected energy $M_2 := \mathbb{E}_{p_0}[\|x\|^2] < \infty$ is finite.*

(Score second moment) *The running average $C_p := \frac{1}{T} \int_0^T \mathbb{E}_{p_t}[\|\nabla \ln p_t(x)\|^2] dt < \infty$ is finite.*

(Lipschitz Hessian) *For any x and t , let $H_t(x) = \max \left\{ \|\nabla_i \nabla^2 \ln p_t(x)\|_{\text{op}} \mid i = 1, 2, \dots, d \right\}$. The running average $H^2 := \frac{1}{T} \int_0^T \mathbb{E}_{p_t}[H_t(x)^2] dt < \infty$ is finite.*

We highlight that we do not require the distribution to be log-concave, nor do we impose that it satisfies the log-Sobolev inequality. Smoothness of the potential and bounded second moments are standard assumptions in the literature [6, 81]. Lipschitz continuity of the Hessian is not as common for diffusion models, but has been used to study Langevin-type methods [54, 79].

Additionally, we assume regularity conditions for the analysis of time derivative of KL. We first state it in its general form.

Assumption 2 (Regularity of KL time derivative). *Consider the SDEs*

$$dY_t = f(Y_t, t)dt + \sqrt{2J(Y_t, t)}dW_t$$

and

$$d\hat{Y}_t = \hat{f}(\hat{Y}_t, t)dt + \sqrt{2\hat{J}(\hat{Y}_t, t)}dW_t$$

where $Y_t, \hat{Y}_t \in \mathbb{R}^d$ and $f, \hat{f}: \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$, and $J, \hat{J}: \mathbb{R}^d \times \mathbb{R} \rightarrow \mathcal{S}_+^d$.⁷ Let ρ_t and ν_t be the distributions of Y_t and $\hat{Y}_t$ respectively. The following two conditions hold.

(Boundary condition) *For all t ,*

$$\int_{\mathbb{R}^d} \nabla \cdot \left\{ \ln \frac{\rho_t}{\nu_t} [\nabla \cdot (\rho_t J) - \rho_t f] \right\} dx = 0$$

and $\int_{\mathbb{R}^d} \nabla \cdot \left\{ \frac{\rho_t}{\nu_t} [\nabla \cdot (\nu_t \hat{J}) - \nu_t \hat{f}] \right\} dx = 0.$

Here, we write $\rho_t, \nu_t, f, \hat{f}, J$ and $\hat{J}$ in place of $\rho_t(x), \nu_t(x), f(x, t), \hat{f}(x, t), J(x, t)$ and $\hat{J}(x, t)$, respectively.

(Dominated convergence) *There exist two integrable functions $\theta, \kappa: \mathbb{R}^d \rightarrow \mathbb{R}$ such that*

$$\left| \frac{\partial}{\partial t} \left(\rho_t(x) \ln \frac{\rho_t(x)}{\nu_t(x)} \right) \right| \leq \theta(x) \quad \text{and} \quad \left| \frac{\partial}{\partial t} \rho_t(x) \right| \leq \kappa(x)$$

for all t and almost every x .⁸

To analyze the backward algorithm (ProxDM with hybrid flag set to false), we instantiate Assumption 2 for the reverse-time and interpolating processes.

Assumption 3 (Regularity Condition for Interpolating Process of Backward Algorithm). *For each $k \in \{0, \dots, N-1\}$, let μ_k, G_k , and $X_t^{(k)}$ be defined as in (20) and let $\bar{p}_t^{(k)}$ be the density of $X_t^{(k)}$. Similarly, let $\hat{\mu}_k, \hat{G}_k$, and $\hat{X}_t^{(k)}$ be defined as in (19) and let $q_t^{(k)}$ be the density of $\hat{X}_t^{(k)}$. Assumption 2 holds for $f = \mu_k, \hat{f} = \hat{\mu}_k, J = G_k, \hat{J} = \hat{G}_k, \rho_t = \bar{p}_t^{(k)}$, and $\nu_t = q_t^{(k)}$.*

The boundary condition is required for applying an integral by parts in the analysis of time derivative of KL (see Appendix C.2). Similar conditions are implicitly assumed in proofs in prior works [6, 79, 41, 78]. These conditions are met when the functions inside the divergence operator decay

⁷Recall that $\mathcal{S}_+^d$ denotes the set of $d \times d$ symmetric positive semidefinite matrices.

⁸By integrable we mean $\int \theta(x) dx < \infty$ and $\int \kappa(x) dx < \infty$.

sufficiently fast at infinity, which is satisfied by distributions with regular tail behaviors. The domination condition is required to apply the Leibniz integral rule in the analysis of time derivative of KL (see Lemma 20).

To state an analogous assumption for the hybrid algorithm we require a slightly different interpolating SDE, which we introduce in the next result. The proof of this lemma is deferred to Appendix D.2.

Lemma 2 (Interpolating Process for Hybrid Algorithm). *Suppose that $h < \frac{1}{2L}$. Then, the discrete algorithm in (16) has a continuous interpolating process evolving according to the SDE*

$$d\hat{X}_t^{(k)} = \hat{\mu}_k(\hat{X}_t^{(k)}, t; \hat{X}_0^{(k)}) dt + \sqrt{2\hat{G}_k(\hat{X}_t^{(k)}, t)} dW_t \quad t \in [0, h] \quad (21)$$

where

$$\begin{aligned} \hat{G}_k(x, t) &= [I_d - 2t\nabla^2 \ln p_{T-kh-t}(x)]^{-2}, \quad \text{and} \\ \hat{\mu}_k(x, t; a) &= \sqrt{\hat{G}_k(x, t)} \left(a + 2\nabla \ln p_{T-kh-t}(x) + 2t \frac{\partial}{\partial t} \nabla \ln p_{T-kh-t}(x) \right. \\ &\quad \left. + 2t \operatorname{Tr} [\nabla^3 \ln p_{T-kh-t}(x) \hat{G}_k(x, t)] \right). \end{aligned}$$

With this hybrid interpolating process in place, we introduce analogous regularity conditions.

Assumption 4 (Regularity Condition for Interpolating Process of Hybrid Algorithm). *For each $k \in \{0, \dots, N-1\}$, let μ_k , G_k , and $X_t^{(k)}$ be defined as in (20) and let $\tilde{p}_{t|0}^{(k)}(\cdot | a)$ denote the conditional density of $X_t^{(k)}$ given $X_0^{(k)} = a$. Similarly, let $\hat{\mu}_k$, $\hat{G}_k$, and $\hat{X}_t^{(k)}$ be defined as in (21) and let $q_{t|0}^{(k)}(\cdot | a)$ be the conditional density of $\hat{X}_t^{(k)}$ given $\hat{X}_0^{(k)} = a$. Assumption 2 holds for $f = \mu_k$, $\hat{f}(x, t) = \hat{\mu}_k(x, t; a)$, $J = G_k$, $\hat{J} = \hat{G}_k$, $\rho_t = \tilde{p}_{t|0}^{(k)}(\cdot | a)$ and $\nu_t = q_{t|0}^{(k)}(\cdot | a)$.*

Equipped with these assumptions, we are ready to state our main result.

Theorem 2 (Convergence guarantee). *Suppose that Assumption 1 holds. Set the step size and time horizon to satisfy $h \leq \frac{1}{8L+4}$ and $T \geq 0.25$. The following two hold true.*

(Hybrid) If the hybrid flag is set to true and Assumption 4 holds, then

$$\operatorname{KL}(\tilde{p}_T \| q_T) \lesssim (d + M_2)e^{-T} + hTdL^2 + h^2T[(d + M_2 + C_p)L^2 + d^2H^2] + h^4Td^3L^2H^2. \quad (22)$$

(Backward) If the hybrid flag is set to false and Assumption 3 holds, then

$$\operatorname{KL}(\tilde{p}_T \| q_T) \lesssim (d + M_2)e^{-T} + h^2T[(d + M_2 + C_p)L^2 + d^2H^2] + h^4Td^3L^2H^2. \quad (23)$$

Proof. The proof strategies for both hybrid and backward are completely analogous and follow from the identity (17). The two proofs diverge slightly in how we upper bound the derivative of the KL divergence. The core of the argument appears in the next proposition, whose proof is deferred to Appendix C.1.

Proposition 1. *The following two hold for any $k \in \{0, \dots, N-1\}$.*

(Hybrid) If the hybrid flag is set to true and Assumption 4 holds, then

$$\begin{aligned} \operatorname{KL}(\tilde{p}_h^{(k)} \| q_h^{(k)}) - \operatorname{KL}(\tilde{p}_0^{(k)} \| q_0^{(k)}) &\lesssim h^2dL^2 + h^2L^2 \int_0^h \mathbb{E}_{\tilde{p}_t^{(k)}} [\|x\|^2 + \|\nabla \ln \tilde{p}_t^{(k)}\|^2] dt \\ &\quad + (h^2d^2 + h^4d^3L^2) \int_0^h \mathbb{E}_{\tilde{p}_t^{(k)}} [H_{T-kh-t}(x)^2] dt. \end{aligned} \quad (24)$$

(Backward) If the *hybrid* flag is set to false and Assumption 3 holds, then

$$\begin{aligned} \text{KL}(\tilde{p}_h^{(k)} \| q_h^{(k)}) - \text{KL}(\tilde{p}_0^{(k)} \| q_0^{(k)}) &\lesssim h^2 L^2 \int_0^h \mathbb{E}_{\tilde{p}_t^{(k)}} [\|x\|^2 + \|\nabla \ln \tilde{p}_t^{(k)}\|^2] dt \\ &\quad + (h^2 d^2 + h^4 d^3 L^2) \int_0^h \mathbb{E}_{\tilde{p}_t^{(k)}} [H_{T-kt-t}(x)^2] dt. \end{aligned} \quad (25)$$

The main difference arises from the first term in the upper bound of the hybrid case. From now on, we will continue only with the hybrid scenario, as the rest of the argument is analogous to purely backward discretization. Taking the sum of the inequalities in Proposition 1 from 0 to $N - 1$ yields

$$\begin{aligned} \text{KL}(\tilde{p}_T \| q_T) - \text{KL}(\tilde{p}_0 \| q_0) &= \sum_{k=0}^{N-1} \left(\text{KL}(\tilde{p}_h^{(k)} \| q_h^{(k)}) - \text{KL}(\tilde{p}_0^{(k)} \| q_0^{(k)}) \right) \\ &\lesssim N h^2 d L^2 + h^2 L^2 \int_0^T \mathbb{E}_{\tilde{p}_t} [\|x\|^2 + \|\nabla \ln \tilde{p}_t\|^2] dt \\ &\quad + (h^2 d^2 + h^4 d^3 L^2) \int_0^T \mathbb{E}_{\tilde{p}_t} [H_t(x)^2] dt. \end{aligned} \quad (26)$$

To finish the argument, we invoke the following two results, whose proofs are standard, but we include them for completeness in Appendix D.3 and Appendix D.4, respectively.

Lemma 3. *The second moment of $\tilde{p}_t$ is bounded*

$$\mathbb{E}_{\tilde{p}_t} [\|x\|^2] \leq 2(d + M_2) \quad \text{for all } t \in [0, T].$$

Lemma 4 (Lemma 9 in [6]). *Let q_0 be the standard normal distribution in d dimensions. For $T \geq 0.25$, we have*

$$\text{KL}(\tilde{p}_0 \| q_0) \leq 2(d + M_2)e^{-2T}.$$

Substituting these two into (26), using $N = T/h$ and Assumption 1 and reordering yields

$$\text{KL}(\tilde{p}_T \| q_T) \lesssim (d + M_2)e^{-2T} + hTdL^2 + h^2TL^2(d + M_2 + C_p) + (h^2d^2 + h^4d^3L^2)TH^2.$$

Thus, (22) follows. The argument for (23) is analogous, which completes the proof. $\square$

The rest of this section is devoted to proving Proposition 1.

C.1 Proof of Proposition 1

Once more, the goal is to use the identity (17) by bounding the time derivative of the KL discrepancy. The crux of this proof lies in the following lemma, which we will instantiate twice, once for the hybrid discretization and once for the backward discretization. We defer its proof to the end of this section (Appendix C.2).

Lemma 5 (Time derivative of KL between SDEs). *Suppose Assumption 2 holds.*

Then, for all $t > 0$ we have

$$\begin{aligned} \frac{d}{dt} \text{KL}(\rho_t \| \nu_t) &= -\mathbb{E}_{\rho_t} \left[\left\| \nabla \ln \frac{\rho_t}{\nu_t} \right\|^2 \right] + \mathbb{E}_{\rho_t} \left[\left\langle f - \hat{f}, \nabla \ln \frac{\rho_t}{\nu_t} \right\rangle \right] \\ &\quad + \mathbb{E}_{\rho_t} \left[\left\langle (\hat{J} - J) \nabla \ln \rho_t + \nabla \cdot (\hat{J} - J), \nabla \ln \frac{\rho_t}{\nu_t} \right\rangle \right] \end{aligned} \quad (27)$$

where the arguments for f , $\hat{f}$, J and $\hat{J}$ are omitted.

We consider two cases. We start with the fully backward discretization since the argument is slightly more straightforward.

Case 1: Suppose that hybrid flag is set to false. Fix $k \in \{0, \dots, N-1\}$. Let $\hat{X}_t^{(k)}$, $\hat{\mu}_k$ and $\hat{G}_k$ defined as in Lemma 1. First, we state some properties of $\hat{G}_k$, with their proof deferred to Appendix D.5.

Lemma 6. Let $\hat{G}_k$ defined as in Lemma 1. Then, we have that

$$(1 - 4tL)I \preceq \hat{G}_k \preceq (1 + 18tL)I \quad \text{and} \quad (1 - 2tL)I \preceq \sqrt{\hat{G}_k} \preceq (1 + 6tL)I.$$

We instantiate Lemma 5 with $f = \mu_k$, $\hat{f} = \hat{\mu}_k$, $J = G_k$, and $\hat{J} = \hat{G}_k$. In which case, $\rho_t = \tilde{p}_t^{(k)}$ and $\nu_t = q_t^{(k)}$ and the lemma yields

$$\begin{aligned} & \frac{d}{dt} \text{KL}(\tilde{p}_t^{(k)} \| q_t^{(k)}) \\ &= -\mathbb{E}_{\tilde{p}_t^{(k)}} \left[\left\| \nabla \ln \frac{\tilde{p}_t^{(k)}}{q_t^{(k)}} \right\|_{\hat{G}_k}^2 \right] + \mathbb{E}_{\tilde{p}_t^{(k)}} \left[\left\langle \mu_k - \hat{\mu}_k, \nabla \ln \frac{\tilde{p}_t^{(k)}}{q_t^{(k)}} \right\rangle \right] \\ & \quad + \mathbb{E}_{\tilde{p}_t^{(k)}} \left[\left\langle (\hat{G}_k - I) \nabla \ln \tilde{p}_t^{(k)} + \nabla \cdot (\hat{G}_k - I), \nabla \ln \frac{\tilde{p}_t^{(k)}}{q_t^{(k)}} \right\rangle \right] \\ &\leq -\frac{1}{2} \mathbb{E}_{\tilde{p}_t^{(k)}} \left[\left\| \nabla \ln \frac{\tilde{p}_t^{(k)}}{q_t^{(k)}} \right\|^2 \right] + 2\mathbb{E}_{\tilde{p}_t^{(k)}} [\|\mu - \hat{\mu}\|^2] + 2\mathbb{E}_{\tilde{p}_t^{(k)}} [\|(\hat{G}_k - I) \nabla \ln \tilde{p}_t^{(k)}\|^2] \\ & \quad + 2\mathbb{E}_{\tilde{p}_t^{(k)}} [\|\nabla \cdot (\hat{G}_k - I)\|^2] + \frac{3}{8} \mathbb{E}_{\tilde{p}_t^{(k)}} \left[\left\| \nabla \ln \frac{\tilde{p}_t^{(k)}}{q_t^{(k)}} \right\|^2 \right] \\ &\leq 2\mathbb{E}_{\tilde{p}_t^{(k)}} \left[\|\mu - \hat{\mu}\|^2 + \|(\hat{G}_k - I) \nabla \ln \tilde{p}_t^{(k)}\|^2 + \|\nabla \cdot \hat{G}_k\|^2 \right] \end{aligned} \quad (28)$$

where the second relation uses Cauchy-Schwarz and Young's inequalities to bound $\langle a, b \rangle \leq 2\|a\|^2 + \frac{1}{8}\|b\|^2$ and the fact that $\frac{1}{2}\|\cdot\|^2 \leq \|\cdot\|_{\hat{G}_k}^2$ since $\frac{1}{2}I \preceq \hat{G}_k$ thanks to Lemma 6 and the constraint on $t \leq h \leq \frac{1}{8L}$. The following two lemmas will help us bound the first and last terms in the expected value in (28). Their proofs are deferred to Appendices D.7 and D.9, respectively.

Lemma 7. For all $k \in \{0, \dots, N-1\}$, the difference between μ_k and $\hat{\mu}_k$ is bounded by

$$\|\mu_k - \hat{\mu}_k\|^2 \leq t^2 L^2 \left(363 \|x\|^2 + 2187 \|\nabla \ln \tilde{p}_t^{(k)}(x)\|^2 \right) + 24300 t^4 d^3 L^2 H_{T-kh-t}(x)^2.$$

Lemma 8. For all $k \in \{0, \dots, N-1\}$, the norm of $\nabla \cdot \hat{G}_k$ is bounded by

$$\|\nabla \cdot \hat{G}_k(x)\| \leq 364 t d H_{T-kh-t}(x).$$

Furthermore, Lemma 6 gives $\|\hat{G}_k - I\|_{\text{op}}^2 \leq (18tL)^2 = 324t^2L^2$. Substituting all of these bounds into (28) gives

$$\frac{d}{dt} \text{KL}(\tilde{p}_t^{(k)} \| q_t^{(k)}) \lesssim \mathbb{E}_{\tilde{p}_t^{(k)}} \left[t^2 L^2 (\|x\|^2 + \|\nabla \ln \tilde{p}_t^{(k)}(x)\|^2) + (t^2 d^2 + t^4 d^3 L^2 H_{T-kh-t}(x)^2) \right].$$

Upper bounding $t \leq h$ and integrating from $t = 0$ to $t = h$ yields the desired inequality (25); finishing the proof for the fully backward discretization.

Case 2: Suppose that hybrid flag is set to true. Let $\hat{X}_t^{(k)}$, $\hat{\mu}_k$ and $\hat{G}_k$ defined as in Lemma 2. The proof of this case is slightly different: we use the following chain rule [33, 6], before applying the fundamental theorem of calculus

$$\text{KL}(\tilde{p}_h^{(k)} \| q_h^{(k)}) - \text{KL}(\tilde{p}_0^{(k)} \| q_0^{(k)}) \leq \mathbb{E}_{a \sim \tilde{p}_0^{(k)}} \left[\text{KL}(\tilde{p}_{h|0}^{(k)}(\cdot | a) \| q_{h|0}^{(k)}(\cdot | a)) \right] \quad (29)$$

where $q_{t|0}^{(k)}(\cdot | a)$ denote the conditional density of $\hat{X}_t^{(k)}$ given $\hat{X}_0^{(k)} = a$, and $\tilde{p}_{t|0}^{(k)}(\cdot | a)$ denote the conditional density of $X_t^{(k)}$ in (20) given $X_0^{(k)} = a$. Notice that we can express the KL divergence

on the right-hand side by applying the fundamental theorem of calculus in a similar fashion as in (17). However, in this case the value at $t = 0$ satisfies $\text{KL}(\tilde{p}_{0|0}^{(k)}(\cdot | a) \| q_{0|0}^{(k)}(\cdot | a)) = \text{KL}(\delta_a \| \delta_a) = 0$. Thus, the right-hand side is bounded by $\mathbb{E}_{a \sim \tilde{p}_0^{(k)}} \left[\int_0^h \frac{d}{dt} \text{KL}(\tilde{p}_{t|0}^{(k)}(\cdot | a) \| q_{t|0}^{(k)}(\cdot | a)) dt \right]$. Next, we bound the time derivative. Paralleling the proof in **Case 1**, we introduce relevant properties of $\hat{G}_k$; their proof is deferred to Appendix D.6.

Lemma 9. *Let $\hat{G}_k$ be defined as in Lemma 2. Then, we have that*

$$(1 - 4tL)I \preceq \hat{G}_k \preceq (1 + 12tL)I \quad \text{and} \quad (1 - 2tL)I \preceq \sqrt{\hat{G}_k} \preceq (1 + 4tL)I.$$

We invoke Lemma 5 with $f = \mu_k$, $\hat{f}(x, t) = \hat{\mu}_k(x, t; a)$, $J = G_k$, $\hat{J} = \hat{G}_k$, $\rho_t = \tilde{p}_{t|0}^{(k)}(\cdot | a)$ and $\nu_t = q_{t|0}^{(k)}(\cdot | a)$. The lemma yields

$$\begin{aligned} & \frac{d}{dt} \text{KL}(\tilde{p}_{t|0}^{(k)}(\cdot | a) \| q_{t|0}^{(k)}(\cdot | a)) \\ &= -\mathbb{E}_{\tilde{p}_{t|0}^{(k)}(\cdot | a)} \left[\left\| \nabla \ln \frac{\tilde{p}_{t|0}^{(k)}(\cdot | a)}{q_{t|0}^{(k)}(\cdot | a)} \right\|_{\hat{G}_k}^2 \right] \\ & \quad + \mathbb{E}_{\tilde{p}_{t|0}^{(k)}(\cdot | a)} \left[\left\langle \mu_k - \hat{\mu}_k, \nabla \ln \frac{\tilde{p}_{t|0}^{(k)}(\cdot | a)}{q_{t|0}^{(k)}(\cdot | a)} \right\rangle \right] \\ & \quad + \mathbb{E}_{\tilde{p}_{t|0}^{(k)}(\cdot | a)} \left[\left\langle \left(\hat{G}_k - G_k \right) \nabla \ln \tilde{p}_{t|0}^{(k)}(\cdot | a) + \nabla \cdot \left(\hat{G}_k - G_k \right), \nabla \ln \frac{\tilde{p}_{t|0}^{(k)}(\cdot | a)}{q_{t|0}^{(k)}(\cdot | a)} \right\rangle \right]. \end{aligned}$$

Applying $\frac{1}{2}I \preceq \hat{G}_k$ —thanks to Lemma 9—in tandem with the analogous argument we used for the pure backward method (28) we derive

$$\begin{aligned} & \frac{d}{dt} \text{KL}(\tilde{p}_{t|0}^{(k)}(\cdot | a) \| q_{t|0}^{(k)}(\cdot | a)) \\ & \leq 2\mathbb{E}_{\tilde{p}_{t|0}^{(k)}(\cdot | a)} \left[\|\mu_k - \hat{\mu}_k\|^2 + \left\| (\hat{G}_k - I) \nabla \ln \tilde{p}_{t|0}^{(k)}(\cdot | a) \right\|^2 + \left\| \nabla \cdot \hat{G}_k \right\|^2 \right]. \end{aligned}$$

Integrating over $t \in [0, h]$ and taking expectation over $a \sim \tilde{p}_0^{(k)}$, we have

$$\begin{aligned} & \mathbb{E}_{a \sim \tilde{p}_0^{(k)}} \left[\text{KL}(\tilde{p}_{h|0}^{(k)}(\cdot | a) \| q_{h|0}^{(k)}(\cdot | a)) \right] \\ & \leq 2 \int_0^h \mathbb{E}_{a \sim \tilde{p}_0^{(k)}} \mathbb{E}_{x \sim \tilde{p}_{t|0}^{(k)}(\cdot | a)} \left[\|\mu_k - \hat{\mu}_k\|^2 + \left\| (\hat{G}_k - I) \nabla \ln \tilde{p}_{t|0}^{(k)}(x | a) \right\|^2 + \left\| \nabla \cdot \hat{G}_k \right\|^2 \right] dt. \end{aligned} \tag{30}$$

The following two lemmas will help us bound the first and last terms in the expected value above. Their proofs are deferred to Appendices D.8 and D.10, respectively.

Lemma 10. *For all $k \in \{0, \dots, N-1\}$, the difference between μ_k and $\hat{\mu}_k$ is bounded by*

$$\|\mu_k - \hat{\mu}_k\| \leq tL(8\|x\| + 20\|\nabla \ln p_{T-kh-t}(x)\|) + 48t^2d^{3/2}LH_{T-kh-t}(x) + 2\|a - x\|.$$

Lemma 11. *For all $k \in \{0, \dots, N-1\}$, the norm of $\nabla \cdot \hat{G}_k$ is bounded by*

$$\left\| \nabla \cdot \hat{G}_k(x) \right\| \leq 192tdH_{T-kh-t}(x).$$

Furthermore, Lemma 9 gives that $\|\widehat{G}_k - I\|_{\text{op}} \leq 12tL$. Substituting all of these bounds into (30) gives

$$\begin{aligned}
& \mathbb{E}_{a \sim \tilde{p}_0^{(k)}} \left[\text{KL} \left(\tilde{p}_{h|0}^{(k)}(\cdot | a) \parallel q_{h|0}^{(k)}(\cdot | a) \right) \right] \\
& \lesssim \int_0^h t^2 L^2 \mathbb{E}_{x \sim \tilde{p}_t^{(k)}} [\|x\|^2 + \|\nabla \ln p_{T-kh-t}(x)\|^2] dt \\
& \quad + \int_0^h (t^4 d^3 L^2 + t^2 d^2) \mathbb{E}_{x \sim \tilde{p}_t^{(k)}} [H_{T-kh-t}(x)^2] dt \\
& \quad + \int_0^h \mathbb{E}_{(a,x) \sim (\tilde{p}_0^{(k)}, \tilde{p}_t^{(k)})} [\|a - x\|^2] dt \\
& \quad + \int_0^h t^2 L^2 \mathbb{E}_{(a,x) \sim (\tilde{p}_0^{(k)}, \tilde{p}_t^{(k)})} [\|\nabla \ln \tilde{p}_{t|0}^{(k)}(x | a)\|^2] dt. \tag{31}
\end{aligned}$$

We now establish two lemmas that bound the last two terms on the right hand side above. Lemma 12 is adapted from [6, Lemma 10]. We defer the proof of Lemma 13 to Appendix D.11.

Lemma 12. *For all $k \in \{0, \dots, N-1\}$ we have*

$$\mathbb{E}_{(a,x) \sim (\tilde{p}_0^{(k)}, \tilde{p}_t^{(k)})} [\|a - x\|^2] \leq 4td + 2t^2 \mathbb{E}_{x \sim \tilde{p}_t^{(k)}} [\|x\|^2].$$

Proof of Lemma 12. For ease of notation, we let $\nu_0 = \tilde{p}_0^{(k)}$, $\nu_t = \tilde{p}_t^{(k)}$, and $\nu_{0|t} = \tilde{p}_{0|t}^{(k)}$ within this proof. Applying the triangle inequality and $(a+b)^2 \leq 2a^2 + 2b^2$ yields

$$\mathbb{E}_{(a,x) \sim (\nu_0, \nu_t)} [\|a - x\|^2] \leq 2\mathbb{E}_{x \sim \nu_t} \mathbb{E}_{a \sim \nu_{0|t}(\cdot|x)} [\|a - e^{-t}x\|^2] + 2\mathbb{E}_{x \sim \nu_t} [\|e^{-t}x - x\|^2].$$

Next we bound each term in this sum. Since $\nu_{0|t}(\cdot | x) = \mathcal{N}(e^{-t}x, (1 - e^{-2t})I)$, we have

$$\mathbb{E}_{x \sim \nu_t} \mathbb{E}_{a \sim \nu_{0|t}(\cdot|x)} [\|a - e^{-t}x\|^2] = d(1 - e^{-2t}) \leq 2td.$$

For the second term, factorizing and using the fact that $(1 - e^{-t})^2 \leq t^2$ for all $t \geq 0$ gives

$$\mathbb{E}_{x \sim \nu_t} [\|e^{-t}x - x\|^2] = (1 - e^{-t})^2 \mathbb{E}_{x \sim \nu_t} [\|x\|^2] \leq t^2 \mathbb{E}_{x \sim \nu_t} [\|x\|^2].$$

Combining the previous three bounds finishes the proof of Lemma 12. $\square$

Lemma 13. *For all $k \in \{0, \dots, N-1\}$ we have*

$$t^2 L^2 \mathbb{E}_{(a,x) \sim (\tilde{p}_0^{(k)}, \tilde{p}_t^{(k)})} [\|\nabla \ln \tilde{p}_{t|0}^{(k)}(x | a)\|^2] \lesssim tdL^2 + t^2 L^2 \mathbb{E}_{x \sim \tilde{p}_t^{(k)}} [\|\nabla \ln p_{T-kh-t}(x)\|^2].$$

Applying Lemmas 12 and 13 into (31), using the chain rule (29), and recalling that $t \leq h$ and $L \geq 1$ yields inequality (24), finishing the proof of **Case 2**.

This concludes the proof of Proposition 1. $\square$

C.2 Proof of Lemma 5

Proof. By the Fokker-Planck equation,

$$\begin{aligned}
\frac{\partial \nu_t}{\partial t} &= \nabla \cdot [\nabla \cdot (\nu_t \widehat{J}) - \nu_t \widehat{f}], \\
\frac{\partial \rho_t}{\partial t} &= \nabla \cdot [\nabla \cdot (\rho_t J) - \rho_t f].
\end{aligned}$$

We now use a result on the time derivative of KL from prior work [78, Section 8.1.2]; see also [79, 6]. For completeness, we provide a full statement of the result in Lemma 20 and a proof in Appendix D.12, which relies the domination conditions. By Lemma 20, the time derivative of KL satisfies

$$\frac{d}{dt} \text{KL}(\rho_t \| \nu_t) = \int \left(\frac{\partial \rho_t}{\partial t} \ln \frac{\rho_t}{\nu_t} - \frac{\partial \nu_t}{\partial t} \frac{\rho_t}{\nu_t} \right) dx.$$

Plugging in the Fokker-Planck equations,

$$\begin{aligned} \frac{d}{dt} \text{KL}(\rho_t \| \nu_t) &= \int \left\{ \nabla \cdot [\nabla \cdot (\rho_t J) - \rho_t f] \ln \frac{\rho_t}{\nu_t} - \nabla \cdot [\nabla \cdot (\nu_t \hat{J}) - \nu_t \hat{f}] \frac{\rho_t}{\nu_t} \right\} dx \\ &= \int \left\{ - \left\langle \nabla \cdot (\rho_t J) - \rho_t f, \nabla \ln \frac{\rho_t}{\nu_t} \right\rangle + \left\langle \nabla \cdot (\nu_t \hat{J}) - \nu_t \hat{f}, \nabla \frac{\rho_t}{\nu_t} \right\rangle \right\} dx \end{aligned} \quad (32)$$

where in the second equality we used integration by parts, which holds because of the boundary conditions. In the rest of the proof, we will transform the integrand in the equation above to show the desired result. For better readability, we omit the subscript t in ρ_t and ν_t from this point onward.

For the integrand, we have

$$\begin{aligned} & - \left\langle \nabla \cdot (\rho J) - \rho f, \nabla \ln \frac{\rho}{\nu} \right\rangle + \left\langle \nabla \cdot (\nu \hat{J}) - \nu \hat{f}, \nabla \frac{\rho}{\nu} \right\rangle \\ &= - \left\langle \nabla \cdot (\rho J) - \rho f, \nabla \ln \frac{\rho}{\nu} \right\rangle + \left\langle \nabla \cdot (\nu \hat{J}) - \nu \hat{f}, \frac{\rho}{\nu} \nabla \ln \frac{\rho}{\nu} \right\rangle \\ &= - \left\langle \nabla \cdot (\rho J) - \rho f, \nabla \ln \frac{\rho}{\nu} \right\rangle + \left\langle \frac{\rho}{\nu} \nabla \cdot (\nu \hat{J}) - \rho \hat{f}, \nabla \ln \frac{\rho}{\nu} \right\rangle \\ &= \left\langle \rho (f - \hat{f}), \nabla \ln \frac{\rho}{\nu} \right\rangle + \left\langle \frac{\rho}{\nu} \nabla \cdot (\nu \hat{J}) - \nabla \cdot (\rho J), \nabla \ln \frac{\rho}{\nu} \right\rangle \end{aligned} \quad (33)$$

where the first equality is by $\nabla \frac{\rho}{\nu} = \frac{\rho}{\nu} \nabla \ln \frac{\rho}{\nu}$. The derivation above splits $\frac{d}{dt} \text{KL}$ into two terms: one depends on the drifts $\hat{f}$ and f , while the other depends on the diffusion terms $\hat{J}$ and J . Observe that the drift-related term matches the second term in the final result, which vanishes when $\hat{f} = f$.

We now focus on the remaining, diffusion-related term:

$$\begin{aligned} & \left\langle \frac{\rho}{\nu} \nabla \cdot (\nu \hat{J}) - \nabla \cdot (\rho J), \nabla \ln \frac{\rho}{\nu} \right\rangle \\ &= \left\langle \frac{\rho}{\nu} (\hat{J} \nabla \nu + \nu \nabla \cdot \hat{J}) - J \nabla \rho - \rho \nabla \cdot J, \nabla \ln \frac{\rho}{\nu} \right\rangle \\ &= \rho \left\langle \hat{J} \nabla \ln \nu + \nabla \cdot \hat{J} - J \nabla \ln \rho - \nabla \cdot J, \nabla \ln \frac{\rho}{\nu} \right\rangle \\ &= \rho \left\langle -\hat{J} \nabla \ln \frac{\rho}{\nu} + (\hat{J} - J) \nabla \ln \rho + \nabla \cdot (\hat{J} - J), \nabla \ln \frac{\rho}{\nu} \right\rangle \\ &= -\rho \left\| \nabla \ln \frac{\rho}{\nu} \right\|_{\hat{J}}^2 + \rho \left\langle (\hat{J} - J) \nabla \ln \rho + \nabla \cdot (\hat{J} - J), \nabla \ln \frac{\rho}{\nu} \right\rangle. \end{aligned} \quad (34)$$

Here, the first equality is by $\nabla \cdot (\nu \hat{J}) = \hat{J} \nabla \nu + \nu \nabla \cdot \hat{J}$ and $\nabla \cdot (\rho J) = J \nabla \rho + \rho \nabla \cdot J$, and the second equality uses $\frac{1}{\nu} \nabla \nu = \nabla \ln \nu$ and $\frac{1}{\rho} \nabla \rho = \nabla \ln \rho$. Combining Eqs. (32) to (34) yields the desired result. $\square$

D Proofs of auxiliary lemmata

D.1 Proof of Lemma 1 – Interpolation SDE for the Backward Algorithm

For ease of notation, within this proof we omit the step index k in $\hat{X}_t^{(k)}$, $\hat{\mu}_k$, and $\hat{G}_k$. For $t \geq 0$, let

$$\tilde{X}_t = (1 - t) \hat{X}_t - 2t \nabla \ln p_{T-kh-t}(\hat{X}_t)$$

so $\tilde{X}_0 = \hat{X}_0$, and we can write (18) as

$$\tilde{X}_t = \tilde{X}_0 + \sqrt{2} W_t.$$

That is, $\tilde{X}_t$ evolves following the SDE

$$d\tilde{X}_t = \sqrt{2} dW_t. \quad (35)$$

Suppose that the process $(\hat{X}_t)_{t \in [0, h]}$ evolves via an SDE of the form

$$d\hat{X}_t = \hat{\mu}(\hat{X}_t, t) dt + \sqrt{2\hat{G}(\hat{X}_t, t)} dW_t$$

for some functions $\hat{\mu}$ and $\hat{G}$. In what follows, we solve for the form of these two functions. Let $T_t(x) = (1-t)x - 2t\nabla \ln p_{T-kh-t}(x) \in \mathbb{R}^d$. Then, we have

$$\begin{aligned}\frac{\partial T_t}{\partial t}(x) &= -x - 2\nabla \ln p_{T-kh-t}(x) - 2t \frac{\partial}{\partial t} \nabla \ln p_{T-kh-t}(x), \\ \nabla T_t(x) &= (1-t)I_d - 2t\nabla^2 \ln p_{T-kh-t}(x), \quad \text{and} \\ \nabla^2 T_t(x) &= -2t\nabla^3 \ln p_{T-kh-t}(x).\end{aligned}$$

We now apply high-dimensional Itô's lemma [86, Chapter 4.2] to $\tilde{X}_t = T_t(\hat{X}_t)$ and obtain the following

$$\begin{aligned}d\tilde{X}_t &= dT_t(\hat{X}_t) \\ &= \frac{\partial T_t}{\partial t}(\hat{X}_t)dt + \nabla T_t(\hat{X}_t)d\hat{X}_t + \sum_{i,j=1}^d \frac{1}{2} \frac{\partial^2 T_t}{\partial x_i \partial x_j}(\hat{X}_t) d\hat{X}_t^i d\hat{X}_t^j \\ &= \frac{\partial T_t}{\partial t}(\hat{X}_t)dt + \nabla T_t(\hat{X}_t)\hat{\mu}dt + \nabla T_t(\hat{X}_t)\sqrt{2\hat{G}}dW_t + \sum_{i,j=1}^d \frac{\partial^2 T_t}{\partial x_i \partial x_j}(\hat{X}_t)\hat{G}_{i,j}dt \\ &= \left(\frac{\partial T_t}{\partial t}(\hat{X}_t) + \nabla T_t(\hat{X}_t)\hat{\mu} + \text{Tr}[\nabla^2 T_t(\hat{X}_t)\hat{G}] \right)dt + \sqrt{2}\nabla T_t(\hat{X}_t)\sqrt{\hat{G}}dW_t. \quad (36)\end{aligned}$$

where the third equality is by plugging in $d\hat{X}_t = \hat{\mu}dt + \sqrt{2\hat{G}}dW_t$ and noting that

$$d\hat{X}_t^i d\hat{X}_t^j = 2\hat{G}_{i,j}dt,$$

and the last step uses $\text{Tr}(AB^T) = \sum_{i,j} A_{i,j}B_{i,j}$. Note that in the equations above, $T_t(\cdot) \in \mathbb{R}^d$ is a vector and therefore, $\frac{\partial^2 T_t(\cdot)}{\partial x_i \partial x_j} \in \mathbb{R}^d$, $\nabla^2 T_t(\cdot) \in \mathbb{R}^{d \times d \times d}$, and $\text{Tr}[\nabla^2 T_t(\cdot)G] \in \mathbb{R}^d$ are obtained by broadcasting the corresponding operator to each entry in $T_t(\cdot)$.

Matching (35) and (36), we must have

$$\begin{aligned}\frac{\partial T_t}{\partial t}(x) + \nabla T_t(x)\hat{\mu} + \text{Tr}[\nabla^2 T_t(x)\hat{G}] &= 0, \\ \nabla T_t(x)\sqrt{\hat{G}} &= I_d\end{aligned}$$

which implies

$$\begin{aligned}\hat{G}(x, t) &= [\nabla T_t(x)]^{-2} \\ \hat{\mu}(x, t) &= \sqrt{\hat{G}} \left\{ -\frac{\partial T_t}{\partial t}(x) - \text{Tr}[\nabla^2 T_t(x)\hat{G}] \right\}.\end{aligned}$$

Plugging in $\frac{\partial T_t}{\partial t}(x)$, $\nabla T_t(x)$, and $\nabla^2 T_t(x)$ completes the proof. $\square$

D.2 Proof of Lemma 2 – Interpolation SDE for the Hybrid Algorithm

Observe that the next iterate of (16), $\hat{X}_{(k+1)h}$, is the value at time $t = h$ of the stochastic process $(\hat{X}_t^{(k)})_{t \in [0, h]}$ given by

$$\hat{X}_t^{(k)} = \hat{X}_0^{(k)} + t\hat{X}_0^{(k)} + 2t\nabla \ln p_{T-kh-t}(\hat{X}_t^{(k)}) + \sqrt{2}W_t, \quad \hat{X}_0^{(k)} = \hat{X}_{kh}. \quad (37)$$

For $t \geq 0$, let

$$\tilde{X}_t = \hat{X}_t^{(k)} - t\hat{X}_0^{(k)} - 2t\nabla \ln p_{T-kh-t}(\hat{X}_t^{(k)})$$

so $\tilde{X}_0 = \hat{X}_0^{(k)}$ and $\tilde{X}_t^{(k)} = \tilde{X}_0^{(k)} + \sqrt{2}W_t$.

The rest of the argument is analogous to the proof of Lemma 1 (see Appendix D.2), but with $T_t(x) = x - t\hat{X}_0^{(k)} - 2t\nabla \ln p_{T-kh-t}(x) \in \mathbb{R}^d$. $\square$

D.3 Proof of Lemma 3 – Second Moment of OU Process

This is a standard property of the OU process. We include a proof here for completeness. Along the OU process (12), we have

$$X_t \stackrel{d}{=} e^{-t} X_0 + \sqrt{1 - e^{-2t}} Z, \quad Z \sim \mathcal{N}(0, I).$$

Therefore,

$$\begin{aligned} \mathbb{E} [\|X_t\|^2] &= \mathbb{E} \left[\left\| e^{-t} X_0 + \sqrt{1 - e^{-2t}} Z \right\|^2 \right] \\ &\leq 2e^{-2t} \mathbb{E} [\|X_0\|^2] + 2(1 - e^{-2t}) \mathbb{E} [\|Z\|^2] \\ &= 2e^{-2t} \mathbb{E} [\|X_0\|^2] + 2(1 - e^{-2t})d \\ &\leq 2\mathbb{E} [\|X_0\|^2] + 2d \end{aligned}$$

where the second line follows by Young's inequality, completing the proof of Lemma 3. $\square$

D.4 Proof of Lemma 4 – Bound on Initialization Mismatch

Proof. Recall that p_t denotes the density of the OU process at time t , thus $p_T = \bar{p}_0$. We have

$$\begin{aligned} \text{KL}(p_t \| q_0) &= \int_{\mathbb{R}^d} p_t(x) \ln p_t(x) dx - \int_{\mathbb{R}^d} p_t(x) \ln q_0(x) dx \\ &= \int_{\mathbb{R}^d} p_t(x) \ln p_t(x) dx + \int_{\mathbb{R}^d} p_t(x) \left(\frac{\|x\|^2}{2} + \frac{d}{2} \ln(2\pi) \right) dx \\ &\leq \int_{\mathbb{R}^d} p_t(x) \ln p_t(x) dx + \frac{1}{2} \mathbb{E}_{p_t} [\|x\|^2] + \frac{d}{2} \ln(2\pi). \end{aligned}$$

For the first term, letting $p_{t|0}$ be the conditional density of X_t given X_0 , we have

$$\begin{aligned} \int_{\mathbb{R}^d} p_t(x) \ln p_t(x) dx &= \int_{\mathbb{R}^d} \left(\int_{\mathbb{R}^d} p_{t|0}(x|y) p_0(y) dy \right) \ln \left(\int_{\mathbb{R}^d} p_{t|0}(x|y) p_0(y) dy \right) dx \\ &\leq \int_{\mathbb{R}^d} \left[\int_{\mathbb{R}^d} p_{t|0}(x|y) \ln p_{t|0}(x|y) p_0(y) dy \right] dx \\ &= \int_{\mathbb{R}^d} \left(\int_{\mathbb{R}^d} p_{t|0}(x|y) \ln p_{t|0}(x|y) dx \right) p_0(y) dy \end{aligned}$$

where the inequality is by Jensen's inequality and that $x \mapsto x \ln x$ is a convex function for $x > 0$. Since $X_t | X_0 \sim \mathcal{N}(e^{-t} X_0, (1 - e^{-2t})I)$, by entropy of Gaussian distributions, we have

$$\int_{\mathbb{R}^d} p_{t|0}(x|y) \ln p_{t|0}(x|y) dx = -\frac{d}{2} \ln(2\pi(1 - e^{-2t})) - \frac{d}{2}.$$

Therefore,

$$\begin{aligned} \text{KL}(p_t \| q_0) &\leq -\frac{d}{2} \ln(2\pi(1 - e^{-2t})) - \frac{d}{2} + \frac{1}{2} \mathbb{E}_{p_t} [\|x\|^2] + \frac{d}{2} \ln(2\pi) \\ &= \frac{1}{2} \mathbb{E}_{p_t} [\|x\|^2] + \frac{d}{2} \left(\ln \frac{1}{1 - e^{-2t}} - 1 \right) \\ &\leq M_2 + d + \frac{d}{2} \left(\ln \frac{1}{1 - e^{-2t}} - 1 \right) \end{aligned}$$

where the last inequality is by Lemma 3. Taking $t' = \frac{1}{2} \ln(\frac{e}{e-1}) \approx 0.229$, we have $\ln \frac{1}{1 - e^{-2t'}} = 1$, thus

$$\text{KL}(p_{t'} \| q_0) \leq M_2 + d.$$

By exponential convergence of the OU process, we have for all $T \geq t'$,

$$\text{KL}(p_T \| q_0) \leq e^{-2(T-t')} \text{KL}(p_{t'} \| q_0) \leq e^{-2(T-t')} \cdot (M_2 + d) < 2e^{-2T} (M_2 + d)$$

as desired. $\square$

D.5 Proof of Lemma 6 – Bounds on the Diffusion Matrix $\widehat{G}_k$ for Backward Algorithm

Since p_{T-kh-t} is L -smooth,

$$-LI \preceq \nabla^2 \ln p_{T-kh-t}(x) \preceq LI.$$

So

$$(1-t-2tL)I \preceq (1-t)I_d - 2t\nabla^2 \ln p_{T-kh-t}(x) \preceq (1-t+2tL)I.$$

Check that $0 < 1-t-2tL \leq 1-t+2tL$ since $t \leq \frac{1}{2(2L+1)}$. Therefore,

$$(1-t+2tL)^{-2}I \preceq \widehat{G}_k = [(1-t)I_d - 2t\nabla^2 \ln p_{T-kh-t}(x)]^{-2} \preceq (1-t-2tL)^{-2}I$$

and

$$(1-t+2tL)^{-1}I \preceq \sqrt{\widehat{G}_k} = [(1-t)I_d - 2t\nabla^2 \ln p_{T-kh-t}(x)]^{-1} \preceq (1-t-2tL)^{-1}I.$$

We collect some useful facts:

$$\begin{aligned} (1-x)^{-2} &\leq 1+6x & \text{if } 0 \leq x \leq \frac{1}{2}, \\ (1-x)^{-2} &\geq 1+2x & \text{if } x < 1, \\ (1-x)^{-1} &\leq 1+2x & \text{if } 0 \leq x \leq \frac{1}{2}, \\ (1-x)^{-1} &\geq 1+x & \text{if } x < 1. \end{aligned}$$

Since $t-2tL \leq \frac{1}{2}$, $0 \leq t+2tL \leq \frac{1}{2}$, and $L \geq 1$,

$$(1-4tL)I \preceq [1+2t(1-2L)]I \preceq \widehat{G}_k \preceq [1+6t(1+2L)]I \preceq (1+18tL)I$$

and

$$(1-2tL)I \leq [1+t(1-2L)]I \preceq \sqrt{\widehat{G}_k} \preceq [1+2t(1+2L)]I \preceq (1+6tL)I$$

as desired.

D.6 Proof of Lemma 9 – Bounds on the Diffusion Matrix $\widehat{G}_k$ for Hybrid Algorithm

The proof is analogous to that of Lemma 6 (see Appendix D.5). The resulting bound is identical up to changes in constant factors.

D.7 Proof of Lemma 7 – Bound on the Drift Difference $\mu_k - \widehat{\mu}_k$ for Backward Algorithm

We start by establishing a few auxiliary lemmas (Lemmas 14 to 16). We let $[b_i]_{i=1}^d \in \mathbb{R}^d$ denote a vector whose i -th entry is b_i .

Lemma 14. *The following two equalities hold true.*

$$\begin{aligned} \frac{\partial}{\partial t} \ln p_t &= \|\nabla \ln p_t\|^2 + \langle \nabla \ln p_t, x \rangle + \Delta(\ln p_t) + d, \quad \text{and} \\ \frac{\partial}{\partial t} \nabla \ln p_{T-kh-t}(x) &= -\text{Tr}(\nabla^3 \ln p_{T-kh-t}(x) \widehat{G}_k) - \nabla (\|\nabla \ln p_{T-kh-t}\|^2) \\ &\quad - \nabla (\langle \nabla \ln p_{T-kh-t}, x \rangle) + \left[\text{Tr}(\nabla_i \nabla^2 \ln p_{T-kh-t}(x) (\widehat{G}_k - I)) \right]_{i=1}^d. \end{aligned}$$

Proof. Applying the Fokker-Planck equation to (12) yields $\frac{\partial p_t}{\partial t} = \nabla \cdot [p_t (\nabla \ln p_t + x)]$. Therefore,

$$\begin{aligned} \frac{\partial}{\partial t} \ln p_t &= \frac{1}{p_t} \frac{\partial p_t}{\partial t} \\ &= \frac{1}{p_t} \nabla \cdot [p_t (\nabla \ln p_t + x)] \\ &= \frac{1}{p_t} (\langle \nabla p_t, \nabla \ln p_t + x \rangle + p_t \nabla \cdot (\nabla \ln p_t + x)) \\ &= \langle \nabla \ln p_t, \nabla \ln p_t \rangle + \langle \nabla \ln p_t, x \rangle + \Delta(\ln p_t) + d \end{aligned}$$

where the second equality uses $\nabla \cdot (fG) = \langle \nabla f, G \rangle + f \nabla \cdot G$, for $f: \mathbb{R}^d \rightarrow \mathbb{R}$ and $G: \mathbb{R}^d \rightarrow \mathbb{R}^d$. We have shown the first result.

Then, note that

$$\begin{aligned} \frac{\partial}{\partial t} \ln p_{T-kh-t}(x) &= -\frac{\partial}{\partial t'} \ln p_{t'}(x) \Big|_{t'=T-kh-t} \\ &= -\|\nabla \ln p_{T-kh-t}\|^2 - \langle \nabla \ln p_{T-kh-t}, x \rangle - \Delta(\ln p_{T-kh-t}) - d. \end{aligned}$$

Therefore,

$$\begin{aligned} \nabla \left(\frac{\partial}{\partial t} \ln p_{T-kh-t}(x) \right) &= -\nabla (\|\nabla \ln p_{T-kh-t}\|^2) - \nabla (\langle \nabla \ln p_{T-kh-t}, x \rangle) - \nabla (\Delta(\ln p_{T-kh-t})) \\ &= -\nabla (\|\nabla \ln p_{T-kh-t}\|^2) - \nabla (\langle \nabla \ln p_{T-kh-t}, x \rangle) - [\text{Tr}(\nabla_i \nabla^2 \ln p_{T-kh-t})]_{i=1}^d. \end{aligned}$$

Recall that

$$\text{Tr}(\nabla^3 \ln p_{T-kh-t}(x) \hat{G}_k) = \left[\text{Tr}(\nabla_i \nabla^2 \ln p_{T-kh-t}(x) \hat{G}_k) \right]_{i=1}^d.$$

Adding the two equations above yields the second result. $\square$

Lemma 15. *The following inequality holds.*

$$\left\| \left[\text{Tr}(\nabla_i \nabla^2 \ln p_{T-kh-t}(x) (\hat{G}_k - I)) \right]_{i=1}^d \right\|^2 \leq 324t^2 d^3 L^2 H_{T-kh-t}(x)^2.$$

Proof. For each entry, we have

$$\begin{aligned} \left| \text{Tr}(\nabla_i \nabla^2 \ln p_{T-kh-t}(x) (\hat{G}_k - I)) \right| &\leq \|\nabla_i \nabla^2 \ln p_{T-kh-t}(x)\|_{\text{F}}^2 \|\hat{G}_k - I\|_{\text{F}}^2 \\ &\leq d^2 \|\nabla_i \nabla^2 \ln p_{T-kh-t}(x)\|_{\text{op}}^2 \|\hat{G}_k - I\|_{\text{op}}^2 \\ &\leq 324t^2 d^2 L^2 H_{T-kh-t}(x)^2. \end{aligned}$$

Summing over the entries yields the desired result. $\square$

Lemma 16. *The following inequality holds.*

$$\left\| \frac{\partial}{\partial t} \nabla \ln p_{T-kh-t}(x) + \text{Tr}(\nabla^3 \ln p_{T-kh-t}(x) \hat{G}_k) \right\| \leq (2L + 1) \|\nabla \ln p_{T-kh-t}\| + L\|x\| + 18td^{\frac{3}{2}} L H_{T-kh-t}(x).$$

Proof. By Lemma 14, we have

$$\begin{aligned} &\left\| \frac{\partial}{\partial t} \nabla \ln p_{T-kh-t}(x) + \text{Tr}(\nabla^3 \ln p_{T-kh-t}(x) \hat{G}_k) \right\| \\ &\leq \|\nabla (\|\nabla \ln p_{T-kh-t}\|^2)\| + \|\nabla (\langle \nabla \ln p_{T-kh-t}, x \rangle)\| + \left\| \left[\text{Tr}(\nabla_i \nabla^2 \ln p_{T-kh-t}(x) (\hat{G}_k - I)) \right]_{i=1}^d \right\|. \end{aligned}$$

From this point onward, we let $\nu_t = p_{T-kh-t}$ for ease of notation. Next, we bound the three terms in this sum. For the first term, we have

$$\nabla (\|\nabla \ln \nu_t\|^2) = \nabla (\langle \nabla \ln \nu_t, \nabla \ln \nu_t \rangle) = 2\nabla^2 \ln \nu_t \nabla \ln \nu_t$$

where the second equality uses $\nabla(\langle f, g \rangle) = J_f^T g + J_g^T f$, where $f, g: \mathbb{R}^n \rightarrow \mathbb{R}^m$, and J_f is the Jacobian of f . Therefore,

$$\|\nabla (\|\nabla \ln \nu_t\|^2)\| \leq 2 \|\nabla^2 \ln \nu_t\|_{\text{op}} \|\nabla \ln \nu_t\| \leq 2L \|\nabla \ln \nu_t\|.$$

To bound the second term, note that $\nabla (\langle \nabla \ln \nu_t, x \rangle) = \nabla \ln \nu_t + \nabla^2 \ln \nu_t x$. So

$$\|\nabla (\langle \nabla \ln \nu_t, x \rangle)\| \leq \|\nabla \ln \nu_t\| + \|\nabla^2 \ln \nu_t\|_{\text{op}} \|x\| \leq \|\nabla \ln \nu_t\| + L\|x\|.$$

Lemma 15 gives a bound on the third term. Combining the three upper bounds finishes the proof of Lemma 16. $\square$

We are ready to prove Lemma 7. We start by decomposing $\hat{\mu}_k - \mu_k$ as

$$\begin{aligned}\hat{\mu}_k - \mu_k &= \left(\sqrt{\hat{G}_k} - I \right) (x + 2\nabla \ln p_{T-kh-t}(x)) \\ &\quad + 2t\sqrt{\hat{G}_k} \left(\frac{\partial}{\partial t} \nabla \ln p_{T-kh-t}(x) + \text{Tr} \left(\nabla^3 \ln p_{T-kh-t}(x) \hat{G}_k \right) \right).\end{aligned}$$

Next, we bound the two terms on the right-hand side. For the first term, we have

$$\begin{aligned}\left\| \left(\sqrt{\hat{G}_k} - I \right) (x + 2\nabla \ln p_{T-kh-t}(x)) \right\| &\leq \left\| \sqrt{\hat{G}_k} - I \right\|_{\text{op}} \|x + 2\nabla \ln p_{T-kh-t}(x)\| \\ &\leq 6tL (\|x\| + 2\|\nabla \ln p_{T-kh-t}(x)\|)\end{aligned}$$

where the second inequality uses the bound $\left\| \sqrt{G} - I \right\|_{\text{op}} \leq 6tL$ by Lemma 6.

For the second term, we have

$$\begin{aligned}&\left\| 2t\sqrt{\hat{G}_k} \left(\frac{\partial}{\partial t} \nabla \ln p_{T-kh-t}(x) + \text{Tr} \left(\nabla^3 \ln p_{T-kh-t}(x) \hat{G}_k \right) \right) \right\| \\ &\leq 2t \left\| \sqrt{\hat{G}_k} \right\|_{\text{op}} \left\| \frac{\partial}{\partial t} \nabla \ln p_{T-kh-t}(x) + \text{Tr} \left(\nabla^3 \ln p_{T-kh-t}(x) \hat{G}_k \right) \right\| \\ &\leq 5t \left[(2L + 1) \|\nabla \ln p_{T-kh-t}\| + L\|x\| + 18td^{\frac{3}{2}} LH_{T-kh-t}(x) \right].\end{aligned}$$

where the second inequality is by Lemma 16 and $\left\| \sqrt{\hat{G}_k} \right\|_{\text{op}} \leq \frac{5}{2}$ from Lemma 6.

The desired bound follows from combining the above, noting that $L \geq 1$ and using $(a + b + c)^2 \leq 3a^2 + 3b^2 + 3c^2$ for $a, b, c \in \mathbb{R}$. $\square$

D.8 Proof of Lemma 10 – Bound on the Drift Difference $\mu_k - \hat{\mu}_k$ for Hybrid Algorithm

The proof follows the same structure as that of Lemma 7 (see Appendix D.7). We start by decomposing $\hat{\mu}_k - \mu_k$ as

$$\begin{aligned}\hat{\mu}_k - \mu_k &= (\sqrt{\hat{G}_k} - I) (x + 2\nabla \ln p_{T-kh-t}(x)) \\ &\quad + 2t\sqrt{\hat{G}_k} \left(\frac{\partial}{\partial t} \nabla \ln p_{T-kh-t}(x) + \text{Tr} \left(\nabla^3 \ln p_{T-kh-t}(x) \hat{G}_k \right) \right) \\ &\quad + \sqrt{\hat{G}_k} (a - x).\end{aligned}$$

The first two terms are bounded analogously to the proof of Lemma 7 (see Appendix D.7). The last term is bounded by $\left\| \sqrt{\hat{G}_k} \right\|_{\text{op}} \leq 2$ thanks to Lemma 9. This completes the proof.

D.9 Proof of Lemma 8 – Bound on the Divergence of Diffusion Matrix $\nabla \cdot \hat{G}_k$ for Backward Algorithm

In this section, we prove Lemma 8, which provides a bound on $\|\nabla \cdot \hat{G}_k\|$. The derivation in this section is largely inspired by [79, Proof of Lemma 6], but we improved the technique to obtain a tighter bound in terms of d and with weaker assumptions.

Before proving Lemma 8, we show a few general lemmas related to matrix perturbation (Lemmas 17 and 18) and divergence of matrix-valued functions (Lemma 19). These results are then used to prove Lemma 8. The next two lemmas are folklore; we include their proofs for completeness. Recall that a matrix norm $\|\cdot\|$ is sub-multiplicative if for all A, B we have

$$\|AB\| \leq \|A\|\|B\|.$$

Lemma 17. *Let A and B be real invertible matrices. Let $\|\cdot\|$ be a sub-multiplicative matrix norm. Then, the following three inequalities hold true*

$$\begin{aligned}\|A^2 - B^2\| &\leq 2\|A - B\|(\|A\| + \|B\|), \\ \|A^{-1} - B^{-1}\| &\leq \|A - B\| \|A^{-1}\| \|B^{-1}\|, \quad \text{and} \\ \|A^{-2} - B^{-2}\| &\leq 2\|A - B\|(\|A\| + \|B\|) \|A^{-2}\| \|B^{-2}\|.\end{aligned}$$

Proof. We start with the first inequality. By the triangular inequality and sub-multiplicativity, we have

$$\begin{aligned}\|A^2 - B^2\| &= \|(B + A - B)^2 - B^2\| \\ &= \|(A - B)^2 + B(A - B) + (A - B)B\| \\ &\leq \|A - B\|^2 + 2\|A - B\| \|B\|.\end{aligned}$$

A combination of symmetry and the triangle inequality yields

$$\begin{aligned}\|A^2 - B^2\| &\leq \|A - B\|^2 + 2\|A - B\| \min(\|A\|, \|B\|) \\ &\leq \|A - B\|(\|A\| + \|B\| + 2\min(\|A\|, \|B\|)) \\ &\leq 2\|A - B\|(\|A\| + \|B\|)\end{aligned}$$

as desired.

The second inequality follows from the identity $A^{-1} - B^{-1} = A^{-1}(B - A)B^{-1}$ and the sub-multiplicativity of the operator norm.

Iteratively applying the first and second results gives the final inequality

$$\|A^{-2} - B^{-2}\| \leq \|A^2 - B^2\| \|A^{-2}\| \|B^{-2}\| \leq 2\|A - B\|(\|A\| + \|B\|) \|A^{-2}\| \|B^{-2}\|,$$

this concludes the proof of Lemma 17. $\square$

Lemma 18. *Let $A: \mathbb{R}^n \rightarrow \mathbb{R}^{m \times m}$ be a matrix-valued function. Assume at a point x , A is continuous and differentiable, and $[A(x)]^{-2}$ exists within a neighborhood of x . Let $\|\cdot\|$ be a sub-multiplicative matrix norm. Let $\varphi(x) = A(x)^{-2}$ and fix a norm-one vector $v \in \mathbb{R}^n$, then*

$$\|\nabla_v \varphi(x)\| \leq 4\|A(x)\| \|[A(x)]^{-2}\|^2 \|\nabla_v A(x)\|.$$

Proof. Let $\varphi(x) = [A(x)]^{-2}$. We have

$$\begin{aligned}\|\nabla_v \varphi(x)\| &= \left\| \lim_{u \rightarrow 0} \frac{\varphi(x + uv) - \varphi(x)}{u} \right\| \\ &= \lim_{u \rightarrow 0} \frac{\|\varphi(x + uv) - \varphi(x)\|}{u} \\ &\leq \lim_{u \rightarrow 0} \frac{2\|A(x + uv) - A(x)\|(\|A(x + uv)\| + \|A(x)\|) \|\varphi(x + uv)\| \|\varphi(x)\|}{u} \\ &= 4\|A(x)\| \|\varphi(x)\|^2 \lim_{u \rightarrow 0} \frac{\|A(x + uv) - A(x)\|}{u} \\ &= 4\|A(x)\| \|\varphi(x)\|^2 \|\nabla_v A(x)\|\end{aligned}$$

where the inequality is by Lemma 17. $\square$

Next, we establish a general result that bounds the divergence of a matrix-valued function by the norm of its derivatives.

Lemma 19. *Let $M: \mathbb{R}^d \rightarrow \mathbb{R}^{d \times d}$ be a matrix-valued function. (Note that $M(x)$ does not need to be a symmetric matrix.) Then, $\|\nabla \cdot M(x)\|^2 \leq d \sum_{j=1}^d \|\nabla_j M(x)\|_{\text{op}}^2$.*

Proof. Denote the i, j -th entry of a matrix A by A_{ij} . Recall that the ℓ_2 norm of each row or column vector of a matrix is upper bounded by the operator norm of the matrix. Thus,

$$\sum_{i=1}^d \left(\frac{\partial M_{ij}(x)}{\partial x_j} \right)^2 = \sum_{i=1}^d (\nabla_j M(x))_{ij}^2 \leq \|\nabla_j M(x)\|_{\text{op}}^2.$$

Hence, applying $\|u\|_1^2 \leq d\|u\|_2^2$ for all $u \in \mathbb{R}^d$ yields

$$\|\nabla \cdot M(x)\|^2 = \sum_{i=1}^d \left(\sum_{j=1}^d \frac{\partial M_{ij}(x)}{\partial x_j} \right)^2 \leq d \sum_{i=1}^d \sum_{j=1}^d \left(\frac{\partial M_{ij}(x)}{\partial x_j} \right)^2 \leq d \sum_{j=1}^d \|\nabla_j M(x)\|_{\text{op}}^2,$$

as desired. This concludes the proof of Lemma 19. $\square$

Finally, we prove Lemma 8.

Proof of Lemma 8. By Lemma 19, we have $\|\nabla \cdot \hat{G}_k\|^2 \leq d \sum_{j=1}^d \|\nabla_j \hat{G}_k\|_{\text{op}}^2$. We now claim that for each $j = 1, \dots, d$,

$$\|\nabla_j \hat{G}_k\|_{\text{op}} \leq 364 t H_{T-kh-t}(x)$$

which will imply the desired bound $\|\nabla \cdot \hat{G}_k\| \leq 364 t d H_{T-kh-t}(x)$. Let $T(x) = (1-t)I_d - 2t\nabla^2 \ln p_{T-kh-t}(x)$ so that $\hat{G}_k(x) = [T(x)]^{-2}$. We invoke Lemma 18 with $v = e_j$ (note that $\|\cdot\|_{\text{op}}$ is sub-multiplicative) to obtain

$$\|\nabla_j \hat{G}_k(x)\|_{\text{op}} \leq 4 \|T(x)\|_{\text{op}} \|\hat{G}_k(x)\|_{\text{op}}^2 \|\nabla_j T(x)\|_{\text{op}}.$$

Note that both $\|T(x)\|_{\text{op}}$ and $\|\hat{G}_k(x)\|_{\text{op}}$ are bounded by constants. On the one hand, Lemma 6 ensures $\|\hat{G}_k\|_{\text{op}} \leq \frac{11}{2}$. On the other, since $\|\nabla^2 \ln p_{T-kh-t}\|_{\text{op}} \leq L$ we have

$$\|T(x)\|_{\text{op}} \leq 1 - t + 2tL \leq 1 + 2tL < \frac{3}{2}$$

where the last inequality follows since $t \leq \frac{1}{2(2L+1)}$ and so $tL < 1/4$. Plugging in the bounds for $\|T(x)\|_{\text{op}}$ and $\|\hat{G}_k(x)\|_{\text{op}}$ gives

$$\|\nabla_j \hat{G}_k(x)\|_{\text{op}} \leq 182 \|\nabla_j T(x)\|_{\text{op}} = 364 t \|\nabla_j \nabla^2 \ln p_{T-kh-t}(x)\|_{\text{op}} \leq 364 t H_{T-kh-t}(x)$$

as desired; concluding the proof Lemma 8. $\square$

D.10 Proof of Lemma 11 – Bound on the Divergence of Diffusion Matrix $\nabla \cdot \hat{G}_k$ for Hybrid Algorithm

The proof follows the same structure as that of Lemma 8 (see Appendix D.9). The key difference is that we let

$$T(x) = I_d - 2t\nabla^2 \ln p_{T-kh-t}(x)$$

so that $\hat{G}_k(x) = [T(x)]^{-2}$. The resulting bound only differs in the constant factor.

D.11 Proof of Lemma 13

For ease of notation, we let $\nu_0 = \hat{p}_0^{(k)}$, $\nu_t = \hat{p}_t^{(k)}$, $\nu_{t|0} = \hat{p}_{t|0}^{(k)}$ and $\nu_{0|t} = \hat{p}_{0|t}^{(k)}$ within this proof.

By Bayes rule, we have

$$\nabla_x \ln \nu_{t|0}(x | a) = \nabla_x [\ln \nu_{0|t}(a | x) + \ln \nu_t(x)].$$

Let $g(z; \mu, \Sigma)$ denote the density of $\mathcal{N}(\mu, \Sigma)$ at z . Then,

$$\begin{aligned}\nu_{0|t}(a | x) &= p_{T-kh|T-kh-t}(a | x) \\ &= g(a; e^{-t}x, (1 - e^{-2t})I).\end{aligned}$$

Therefore,

$$\begin{aligned}\nabla_x \ln \nu_{0|t}(a | x) &= \nabla_x \left[-\frac{\|a - e^{-t}x\|^2}{2(1 - e^{-2t})} \right] \\ &= \frac{e^{-t}(a - e^{-t}x)}{1 - e^{-2t}}.\end{aligned}$$

Plugging the above into the Bayes rule identity, we have

$$\nabla_x \ln \nu_{t|0}(x | a) = \frac{e^{-t}(a - e^{-t}x)}{1 - e^{-2t}} + \nabla_x \ln \nu_t(x).$$

Therefore,

$$\begin{aligned}\mathbb{E}_{(a,x) \sim (\nu_0, \nu_t)} \left[\|\nabla_x \ln \nu_{t|0}(x | a)\|^2 \right] &= \mathbb{E}_{(a,x) \sim (\nu_0, \nu_t)} \left[\left\| \frac{e^{-t}(a - e^{-t}x)}{1 - e^{-2t}} + \nabla_x \ln \nu_t(x) \right\|^2 \right] \\ &\lesssim \mathbb{E}_{(a,x) \sim (\nu_0, \nu_t)} \left[\left\| \frac{e^{-t}(a - e^{-t}x)}{1 - e^{-2t}} \right\|^2 + \|\nabla_x \ln \nu_t(x)\|^2 \right] \\ &= \frac{e^{-2t}}{(1 - e^{-2t})^2} \mathbb{E}_{(a,x) \sim (\nu_0, \nu_t)} \left[\|a - e^{-t}x\|^2 \right] + \mathbb{E}_{\nu_t} \left[\|\nabla_x \ln \nu_t(x)\|^2 \right].\end{aligned}$$

Then, since $\nu_{0|t}(\cdot | x) = \mathcal{N}(e^{-t}x, (1 - e^{-2t})I)$, we have

$$\begin{aligned}\mathbb{E}_{(a,x) \sim (\nu_0, \nu_t)} \left[\|a - e^{-t}x\|^2 \right] &= \mathbb{E}_{x \sim \nu_t} \mathbb{E}_{a \sim \nu_{0|t}(\cdot | x)} \left[\|a - e^{-t}x\|^2 \right] \\ &= d(1 - e^{-2t}).\end{aligned}$$

Therefore,

$$\begin{aligned}t^2 L^2 \mathbb{E}_{(a,x) \sim (\nu_0, \nu_t)} \left[\|\nabla \ln \nu_{t|0}(x | a)\|^2 \right] &\lesssim t^2 L^2 \left\{ \frac{de^{-2t}}{1 - e^{-2t}} + \mathbb{E}_{\nu_t} \left[\|\nabla_x \ln \nu_t(x)\|^2 \right] \right\} \\ &= dL^2 \frac{t^2 e^{-2t}}{1 - e^{-2t}} + t^2 L^2 \mathbb{E}_{\nu_t} \left[\|\nabla_x \ln \nu_t(x)\|^2 \right]\end{aligned}$$

Note that for $0 \leq t \leq h \leq \frac{1}{8L} \leq \frac{1}{8}$, it holds that $1 - e^{-2t} \geq t$. Therefore,

$$\frac{t^2 e^{-2t}}{1 - e^{-2t}} \leq \frac{t^2}{t} = t.$$

The claim has been proved.

D.12 A General Lemma for Time Derivative of KL

We restate the following lemma from prior work [78, Section 8.1.2]; see also [79, 6], and include a proof for completeness.

Lemma 20 (Time derivative of KL). *Let $p_t, q_t: \mathbb{R}^d \rightarrow \mathbb{R}$ be probability density functions and $t \geq 0$. Suppose the following conditions hold.*

1. *There is an integrable function $\theta: \mathbb{R}^d \rightarrow \mathbb{R}$, i.e. $\int \theta(x)dx < \infty$, such that $\left| \frac{\partial}{\partial t} \left(p_t \ln \frac{p_t}{q_t} \right) \right| \leq \theta(x)$ for all t and almost every x .*
2. *There is an integrable function $\kappa: \mathbb{R}^d \rightarrow \mathbb{R}$, i.e. $\int \kappa(x)dx < \infty$, such that $\left| \frac{\partial}{\partial t} p_t \right| \leq \kappa(x)$ for all t and almost every x .*

Then,

$$\frac{d}{dt} \text{KL}(p_t \| q_t) = \int \left(\frac{\partial p_t}{\partial t} \ln \frac{p_t}{q_t} - \frac{\partial q_t}{\partial t} \frac{p_t}{q_t} \right) dx.$$

Proof. We have

$$\begin{aligned} \frac{d}{dt} \text{KL}(p_t \| q_t) &= \frac{d}{dt} \int p_t \ln \frac{p_t}{q_t} dx \\ &= \int \frac{\partial}{\partial t} \left(p_t \ln \frac{p_t}{q_t} \right) dx \\ &= \int \left(\ln \frac{p_t}{q_t} \frac{\partial p_t}{\partial t} + p_t \frac{\partial}{\partial t} \ln \frac{p_t}{q_t} \right) dx \\ &= \int \left(\ln \frac{p_t}{q_t} \frac{\partial p_t}{\partial t} - \frac{p_t}{q_t} \frac{\partial q_t}{\partial t} + \frac{\partial p_t}{\partial t} \right) dx \\ &= \int \left(\frac{\partial p_t}{\partial t} \ln \frac{p_t}{q_t} - \frac{\partial q_t}{\partial t} \frac{p_t}{q_t} \right) dx. \end{aligned}$$

Here, the second equality uses the Leibniz integral rule, which holds because of the first domination condition. The last step is by

$$\int \frac{\partial p_t}{\partial t} dx = \frac{d}{dt} \int p_t dx = 0$$

which applies the Leibniz integral rule on p_t thanks to the second domination condition. $\square$

E Method and Experimental Details

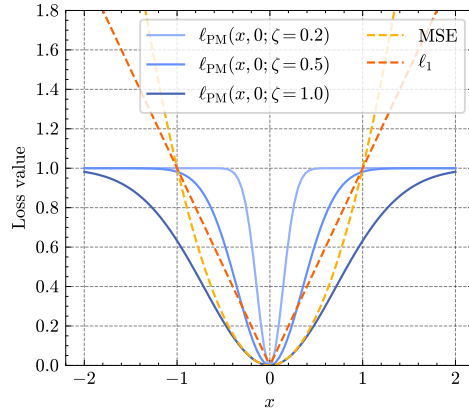


Figure 6: One-dimensional visualization of the proximal matching loss ℓ_{PM} , in comparison with the mean squared error (MSE) and mean absolute error (ℓ_1) losses.

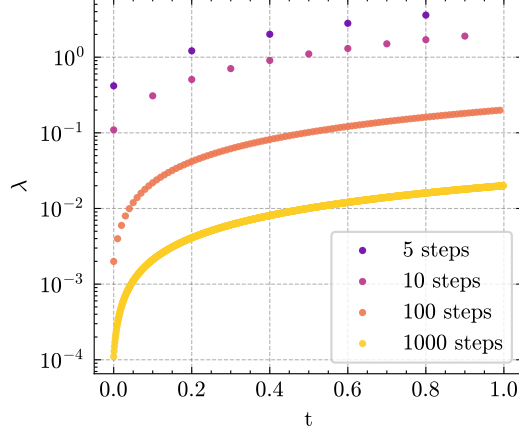


Figure 7: Values of t and λ used by ProxDM-hybrid under varying numbers of sampling steps. Here, we set $\beta(t) = \beta_{\min} + (\beta_{\max} - \beta_{\min})t$, with $\beta_{\min} = 0.1$, $\beta_{\max} = 20$, as used in experiments.

E.1 Pseudo-code for Proximal Matching Training

Algorithm 2 Proximal Matching Training

Input: Forward SDE noise schedule $\beta(t)$ (c.f. (1)); sampling distribution $q(t, \lambda)$ over (t, λ) ; data distribution p_0 ; annealing schedule $\text{Anneal}_\zeta(k)$ for ζ ; optimization algorithm OptimizationStep (e.g., SGD, Adam)
Initialize network parameters θ_0
repeat
 $k \leftarrow k + 1$
 $\zeta_k \leftarrow \text{Anneal}_\zeta(k)$ ▷ Update ζ according to the annealing schedule
 Sample $t, \lambda \sim q(t, \lambda)$, $X_0 \sim p_0$, and $\eta, \varepsilon \sim \mathcal{N}(0, I)$ independently
 $\alpha_t \leftarrow \exp\left(-\int_0^t \beta(s) ds\right)$
 $X_t \leftarrow \sqrt{\alpha_t} X_0 + \sqrt{1 - \alpha_t} \eta$ ▷ Generate $X_t \sim p_t$
 $Y \leftarrow X_t + \sqrt{\lambda} \varepsilon$
 Compute loss $\ell_{\text{PM}} \leftarrow \ell_{\text{PM}}(\epsilon_{\theta_k}(Y; t, \lambda), \varepsilon; \zeta_k)$ ▷ Proximal matching loss (cf. Eq. (9))
 $\theta_{k+1} \leftarrow \text{OptimizationStep}(\theta_k, \nabla_{\theta} \ell_{\text{PM}})$
until convergence
Output: $f_{\theta}(x; t, \lambda) = x - \sqrt{\lambda} \epsilon_{\theta}(x; t, \lambda)$ ▷ Approximated $\text{prox}_{-\lambda \ln p_t}(x)$

E.2 Sampling t and λ

We provide more details of the sampling strategy for t and λ used for proximal matching. To motivate our design, Fig. 7 displays the (t, λ) pairs encountered by ProxDM-hybrid under different numbers of sampling steps: 5, 10, 100, and 1000. The t values are evenly distributed in $[0, 1]$ regardless of the step count, whereas the range of λ varies by the number of steps. Based on this observation, we design the following sampling scheme. We first sample a step number N from a predefined set of candidates. Then, for the chosen N , we uniformly sample one of the possible (t, λ) pairs associated with that step number. This procedure is repeated independently for each sample in the training batch and is performed on-the-fly. For ProxDM-hybrid, the step number candidates are $\{5, 10, 20, 50, 100, 1000\}$. For ProxDM-backward, we exclude 5 and 10 due to the instability associated with small step numbers. To promote balanced learning across different step numbers, we apply weighted sampling, with weights proportional to $\log(N)$ for MNIST and $N^{1/3}$ for CIFAR-10.

E.3 Image Generation

For CIFAR-10, we adopt the same U-Net architecture as in [27]. The ProxDM is trained for a total of 375k iterations with a batch size of 512. We use the ℓ_1 loss during the first 75k iterations as a pretraining step, followed by proximal matching loss with $\zeta = 2$ for 150k iterations, and then $\zeta = 1$ for the last 150k iterations. The learning rate is set to 10^{-4} . For score models, we follow the standard practice [27, 72, 75] of using a batch size of 128, and train for 1.5M iterations to ensure the number of training samples (i.e., effective training epochs) matches that of the proximal model. We set the learning rate to 2×10^{-4} and use gradient clipping and learning rate warm-up, following [27, 75]. For both score and proximal models, the last checkpoint is used, without model selection.

For MNIST, we halve the number of filters in the U-Net, following [72]. The ProxDM is trained for a total of 225k iterations with a batch size of 512, using ℓ_1 loss with learning rate of 10^{-4} during the first 75k iterations, followed by proximal matching loss with $\zeta = 1$ and $\zeta = 0.5$ for 75k iterations each, with a learning rate of 10^{-5} . For the score model, we use a batch size of 128 following the standard practice [27, 72, 75] and train for 900k iterations to ensure the number of training samples (i.e., effective training epochs) matches that of the proximal model, using a learning rate of 2×10^{-4} .

Precision and recall metrics. Following [38], precision measures the proportion of generated samples that fall within the manifold of real data. Specifically, real images are first embedded into a feature space via a pre-trained VGG-16 network [69]. Then, a hypersphere is defined around each feature vector with radius equal to the distance to its k th nearest neighbor, forming a volume that approximates the data manifold. Precision is then computed as the fraction of generated samples located within this volume, intuitively capturing sample quality and fidelity. Analogously, recall is defined as the fraction of real samples that lie within the manifold of generated samples, reflecting the diversity and data coverage of the generation.

F Extension of ProxDM to Other SDEs and ODEs

In this section, we demonstrate conceptually how the proximal diffusion model can be extended to other SDEs and ODEs, such as the variance exploding SDE and probability flow ODE [75]. We leave the theoretical and empirical study of these extensions to future work.

Besides the variance preserving (VP) SDE in (1), an alternative forward process used in diffusion models is the variance exploding (VE) SDE [75], given by

$$dX_t = dW_t. \quad (38)$$

Here we omit the time-dependent diffusion term for simplicity. This corresponds to the standard Brownian motion initialized at the data distribution. The associated reverse-time SDE takes the form

$$dX_t = -\nabla \ln p_t(X_t)dt + d\bar{W}_t$$

where time flows backwards from T to 0.

Given a time grid $\{0 = t_0 < t_1 < \dots < t_N = T\}$, applying forward discretization yields the following score-based algorithms widely used in generative models [72, 73]

$$X_{k-1} = X_k + \gamma_k \ln p_{t_k}(X_k) + \sqrt{\gamma_k}z$$

where $\gamma_k = t_k - t_{k-1}$ and $z \sim \mathcal{N}(0, I)$. Analogous to the derivation of ProxDM for the VP SDE, we can apply backward discretization to the reverse-time VE SDE:

$$X_{k-1} = X_k + \gamma_k \ln p_{t_{k-1}}(X_{k-1}) + \sqrt{\gamma_k}z,$$

and obtain a new proximal sampling scheme:

$$X_{k-1} = \text{prox}_{\gamma_k \ln p_{t_{k-1}}}(X_k + \sqrt{\gamma_k}z).$$

Furthermore, consider a general forward SDE of the form

$$dX_t = f(X_t, t)dt + g(t)dW_t,$$

where $f: \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$ denotes the drift coefficient and $g: \mathbb{R} \rightarrow \mathbb{R}$ the diffusion coefficient. [75] introduce the probability flow (PF) ODE, given by

$$dX_t = \left[f(X_t, t) - \frac{1}{2}g(t)^2 \nabla \ln p_t(X_t) \right] dt,$$

where time flows backwards from T to 0. The PF ODE shares the same marginal probability densities as the forward process, and therefore, admits sampling algorithms analogous to those based on the reverse-time SDE. Notably, the DDIM sampler [70] has been shown to correspond to a discretization of the PF ODE [67, 53].

In particular, for the VP SDE (1), we have $f(x, t) = -\frac{1}{2}\beta(t)x$ and $g(t) = \sqrt{\beta(t)}$, which leads to the following PF ODE

$$dX_t = \left[-\frac{1}{2}\beta(t)X_t - \frac{1}{2}\beta(t)\nabla \ln p_t(X_t) \right] dt.$$

Applying backward discretization yields

$$X_{k-1} = X_k + \frac{\gamma_k}{2}X_{k-1} + \frac{\gamma_k}{2}\nabla \ln p_{t_{k-1}}(X_{k-1}),$$

where $\gamma_k = \int_{t_{k-1}}^{t_k} \beta(s)ds$. This, in turn, gives rise to a proximal-based ODE sampler:

$$X_{k-1} = \text{prox}_{-\frac{\gamma_k}{2-\gamma_k} \ln p_{t_{k-1}}} \left(\frac{2}{2-\gamma_k} X_k \right).$$

G Connection between Backward Discretization and Proximal Algorithms

In this section, we provide the detailed derivations for previously explored connections between proximal algorithms and backward discretization, using two continuous-time processes as examples: gradient flow and Langevin dynamics.

First, consider the gradient flow ODE

$$dx_t = -\nabla f(x_t)dt$$

whose forward (Euler) discretization gives the gradient descent algorithm

$$x^+ = x - \Delta t \nabla f(x)$$

where x and x^+ denote the current and the next iterates, respectively. On the other hand, backward discretization yields the proximal point method [62]:

$$x^+ = x - \Delta t \nabla f(x^+) \quad \longrightarrow \quad x^+ = \text{prox}_{\Delta t f}(x).$$

In the context of sampling, forward discretization of the Langevin dynamics

$$dX_t = -\nabla f(X_t)dt + \sqrt{2}dW_t.$$

leads to the unadjusted Langevin algorithm

$$X^+ = X - \Delta t \nabla f(X) + \sqrt{2\Delta t}z, \quad z \sim \mathcal{N}(0, I)$$

while backward discretization leads to the proximal Langevin algorithm [60, 5, 20, 79]:

$$X^+ = X - \Delta t \nabla f(X^+) + \sqrt{2\Delta t}z \quad \longrightarrow \quad X^+ = \text{prox}_{\Delta t f} \left(X + \sqrt{2\Delta t}z \right).$$

H Additional Results

H.1 Ablation on Smoothness of Target Distribution in 1D

We further investigate how the smoothness of the target distribution affects sampling performance. Specifically, we consider a one-dimensional target distribution given by $p(x) \propto (0.5 + x)^\alpha (0.5 -$

$x)^\alpha$, $x \in [-0.5, 0.5]$, with varying smoothness parameters $\alpha \in \{0, 0.5, 1.0, 2.0, 5.0\}$. This family starts from a uniform distribution over $[0, 1]$ with non-smooth boundaries ($\alpha = 0$) and becomes increasingly smooth as α grows. The score-based sampler uses the canonical Euler–Maruyama discretization (without the additional final denoising step), while the proximal sampler follows the hybrid update in Algorithm 1. As shown in Fig. 8, both samplers recover the target distribution when given sufficient sampling steps. The dependence of sample quality on the smoothness parameter α is relatively subtle—likely due to the smoothing effect inherent in the forward diffusion process. Nevertheless, ProxDM provides samples that consistently fall within the true support and achieves better alignment with the target density, especially at smaller step counts.

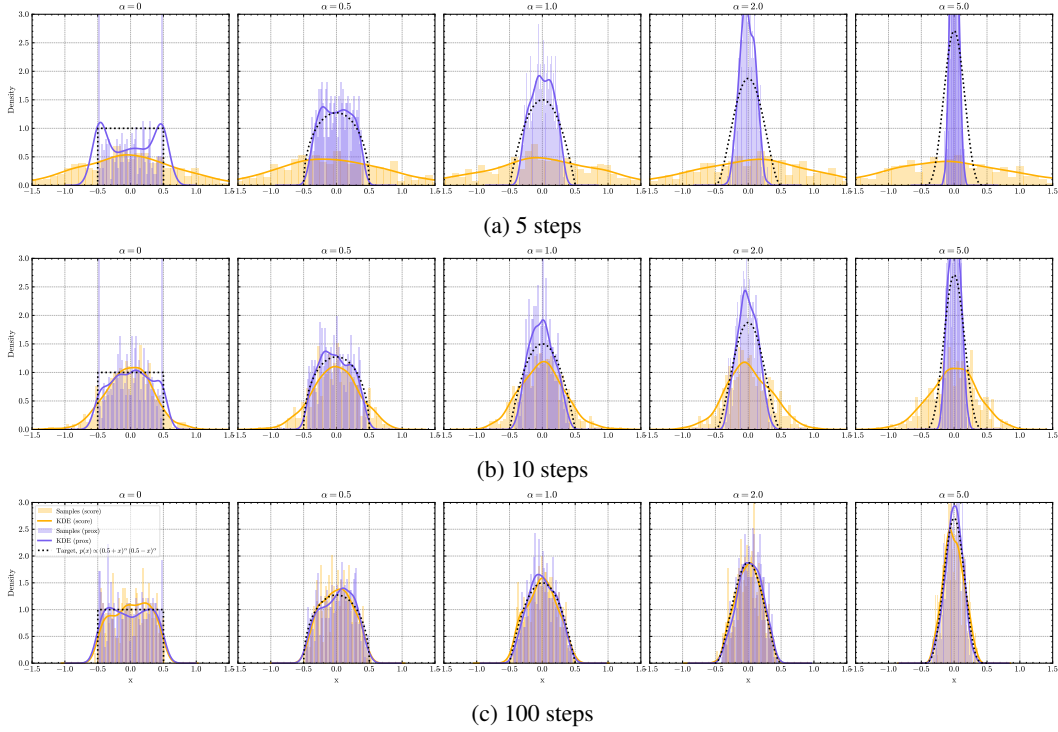


Figure 8: Samples from score-based and proximal samplers using 5, 10, and 100 sampling steps for distributions with varying level of smoothness at the boundary (controlled by α).

H.2 Additional Results

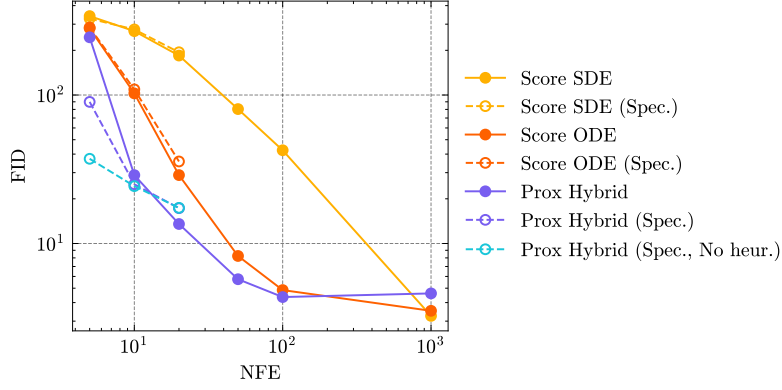


Figure 9: FID vs. number of sampling steps (or number of function evaluations, NFE) on CIFAR10. Dashed lines indicate models trained specifically for 5, 10 and 20 steps (“Spec.”), while solid lines represent full-range models. The result labeled “Prox Hybrid (Spec., No Heur.)” is obtained without applying the heuristics for network parameterization and objective balancing described in Section 3.3.

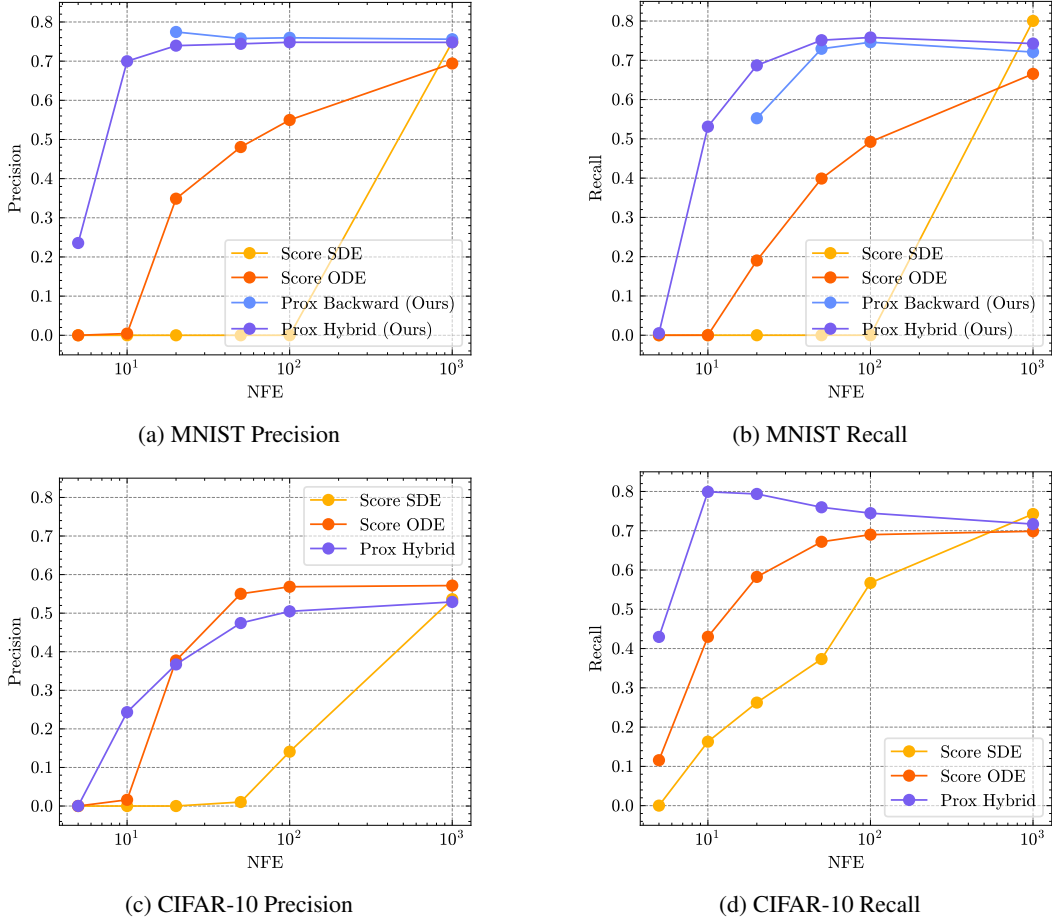


Figure 10: Precision and recall metrics [38] for MNIST and CIFAR-10 datasets. Precision measures the sample quality while recall reflects the coverage of the data distribution. Across most sampling steps (NFE), ProxDM achieves higher precision and recall than score-based baselines, indicating improved sample quality and diversity.

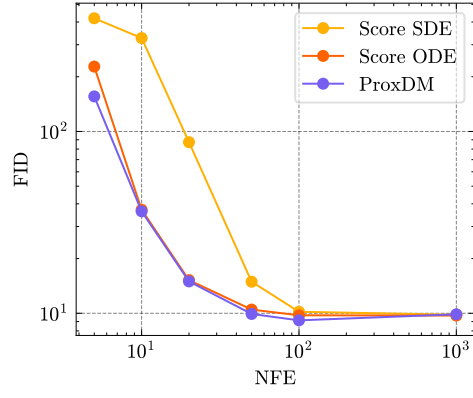


Figure 11: FID vs. number of sampling steps (NFE) on CelebA-HQ (256×256) from score SDE, score ODE and hybrid ProxDM samplers. FID is computed over 30,000 generated samples.



Figure 12: Uncurated CelebA-HQ (256×256) samples from score SDE, score ODE and hybrid ProxDM samplers. ProxDM produces cleaner and more coherent samples with fewer sampling steps.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction clearly state the scope of the paper, its contributions, and results.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We discuss limitations in Section 6.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: The full set of assumptions and proof are provided in appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We provide details on network architecture, loss function, training setups in Sections 3 and 5, as well as in appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All data and models used in this paper are public. We make our code available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Yes, we provide all the details necessary to understand the results presented in the main text. We include further details in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Experiments require significant amount of GPU time to run, so it is quite costly to compute error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Yes, we specify the hardware details used to run experiments in appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Yes, our research complies with the Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss societal impacts of our work in appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: We will implement safeguards when releasing our model.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The sources of datasets are properly cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: No new assets introduced.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: No crowdsourcing or research with human is conducted.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The method does not involve LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.