Journal of Statistical
Mechanics: Theory and
Experiment

PAPER • OPEN ACCESS

Geometric regularity in deterministic sampling dynamics of diffusion-based generative models*

To cite this article: Defang Chen *et al J. Stat. Mech.* (2025) 124002

OPEN ACCESS

Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

PAPER: ML 2025

Geometric regularity in deterministic sampling dynamics of diffusion-based generative models[*]

Defang Chen[1], **Zhenyu Zhou**[2], **Can Wang**[2] and **Siwei Lyu**[1,**]

[1] Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY, United States of America
[2] College of Computer Science, Zhejiang University, Hangzhou, Zhejiang, People's Republic of China
E-mail: siweilyu@buffalo.edu, defangch@buffalo.edu, zhyzhou@zju.edu.cn and wcan@zju.edu.cn

Received 21 July 2025
Accepted for publication 17 October 2025
Published 8 December 2025

Online at stacks.iop.org/JSTAT/2025/124002
https://doi.org/10.1088/1742-5468/ae17ac

Abstract. Diffusion-based generative models employ stochastic differential equations and their equivalent probability flow ordinary differential equations to establish a smooth transformation between complex high-dimensional data distributions and tractable prior distributions. In this paper, we reveal a striking geometric regularity in the deterministic sampling dynamics of diffusion generative models: each simulated sampling trajectory along the gradient field lies within an extremely low-dimensional subspace, and all trajectories exhibit an almost identical 'boomerang' shape, regardless of the model architecture, applied conditions, or generated content. We characterize several intriguing properties

[*] This article is an updated version of a paper presented at ICML2024 conference (Chen D, Zhou Z, Wang C, Shen C and Lyu S 2024 On the trajectory regularity of ODE-based diffusion sampling *Proc. 41st Int. Conf. on Machine Learning* (*Proc. Machine Learning Research* vol 235) ed Salakhutdinov R, Kolter Z, Heller K, Weller A, Oliver N, Scarlett J and Berkenkamp F (PMLR) pp 7905-34) with improved organization and additional contributions in both theoretical and empirical aspects.

[**] Author to whom any correspondence should be addressed.

of these trajectories, particularly under closed-form solutions based on kernel-estimated data modeling. We also demonstrate a practical application of the discovered trajectory regularity by proposing a dynamic programming-based scheme to better align the sampling time schedule with the underlying trajectory structure. This simple strategy requires minimal modification to existing deterministic numerical solvers, incurs negligible computational overhead, and achieves superior image generation performance, especially in regions with only 5–10 function evaluations.

Keywords: diffusion-based generative models, sampling dynamics, trajectory regularity, low-dimensional structure

Contents

1. Introduction

Diffusion-based generative models (Sohl-Dickstein *et al* 2015, Song and Ermon 2019, Ho *et al* 2020, Song *et al* 2021c, Karras *et al* 2022, Chen *et al* 2024), originally inspired by nonequilibrium statistical mechanics (Jarzynski 1997, Sohl-Dickstein *et al* 2015, Bahri *et al* 2020), have recently garnered significant attention and achieved remarkable results in image (Dhariwal and Nichol 2021, Rombach *et al* 2022), audio (Kong *et al* 2021, Huang *et al* 2023), video (Ho *et al* 2022, Blattmann *et al* 2023), and notably in text-to-image synthesis (Saharia *et al* 2022, Ruiz *et al* 2023, Esser *et al* 2024, Podell *et al* 2024). These models introduce noise into data through a *forward process* and subsequently generate data by sampling via a *backward process*. Both processes are characterized and modeled using stochastic differential equations (SDEs) (Song *et al* 2021c). In diffusion-based generative models, the pivotal element is the score function, defined as the gradient of the log data density *w.r.t.* the input (Hyvärinen 2005, Lyu 2009, Raphan and Simoncelli 2011, Vincent 2011), irrespective of specific model configurations. Training such a model involves learning the score function, which can be equivalently achieved by training a noise-dependent denoising model to minimize the mean squared error in data reconstruction, using the data-noise pairings generated during the forward process (Karras *et al* 2022, Chen *et al* 2024). To synthesize new data, diffusion-based generative models solve the acquired score-based backward SDE through a numerical solver. Recent research has shown that the backward SDE can be effectively replaced by an equivalent probability flow ordinary differential equation (PF-ODE), preserving identical marginal distributions (Song *et al* 2021a, 2021c, Lu *et al* 2022a, Zhang and Chen 2023, Zhou *et al* 2024a). This deterministic ODE-based generation reduces the need for stochastic sampling to just the randomness in the initial

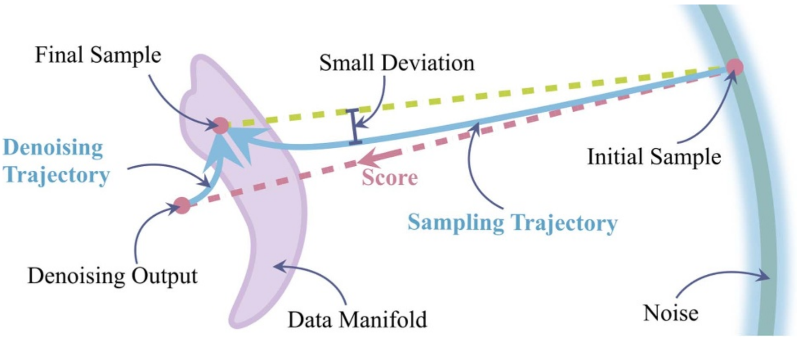


Figure 1. A geometric picture of deterministic sampling dynamics in diffusion-based generative models. Each initial sample (from the noise distribution) starts from a big sphere and converges to the final sample (in the data manifold) along a regular *sampling trajectory*. The score direction points to the denoising output of the current position, and the denoising output forms an implicit *denoising trajectory* controlling the explicit sampling trajectory. Each sampling trajectory inherently lies in a low-dimensional subspace with almost the same shape.

sample selection, thereby simplifying and granting more control over the entire generative process (Song *et al* 2021a, 2021c). Under the PF-ODE formulation, starting from white Gaussian noise, the *sampling trajectory* is formed by running a numerical solver with discretized time steps. These steps collectively constitute the *time schedule* used in sampling.

Despite the impressive generative capabilities exhibited by diffusion-based models, many mathematical and statistical aspects of these models remain veiled in mystery. This obscurity primarily stems from the inherent complexity of the associated SDEs, the nonlinear nature of neural network parameterizations, and the high dimensionality of real-world data (Biroli and Mézard 2023, 2024, Biroli *et al* 2024, Ghio *et al* 2024, Achilli *et al* 2025, Ikeda *et al* 2025, Yu and Huang 2025). In this paper, we reveal a striking regularity in the deterministic sampling dynamics of diffusion models, i.e. the tendency of sample paths to exhibit a consistent 'boomerang' shape, as illustrated in figure 1. More precisely, we observe that each sampling trajectory barely strays from the displacement vector connecting its starting and ending points (section 3.1), while the trajectory deviation can be effectively captured using two orthogonal bases (section 3.2). Therefore, the sampling trajectory in the original high-dimensional data space can be faithfully represented by its projection onto a three-dimensional subspace. These projected spatial curves are fully characterized by the *Frenet–Serret formulas* and exhibit a remarkably consistent geometric structure, irrespective of initial random samples, applied control signals, or target data samples (figure 5 and section 3.3). This intrinsic regularity provides theoretical support for several empirical practices in the literature, such as employing a shared time schedule across different samples and using large sampling steps with negligible truncation error (Song *et al* 2021a, Karras *et al* 2022, Lu *et al* 2022a), particularly during the initial stage of generation (Dockhorn *et al* 2022, Zhou *et al* 2024a).

The geometric trajectory regularity of deterministic sampling trajectories has not been previously investigated. This work aims to elucidate this phenomenon. We begin by simplifying any ODE-based sampling trajectory to its drift-free counterpart (section 2.3), which reveals an implicit *denoising trajectory* controlling the direction of the associated sampling dynamics (section 4.1). Building on this insight, we establish a connection between the closed-form solution of denoising trajectory, which is derived under kernel density estimates (KDEs) with varying bandwidths to approximate the data distribution perturbed by different noise levels, and the classical mean-shift algorithm (Fukunaga and Hostetler 1975, Cheng 1995, Comaniciu and Meer 2002). Although the KDE-based solution is not directly tractable for practical trajectory simulation, it asymptotically converges to the optimal solution derived from the real data distribution and provides a solid foundation for theoretical analysis of our discovered trajectory structure. We further characterize the deterministic sampling dynamics from both local and global perspectives: locally, they exhibit stepwise rotation and monotone likelihood increase; globally, they follow a linear–nonlinear–linear mode-seek path of approximately constant length, as implied by this interpretation of the PF-ODE (section 4.2). Moreover, we theoretically analyze the trajectory deviation under the Gaussian data assumption (section 4.3). This geometric regularity unifies prior empirical observations and clarifies several existing heuristics for accelerating diffusion sampling. As a demonstration of this insight, we develop an efficient and effective accelerated sampling algorithm based on dynamic programming (DP) to determine the optimal time schedule (section 5). Experimental results demonstrate that the proposed approach significantly improves the performance of diffusion-based generative models using only a few ($\leqslant 10$) function evaluations. Our main contributions are summarized as follows:

- We demonstrate and characterize a strong geometric regularity in deterministic sampling dynamics of diffusion-based generative models, i.e. each sampling trajectory exhibits a consistent 'boomerang'-shaped structure confined to an extremely low-dimensional subspace.
- We provide theoretical explanations for this regularity through closed-form analyses of the denoising trajectory under the empirical data distribution and under the Gaussian data assumption. Several derived properties offer insights into both the local and global structures of sampling trajectories.
- We develop a DP-based algorithm that leverages the trajectory regularity to determine an optimal sampling time schedule. It incurs negligible computational overhead while substantially improving image quality, particularly in few-step inference regimes.

2. Preliminaries

2.1. Generative modeling with SDEs

For successful generative modeling, it is essential to connect the data distribution p_{d} with a manageable, non-informative noise distribution p_{n}. Diffusion models achieve this objective by incrementally introducing white Gaussian noise into the data, effectively

https://doi.org/10.1088/1742-5468/ae17ac

obliterating its structures, and subsequently reconstructing the synthesized data from noise samples via a series of denoising steps. A typical choice for p_n is an isotropic multivariate normal distribution with zero mean. The forward step can be modeled as a diffusion process $\{\mathbf{z}_t\}$ for $t \in [0, T]$ starting from the initial condition $\mathbf{z}_0 \sim p_\mathrm{d}$, which corresponds to the solution of an Itô SDE (Oksendal 2013, Song *et al* 2021c)

$$\mathrm{d}\mathbf{z}_t = \mathbf{f}(\mathbf{z}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t, \quad \mathbf{f}(\cdot, t): \mathbb{R}^d \to \mathbb{R}^d, \quad g(\cdot): \mathbb{R} \to \mathbb{R}, \tag{1}$$

where $\mathbf{w}_t$ denotes the Wiener process; $\mathbf{f}(\cdot, t)$ is a vector-valued function referred to as *drift* coefficient and $g(\cdot)$ is a scalar function referred to as *diffusion* coefficient[3]. The temporal marginal distribution of $\mathbf{z}_t$ is denoted as $p_t(\mathbf{z}_t)$, with $p_0(\mathbf{z}_0) = p_\mathrm{d}(\mathbf{z}_0)$. By properly setting the coefficients and terminal time T, the data distribution p_d is smoothly transformed to the approximate noise distribution $p_\mathrm{T}(\mathbf{z}_\mathrm{T}) \approx p_\mathrm{n}$ in a forward manner. The solutions to Itô SDEs are always Markov processes, and they can be fully characterized by the transition kernel $p_{st}(\mathbf{z}_t|\mathbf{z}_s)$ with $0 \leqslant s < t \leqslant T$. This transition kernel becomes a Gaussian distribution when considering the linear SDE with an affine drift coefficient $\mathbf{f}(\mathbf{z}_t, t) = f(t)\mathbf{z}_t$. In this case, we can directly sample data $\mathbf{z}_0$ and its corrupted version $\mathbf{z}_t$ with different levels of noise, which largely simplifies the computation of the forward process and eases the model training. Therefore, linear SDEs are widely used in practice[4]. The transition kernel $p_{0t}(\mathbf{z}_t|\mathbf{z}_0)$ derived with standard techniques (Särkkä and Solin 2019, Karras *et al* 2022) has the following analytic form

$$p_{0t}(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}\left(\mathbf{z}_t; s(t)\,\mathbf{z}_0, s^2(t)\,\sigma^2(t)\,\mathbf{I}\right), \tag{2}$$

or equivalently, $\mathbf{z}_t = s(t)\mathbf{z}_0 + [s(t)\sigma(t)]\,\boldsymbol{\epsilon}_t$, where $s(t) = \exp(\int_0^t f(\xi)\mathrm{d}\xi)$, $\sigma(t) = \sqrt{\int_0^t [g(\xi)/s(\xi)]^2\,\mathrm{d}\xi}$, and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$. For notation simplicity, we hereafter denote them as s_t and σ_t, respectively. Then, we can rewrite the forward linear SDE (1) in terms of s_t and σ_t,

$$\mathrm{d}\mathbf{z}_t = \frac{\mathrm{d}\log s_t}{\mathrm{d}t}\mathbf{z}_t\,\mathrm{d}t + s_t\sqrt{\frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t}}\,\mathrm{d}\mathbf{w}_t, \quad f(t) = \frac{\mathrm{d}\log s_t}{\mathrm{d}t}, \quad \text{and} \quad g(t) = s_t\sqrt{\frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t}}. \tag{3}$$

Furthermore, following previous works (Kingma *et al* 2021, Rombach *et al* 2022), we define the signal-to-noise ratio (SNR) of the transition kernel (2) as $\mathrm{SNR}(t) = s_t^2/(s_t^2\sigma_t^2) = 1/\sigma_t^2$, which is a monotonically non-increasing function of t. A simple corollary is that any linear diffusion process with the same σ_t exhibits an identical SNR function. Two specific forms of linear SDEs, namely, the variance-preserving (VP) SDE and the variance-exploding (VE) SDE (Song *et al* 2021c, Karras *et al* 2022) are widely used in large-scale diffusion models, see more details in appendix A.1.

[3] The noise term in this case is independent of the state $\mathbf{z}_t$ (*a.k.a.* additive noise), and therefore the Itô and Stratonovich interpretations of the above SDE coincide (Stratonovich 1968, Särkkä and Solin 2019). A unique, strong solution of this SDE exists when the time-varying drift and diffusion coefficients are globally Lipschitz in both state and time (Oksendal 2013).

[4] Some non-linear diffusion-based generative models also exist (Zhang and Chen 2021, Chen *et al* 2022, Liu *et al* 2023a), but they are beyond the scope of this paper.

https://doi.org/10.1088/1742-5468/ae17ac

The reversal of the forward linear SDE as expressed in (3) is represented by another backward SDE, which facilitates the synthesis of data from noise through a backward sampling (Feller 1949, Anderson 1982). Based on the well-known Fokker–Planck–Kolmogorov (FPK) equation that describes the evolution of $p_t(\mathbf{z}_t)$ given the initial condition $p_0(\mathbf{z}_0)=p_\mathrm{d}(\mathbf{z}_0)$ (Oksendal 2013), i.e.

$$\frac{\partial p_t\left(\mathbf{z}_t\right)}{\partial t}=-\nabla\cdot\left[p_t\left(\mathbf{z}_t\right)f\left(t\right)\mathbf{z}_t-\frac{g^2\left(t\right)}{2}\nabla_{\mathbf{z}_t}p_t\left(\mathbf{z}_t\right)\right], \tag{4}$$

it is straightforward to verify that a family of backward diffusion processes with varying η_t, as described by the following formula, all maintain the same temporal marginal distributions $\{p_t(\mathbf{z}_t)\}_{t=0}^{\mathrm{T}}$ as the forward SDE at each time throughout the diffusion process

$$\mathrm{d}\mathbf{z}_t=\left[f\left(t\right)\mathbf{z}_t-\frac{1+\eta_t^2}{2}g^2\left(t\right)\nabla_{\mathbf{z}_t}\log p_t\left(\mathbf{z}_t\right)\right]\mathrm{d}t+\eta_t g\left(t\right)\mathrm{d}\bar{\mathbf{w}}_t, \tag{5}$$

where η_t controls the amount of stochasticity and $\bar{\mathbf{w}}_t$ denotes the Wiener process when time flows backwards. Notably, there exists a particular deterministic process with the parameter $\eta_t\equiv 0$, termed PF-ODE in the literature (Song *et al* 2021c, Karras *et al* 2022). PF-ODE describes a time-dependent vector field, which can directly initialize a generative modeling framework and then induce the associated probability path (Albergo *et al* 2023, Lipman *et al* 2023, Liu *et al* 2023b). The deterministic nature of ODE offers several benefits in generative modeling, including efficient sampling, unique encoding, and meaningful latent manipulations (Song *et al* 2021a, 2021c, Chen *et al* 2024). We thus choose this mathematical formula to analyze the sampling behavior of diffusion models throughout this paper.

2.2. Score estimation and diffusion sampling

Simulating the preceding PF-ODE requires having access to the score function $\nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t)$ (Hyvärinen 2005, Lyu 2009), which is typically estimated with denoising score matching (DSM) (Vincent 2011, Song and Ermon 2019, Karras *et al* 2022). Thanks to a profound connection between the score function and the posterior expectation from the perspective of *empirical Bayes* (Robbins 1956, Morris 1983, Efron 2010, Raphan and Simoncelli 2011), we can also train a denoising autoencoder (DAE) (Vincent *et al* 2008, Bengio *et al* 2013b, Alain and Bengio 2014) to estimate the conditional expectation $\mathbb{E}(\mathbf{z}_0|\mathbf{z}_t)$, and then convert it to the score function, see more details in appendix A.2. We summarize this connection as the following lemma.

Lemma 1. *Let the clean data be* $\mathbf{z}_0\sim p_\mathrm{d}$, *and consider a transition kernel that adds Gaussian noise to the data,* $p_{0t}(\mathbf{z}_t|\mathbf{z}_0)=\mathcal{N}\left(\mathbf{z}_t;s_t\mathbf{z}_0,\, s_t^2\sigma_t^2\mathbf{I}\right)$. *Then the score function is related to the posterior expectation by*

$$\nabla_{\mathbf{z}_t}\log p_t\left(\mathbf{z}_t\right)=\left(s_t\sigma_t\right)^{-2}\left(s_t\mathbb{E}\left(\mathbf{z}_0|\mathbf{z}_t\right)-\mathbf{z}_t\right), \tag{6}$$

or equivalently, by linearity of expectation,

$$\nabla_{\mathbf{z}_t} \log p_t\left(\mathbf{z}_t\right) = -\left(s_t\sigma_t\right)^{-1} \mathbb{E}_{p_{t0}(\mathbf{z}_0|\mathbf{z}_t)}\boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t = \left(s_t\sigma_t\right)^{-1}\left(\mathbf{z}_t - s_t\mathbf{z}_0\right). \tag{7}$$

Therefore, we can train a *data-prediction model* $r_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$ to approximate the posterior expectation $\mathbb{E}(\mathbf{z}_0|\mathbf{z}_t)$, or train a *noise-prediction model* $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$ to approximate the posterior expectation $\mathbb{E}\left(\frac{\mathbf{z}_t - s_t\mathbf{z}_0}{s_t\sigma_t}|\mathbf{z}_t\right)$, and then substitute the score in (5) with the learned model for the diffusion sampling process. The DAE objective function of training a data-prediction model $r_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$ across different noise levels with a weighting function $\lambda(t)$ is

$$\mathcal{L}_{\mathrm{DAE}}\left(\boldsymbol{\theta};\lambda\left(t\right)\right) := \int_0^T \lambda\left(t\right)\mathbb{E}_{\mathbf{z}_0\sim p_{\mathrm{d}}}\mathbb{E}_{\mathbf{z}_t\sim p_{0t}(\mathbf{z}_t|\mathbf{z}_0)}\|r_{\boldsymbol{\theta}}\left(\mathbf{z}_t;t\right) - \mathbf{z}_0\|_2^2 \mathrm{d}t. \tag{8}$$

Lemma 2. *The optimal estimator $r^{\star}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$ for the DAE objective, also known as the Bayesian least squares estimator or minimum mean square error (MMSE) estimator, is given by $\mathbb{E}\left(\mathbf{z}_0|\mathbf{z}_t\right)$.*

In particular, this optimal estimator admits a closed-form solution under the empirical data distribution (Karras *et al* 2022, Scarvelis *et al* 2023, Chen *et al* 2023a), as stated in the following lemma.

Lemma 3. *Let $\mathcal{D} := \{\mathbf{y}_i \in \mathbb{R}^d\}_{i\in\mathcal{I}}$ denote a dataset of $|\mathcal{I}|$ i.i.d. data points drawn from p_{d}. When training a DAE with the empirical data distribution $\hat{p}_d$, the optimal denoising output is a convex combination of original data points, namely*

$$r^{\star}_{\boldsymbol{\theta}}\left(\mathbf{z}_t;t\right) = \min_{r_{\boldsymbol{\theta}}} \mathbb{E}_{\mathbf{y}\sim\hat{p}_d}\mathbb{E}_{\mathbf{z}_t\sim p_{0t}(\mathbf{z}_t|\mathbf{y})}\|r_{\boldsymbol{\theta}}\left(\mathbf{z}_t;t\right) - \mathbf{y}\|_2^2 = \sum_i \frac{\exp\left(-\|\mathbf{z}_t - \mathbf{y}_i\|_2^2/2\sigma_t^2\right)}{\sum_j \exp\left(-\|\mathbf{z}_t - \mathbf{y}_j\|_2^2/2\sigma_t^2\right)}\mathbf{y}_i, \tag{9}$$

where $\hat{p}_d(\mathbf{y})$ is the sum of multiple Dirac delta functions, i.e. $\hat{p}_d(\mathbf{y}) = (1/|\mathcal{I}|)\sum_{i\in\mathcal{I}}\delta(\|\mathbf{y} - \mathbf{y}_i\|)$.

In practice, it is assumed that $\nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t) \approx \left(s_t\sigma_t\right)^{-2}\left(s_t r_{\boldsymbol{\theta}}(\mathbf{z}_t;t) - \mathbf{z}_t\right)$ for a converged model[5], and we can plug it into (5) with $\eta_t \equiv 0$ to derive the *empirical* PF-ODE for sampling as follows

$$\begin{aligned}\frac{\mathrm{d}\mathbf{z}_t}{\mathrm{d}t} &= \frac{\mathrm{d}\log s_t}{\mathrm{d}t}\mathbf{z}_t - \frac{\mathrm{d}\log\sigma_t}{\mathrm{d}t}\left(s_t r_{\boldsymbol{\theta}}\left(\mathbf{z}_t;t\right) - \mathbf{z}_t\right)\\ &= \frac{\mathrm{d}\log s_t}{\mathrm{d}t}\mathbf{z}_t + s_t\frac{\mathrm{d}\sigma_t}{\mathrm{d}t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{z}_t;t\right).\end{aligned} \tag{10}$$

Both the data-prediction model $r_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$ and the noise-prediction model $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$ above are widely used in existing works (Ho *et al* 2020, Song *et al* 2021a, Bao *et al* 2022, Karras *et al* 2022, Lu *et al* 2022b, Zhang and Chen 2023, Chen *et al* 2024, Zhou *et al* 2024a).

[5] We slightly abuse the notation and still denote the converged model as $r_{\boldsymbol{\theta}}(\cdot;t)$ hereafter.

Given the empirical PF-ODE (10), we can synthesize novel samples by first drawing pure noises $\hat{\mathbf{z}}_{t_N} \sim p_{\mathrm{n}}$ as the initial condition, and then numerically solving this equation backward with N steps to obtain a sequence $\{\hat{\mathbf{z}}_{t_n}\}_{n=0}^{N}$ with a certain time schedule $\Gamma = \{t_0 = \epsilon, \cdots, t_N = T\}$. We adopt hat notations such as $\hat{\mathbf{z}}_{t_n}$ to denote the samples generated by numerical methods, which differs from the exact solutions denoted as $\mathbf{z}_{t_n}$. The final sample $\hat{\mathbf{z}}_{t_0}$ is considered to approximately follow the data distribution p_{d}. We designate this sequence as a **sampling trajectory** generated by the diffusion model. More details about numerical approximation can be found in appendix A.3.

2.3. The equivalence of diffusion models

We further demonstrate that diffusion models modeled by linear SDEs are equivalent up to a scaling transformation, provided they share the same SNR function of transition kernels (2). In particular, any other model type (e.g. the VP diffusion process) can be transformed into its VE counterparts via the following lemma.

Lemma 4. *The linear diffusion process defined as (3) can be transformed into its VE counterpart with the change of variables $\mathbf{x}_t = \mathbf{z}_t/s_t$, keeping the SNR function unchanged.*

Similarly, we provide the PF-ODE and its empirical version in terms of the $\mathbf{x}$ variable (or say, in the $\mathbf{x}$-space) as follows

$$\mathrm{d}\mathbf{x}_t = -\sigma_t \nabla_{\mathbf{x}_t} \log p_t\left(\mathbf{x}_t\right) \mathrm{d}\sigma_t = \frac{\mathbf{x}_t - r_{\boldsymbol{\theta}}\left(\mathbf{x}_t; t\right)}{\sigma_t} \mathrm{d}\sigma_t = \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{x}_t; t\right) \mathrm{d}\sigma_t, \tag{11}$$

with the score function $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = s_t \nabla_{\mathbf{z}_t} \log p_t(\mathbf{z}_t)$, for $t \in [0, T]$. Because of the above analysis, we can safely remove the drift term in the forward SDE (3) by transforming them into the VE counterparts without changing the essential characteristics of the underlying diffusion model. In the following discussions, we merely focus on the mathematical properties and geometric behaviors of a standardized VE-SDE, i.e.

$$\mathrm{d}\mathbf{x}_t = \sqrt{\mathrm{d}\sigma_t^2/\mathrm{d}t}\, \mathrm{d}\mathbf{w}_t, \quad \sigma_t : \mathbb{R} \to \mathbb{R}, \tag{12}$$

with a pre-defined increasing noise schedule σ_t. Lemma 4 guarantees the applicability of our conclusions to any other types of linear diffusion processes, including the typical flow matching-based models (Albergo *et al* 2023, Lipman *et al* 2023, Liu *et al* 2023b). In this case, the **sampling trajectory** is denoted as $\{\hat{\mathbf{x}}_{t_n}\}_{n=0}^{N}$ with the time schedule $\Gamma = \{t_0 = \epsilon, \cdots, t_N = T\}$ and the initial noise is denoted as $\hat{\mathbf{x}}_{t_N} \sim p_{\mathrm{n}} = \mathcal{N}(0, \sigma_{\mathrm{T}}^2 I)$.

2.4. Conditional and latent diffusion models

It is straightforward to extend the above framework of unconditional diffusion models into the conditional variants (Dhariwal and Nichol 2021, Song *et al* 2021c, Rombach *et al* 2022). Given the class or text-based condition $\mathbf{c}$, the modeled marginal distributions become $p_t(\mathbf{x}_t|\mathbf{c})$ (or $p_t(\mathbf{z}_t|\mathbf{c})$ in the $\mathbf{z}$-space) instead of the original $p_t(\mathbf{x}_t)$, and the sampling process relies on the learned conditional score $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{c})$ at each time. In general, discrete texts are first mapped into a continuous text embedding space (Nichol

https://doi.org/10.1088/1742-5468/ae17ac

et al 2022, Rombach *et al* 2022, Saharia *et al* 2022), which distinguishes text-conditional diffusion models from the diffusion models conditioned on discrete class labels. Another extension from the practical consideration is performing the diffusion process in a low-dimensional latent space rather than the original high-dimensional data space (Vahdat *et al* 2021, Rombach *et al* 2022). With the help of an autoencoder structure, latent diffusion models significantly reduce computational demand and scale up to high-resolution generation.

In the following empirical analysis (section 3), we will demonstrate that strong trajectory regularity is widely present in *unconditional*, *class-conditional*, and *text-conditional* diffusion models. This observation motivates us to investigate the underlying mechanism behind (section 4) and to develop an improved algorithm for sampling acceleration (section 5).

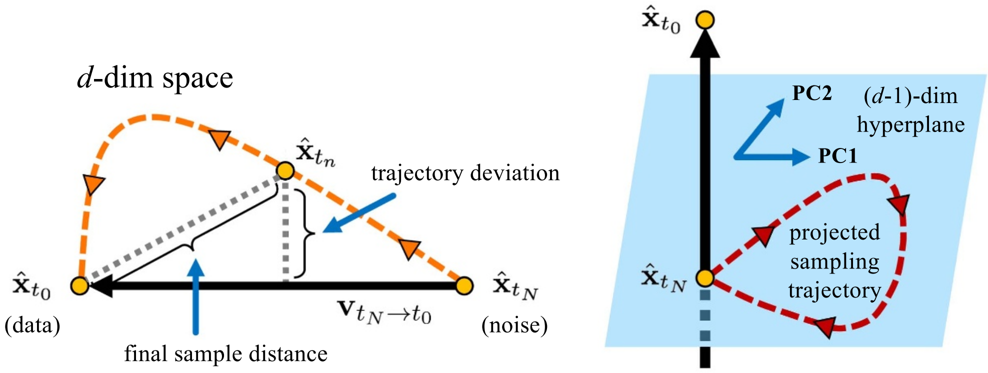
3. Geometric regularity in deterministic sampling dynamics

As mentioned in section 1, the sampling trajectories of diffusion-based generative models under the PF-ODE framework exhibit a certain regularity in their shapes, regardless of the specific content generated. To better demonstrate this concept, we undertake a series of empirical studies in this section, covering unconditional generation (pixel space) on CIFAR-10 (Krizhevsky and Hinton 2009), class-conditional generation (pixel space) on ImageNet (Russakovsky *et al* 2015), and text-conditional generation (latent space) with Stable Diffusion v1.5 (Rombach *et al* 2022). The spatial resolutions used for these diffusion processes are 32×32, 64×64, 64×64, respectively. Given the complexity of visualizing the entire sampling trajectory and analyzing its geometric characteristics in the original high-dimensional space, we develop subspace projection techniques to better capture the intrinsic structure of diffusion models.

3.1. One-dimensional projection

We first examine the *trajectory deviation* from the straight line connecting the two endpoints, which serves to assess the linearity of the sampling trajectory. A sketch of this computation is provided in figure 2(a). This approach allows us to align and collectively observe the general behaviors of all trajectories. Specifically, we denote the *displacement vector* between the two endpoints as $\mathbf{v}_{t_N\to t_0} := \hat{\mathbf{x}}_{t_0} - \hat{\mathbf{x}}_{t_N}$, and compute the trajectory deviation as the perpendicular Euclidean distance (L^2) from each intermediate sample $\hat{\mathbf{x}}_{t_n}$ to the vector $\mathbf{v}_{t_N\to t_0}$, i.e. $d_{\mathrm{td}} := \sqrt{\|\mathbf{v}_{t_n\to t_0}\|_2^2 - \left(\mathbf{v}_{t_n\to t_0}^T \cdot \mathbf{v}_{t_N\to t_0}/\|\mathbf{v}_{t_N\to t_0}\|_2\right)^2}$, Additionally, we calculate the L^2 distance between each intermediate sample $\hat{\mathbf{x}}_{t_n}$ and the final sample $\hat{\mathbf{x}}_{t_0}$, denoted as $d_{\mathrm{fsd}} := \|\hat{\mathbf{x}}_{t_n} - \hat{\mathbf{x}}_{t_0}\|_2$, and refer to it as the *final sample distance*.

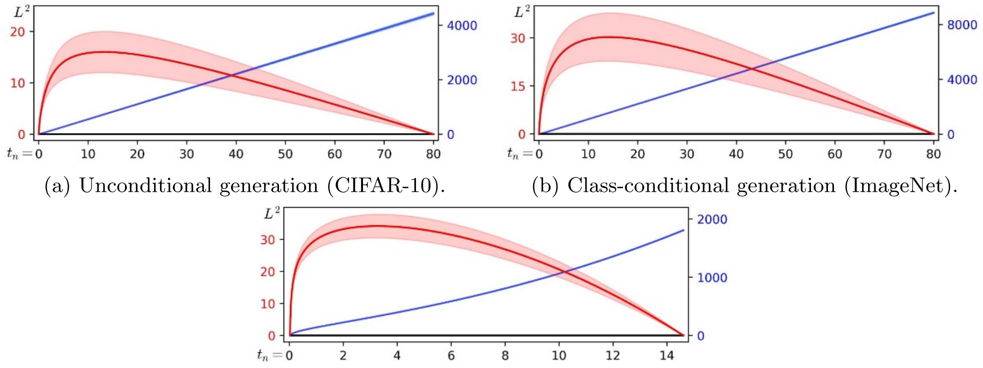
The empirical results of trajectory deviation d_{td} and final sample distance d_{fsd} are depicted as the red curves and blue curves in figure 3, respectively. Note that we use *sampling time as the horizontal axis*, which allows all sampling trajectories to be aligned and compared both within and across different time slices. From figures 3(a) and (b), we observe that the sampling trajectory's deviation gradually increases from $t = 80$



(a) 1-D reconstruction.

(b) 3-D reconstruction.

Figure 2. Illustration of subspace projection techniques. The deterministic sampling trajectory begins with an initial noise $\hat{\mathbf{x}}_{t_N}$ and progresses to the synthesized data $\hat{\mathbf{x}}_{t_0}$. (a) The trajectory deviation equals the reconstruction error when the d-dimensional point of the sampling trajectory is projected onto the displacement vector $\mathbf{v}_{t_N \to t_0} := \hat{\mathbf{x}}_{t_0} - \hat{\mathbf{x}}_{t_N}$. (b) We adopt $\mathbf{v}_{t_N \to t_0}$ and several top principal components (PCs) from its $(d-1)$-dimensional orthogonal complement to approximate the original d-dimensional sampling trajectory.



(a) Unconditional generation (CIFAR-10).

(b) Class-conditional generation (ImageNet).

(c) Text-conditional generation (SDv1.5).

Figure 3. Results of the 1-D trajectory projection. The sampling trajectory exhibits an extremely small trajectory deviation (red curve) compared to the final sample distance (blue curve) in the sampling process. Each trajectory is simulated with the Euler method and 100 number of function evaluations (NFEs). The reported average and standard deviations are based on 5000 randomly generated sampling trajectories, considering variations in initial noises, class labels, and text prompts.

to approximately $t = 10$, then swiftly diminishes as it approaches the final samples. This pattern suggests that initially, each sample might be influenced by various modes, experiencing significant impact, but later becomes strongly guided by its specific mode after a certain turning point. This behavior supports the heuristic approach of arranging time intervals more densely near the minimum timestamp and sparsely towards the maximum one (Song *et al* 2021a, 2023, Karras *et al* 2022, Chen *et al* 2024). However, when we consider the ratio of the maximum deviation to the endpoint distance in figure 3(a) and (b), we find that the trajectory deviation is remarkably slight (e.g. $30/8800 \approx 0.0034$ for ImageNet), indicating a pronounced straightness. Additionally, the generated samples along the sampling trajectory tend to move monotonically from their initial points toward their final points (as illustrated by the blue curves). Similar results can be found for the text-conditional generation in the latent space, as shown in figure 3(c).

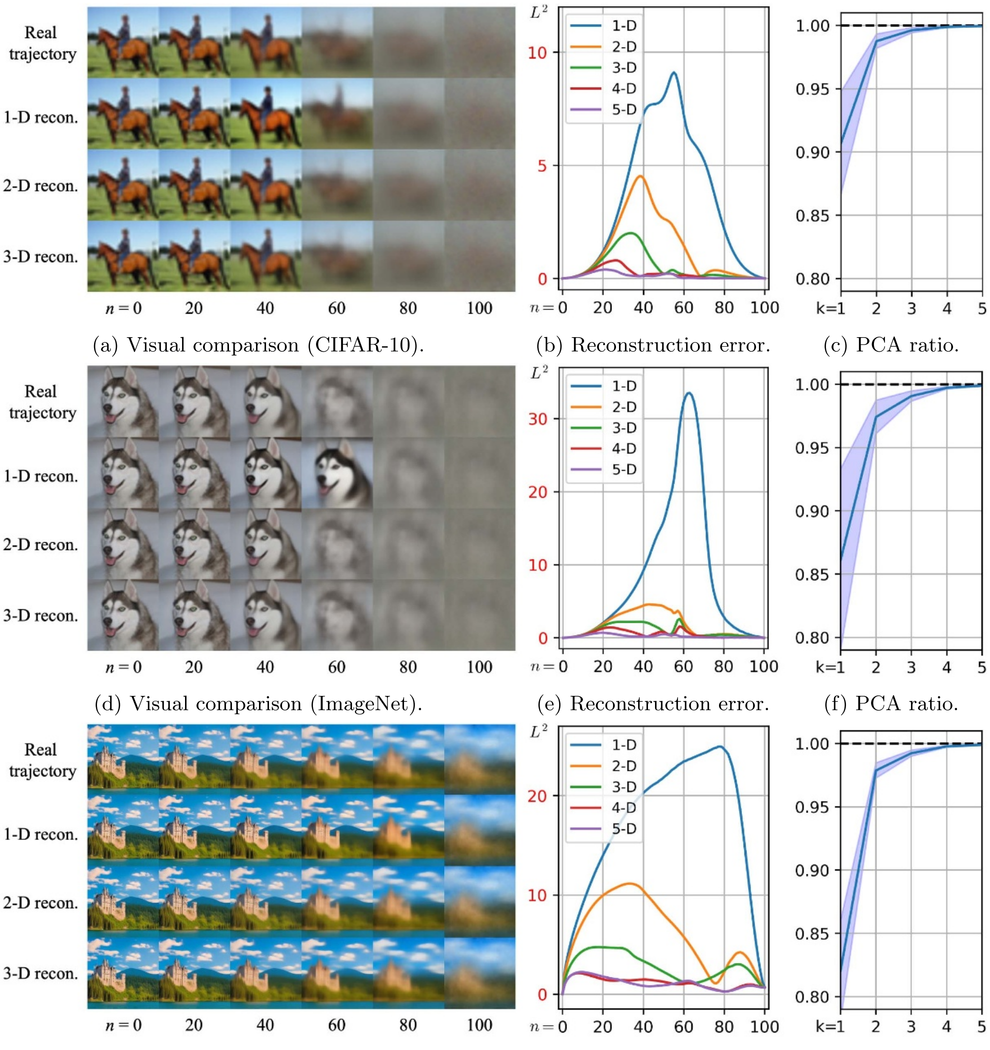
The trajectory deviation also reflects the reconstruction error if we project all d-dimensional points of the sampling trajectory onto the displacement vector $\mathbf{v}_{t_N \to t_0}$. As demonstrated in figure 4, the one-dimensional (1-D) approximation proves inadequate, leading to a significant deviation from the actual trajectory both in terms of visual comparison and quantitative results. These observations imply that while all trajectories share a similar macro-structure, the 1-D projection cannot accurately capture the full trajectory structure, probably due to the failure of modeling rotational properties. Therefore, we further develop a multi-dimensional subspace projection technique, as detailed below.

3.2. Multiple-dimensional projections

We then implement principal component analysis (PCA) on the orthogonal complement of the displacement vector $\mathbf{v}_{t_N \to t_0}$, which assists in assessing rotational properties of the sampling trajectory. A sketch of this computation is provided in figure 2(b). This $(d-1)$-D orthogonal space relative to $\mathbf{v}_{t_N \to t_0}$ is denoted as $\mathcal{V} = \{\mathbf{u} : \mathbf{u}^{\mathrm{T}} \mathbf{v}_{t_N \to t_0} = 0, \forall \mathbf{u} \in \mathbb{R}^d\}$. We begin by projecting each d-D sampling trajectory into $\mathcal{V}$, followed by conducting PCA.

As illustrated in figure 4, the 2D approximation using $\mathbf{v}_{t_N \to t_0}$ and the first principal component markedly narrows the visual discrepancy with the real trajectory, thereby reducing the L^2 reconstruction error. This finding suggests that all points in each d-D sampling trajectory diverge slightly from a 2D plane. Consequently, the tangent and normal vectors of the sampling trajectory can be effectively characterized in this manner. By incorporating an additional principal component, we enhance our ability to capture the torsion of the sampling trajectory, thereby increasing the total explained variance to approximately 85% (figures 4(c), (f) and (i)). This improvement allows for a more accurate approximation of the actual trajectory and further reduces the L^2 reconstruction error (figures 4(b), (e) and (h)). In practical terms, this level of approximation effectively captures all the visually pertinent information, with the deviation from the real trajectory being nearly indistinguishable (figures 4(a), (d) and (g)). Consequently, we can confidently utilize a 3D subspace, formed by two principal components and the displacement vector $\mathbf{v}_{t_N \to t_0}$, to understand the geometric structure of high-dimensional sampling trajectories.

Expanding on this understanding, in figure 5, we present a visualization of randomly selected sampling trajectories created by diffusion models under various generation settings. Note that the scale along the axis corresponding to $\hat{\mathbf{x}}_{t_0} - \hat{\mathbf{x}}_{t_N}$ is orders of magnitude larger than those of the other two principal components. Since we focus



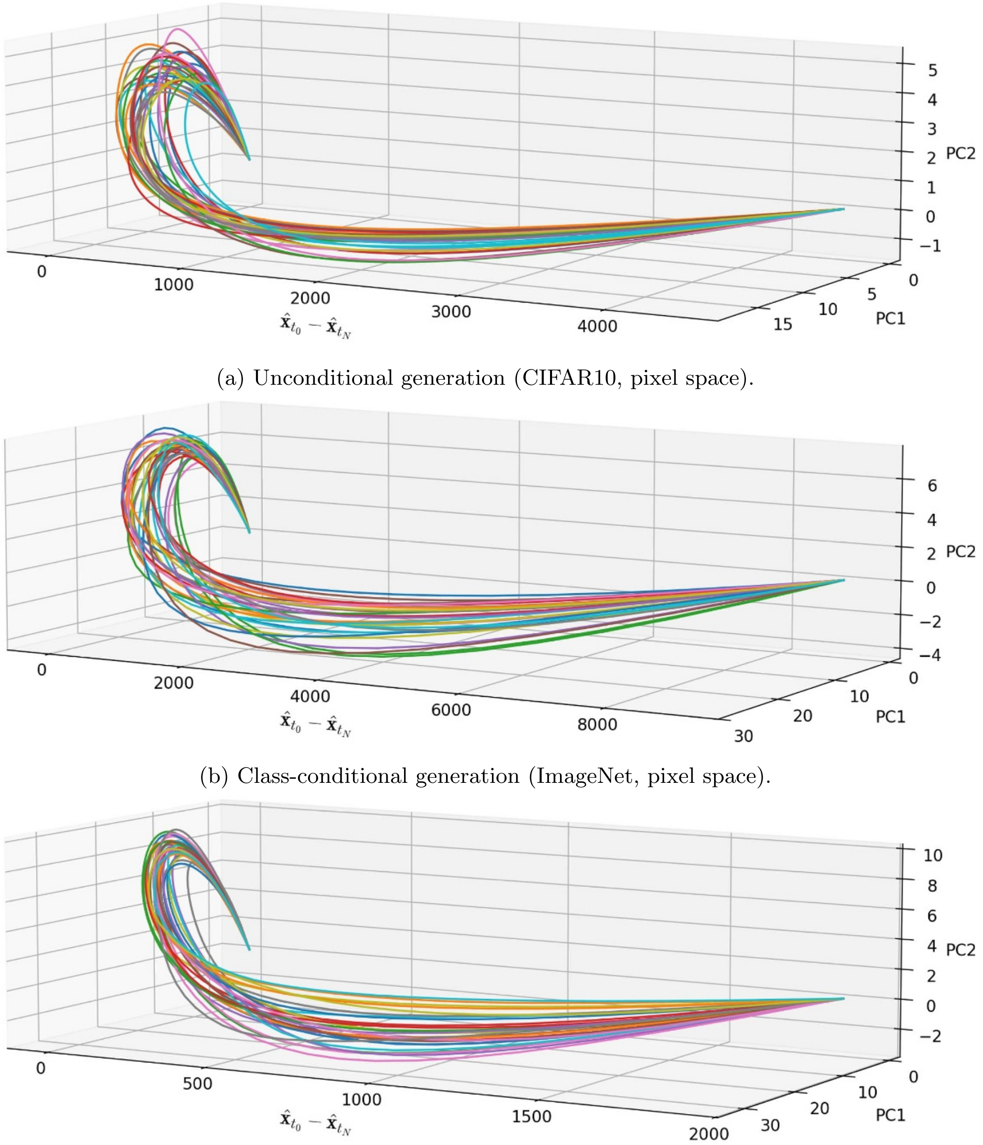
(a) Visual comparison (CIFAR-10). (b) Reconstruction error. (c) PCA ratio.

(d) Visual comparison (ImageNet). (e) Reconstruction error. (f) PCA ratio.

(g) Visual comparison (SDv1.5). (h) Reconstruction error. (i) PCA ratio.

Figure 4. Visual comparison of trajectory reconstruction for (a) unconditional, (d) class-conditional (generated by EDM (Karras *et al* 2022)), and (g) text-conditional generation (generated by SDv1.5 (Rombach *et al* 2022)). The real sampling trajectories (top row) are reconstructed using $\mathbf{v}_{t_N \to t_0}$ (1-D recon.), along with their top 1 or 2 principal components (2D or 3D recon.). To amplify visual differences, we present the denoising outputs of these trajectories. (b/e/h) The L^2 distance between the real trajectory samples and their reconstructed counterparts is computed up to 5D reconstruction. (c/f/i) The variance explained by the top k principal components is reported as the ratio of the sum of the top k eigenvalues to the sum of all eigenvalues.

on the geometric shape regularity of sampling trajectories rather than their absolute locations, we align all trajectories via orthogonal transformations to eliminate arbitrary orientation variations. These transformations, including rotations and reflections, are determined by solving the classic *Orthogonal Procrustes Problem* (Hurley and Cattell



(a) Unconditional generation (CIFAR10, pixel space).

(b) Class-conditional generation (ImageNet, pixel space).

(c) Text-conditional generation (SDv1.5, latent space).

Figure 5. We project 30 sampling trajectories generated by (a) unconditional, (b) class-conditional, and (c) text-conditional diffusion models into the 3D subspaces. Each trajectory is simulated with the Euler method and 100 number of function evaluations (NFEs). These trajectories are first aligned to the direction of the displacement vector $\hat{\mathbf{x}}_{t_0} - \hat{\mathbf{x}}_{t_N}$ (this direction is slightly different for each sample), and then projected to the top 2 principal components in the orthogonal space to $\hat{\mathbf{x}}_{t_0} - \hat{\mathbf{x}}_{t_N}$. See texts for more details.

1962, Schönemann 1966, Golub and Van Loan 2013), after which we visualize the calibrated trajectories. Specifically, we represent each projected sampling trajectory in the 3D subspace as a matrix $\mathbf{R} \in \mathbb{R}^{N\times 3}$, where each row corresponds to a sample coordinate $\mathbf{r} \in \mathbb{R}^3$ at a particular time step, and N is the total number of generated samples in the trajectory. Then, we seek an orthogonal transformation $\mathbf{O}\in\mathbb{R}^{3\times 3}$ to minimize the Frobenius norm of the residual error matrix $\mathbf{E}$ when aligning one matrix $\mathbf{R}\in\mathbb{R}^{N\times 3}$ with a reference projected trajectory matrix $\widetilde{\mathbf{R}}\in\mathbb{R}^{N\times 3}$ of the same dimensions. This optimization problem is formulated as

$$\mathbf{E}=\min_{\mathbf{O}}\|\widetilde{\mathbf{R}}-\mathbf{R}\mathbf{O}\|_{\mathrm{F}}^2,\quad s.t.\quad \mathbf{O}^{\mathrm{T}}\mathbf{O}=\mathbf{I}_{3\times 3}. \tag{13}$$

The optimal solution can be derived using the method of Lagrange multipliers, yielding $\mathbf{O}^{\star}=\mathbf{U}\mathbf{V}^{\mathrm{T}}$, where $\mathbf{U}$ and $\mathbf{V}$ are obtained via singular value decomposition (SVD) of $\mathbf{R}^{\mathrm{T}}\widetilde{\mathbf{R}}$, i.e. $\mathbf{R}^{\mathrm{T}}\widetilde{\mathbf{R}}=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}}$. As shown in figure 3, we adopt *sampling time as the horizontal axis* to better observe the trajectory shapes by aligning different trajectories across time slices. Here, time is scaled by a factor of $\sqrt{d}$ to preserve relative magnitudes between axes. In other words, the visualization remains almost unchanged whether we use the displacement vector $\mathbf{v}_{t_N\to t_0}$ or scaled time as the first axis. Moreover, in this case, an orthogonal transformation $\widetilde{\mathbf{O}}\in\mathbb{R}^{2\times 2}$ that fixes the first axis suffices for alignment.

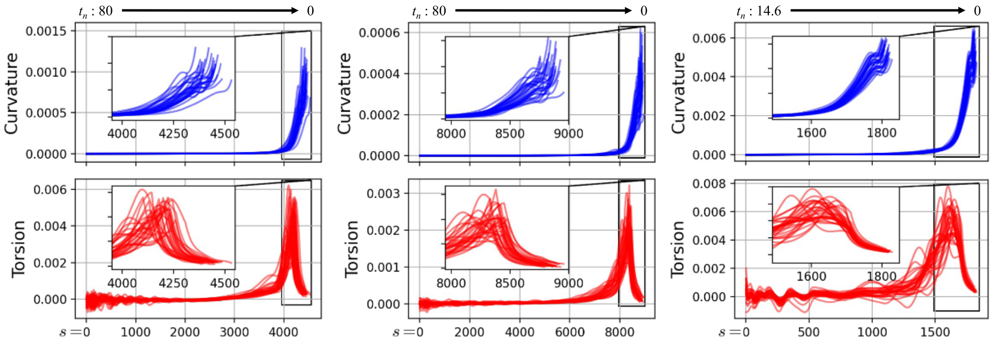
As a result, the calibrated trajectories depicted in figure 5 largely adhere to the straight line connecting their endpoints, corroborating the small trajectory deviation observed in our previous findings (figure 3). Furthermore, figure 5 accurately depicts the sampling trajectory's behavior, showing its gradual departure from the osculating plane during sampling. Interestingly, each trajectory consistently exhibits a simple, approximately *linear-nonlinear* structure. This reveals a strong regularity in all sampling trajectories, independent of the specific content generated and variations in initial noises, class labels, and text prompts.

3.3. Three-dimensional projection revisited

Given the strong trajectory regularity of deterministic diffusion sampling manifested in the three-dimensional Euclidean space, as shown in figure 5, we further resort to a differential geometry tool known as the *Frenet–Serret formulas* (Do Carmo 2016) to precisely characterize geometric properties of the projected sampling trajectory.

We denote the projected sampling trajectory consisting of N discrete points in the 3D subspace as $\mathbf{r}(\xi)$, where $\xi=T-t$ with T as the terminal time and t as the sampling time (see section 2 for detailed notations). Thanks to our proposed subspace projection techniques (sections 3.1 and 3.2), each projected sampling trajectory keeps starting from pure noise $\mathbf{r}(0)$ and ends at the synthesized data $\mathbf{r}(T)$. The *arc-length* of this spatial curve is denoted as $s(\xi)=\int_0^{\xi}\|\mathbf{r}'(u)\|\mathrm{d}u$, which is a strictly monotone increasing function. We then use the arc-length s to parameterize the spatial curve, and define the *tangent unit vector*, *normal unit vector*, and *binormal unit vector* as $\mathbf{T}(s):=\mathbf{r}'(s)$, $\mathbf{N}(s):=\mathbf{r}''(s)/\|\mathbf{r}''(s)\|$, and $\mathbf{B}(s):=\mathbf{T}(s)\times\mathbf{N}(s)$, respectively. These three unit vectors are interrelated, and their relationship is characterized by the well-known Frenet–Serret formulas listed below,

$$\frac{\mathrm{d}\mathbf{T}(s)}{\mathrm{d}s}=\kappa(s)\mathbf{N}(s),\quad \frac{\mathrm{d}\mathbf{N}(s)}{\mathrm{d}s}=-\kappa(s)\mathbf{T}(s)+\tau(s)\mathbf{B}(s),\quad \frac{\mathrm{d}\mathbf{B}(s)}{\mathrm{d}s}=-\tau(s)\mathbf{N}(s), \tag{14}$$



(a) Unconditional generation (CIFAR-10), window size = 101.

(b) Class-conditional generation (ImageNet), window size = 101.

(c) Text-conditional generation (SDv1.5), window size = 151.

Figure 6. We compute the curvature and torsion functions of each 3D spatial curve that approximates the original high-dimensional sampling trajectory. The Euler method with 1000 NFEs is employed to faithfully simulate 30 sampling trajectories with (a) unconditional, (b) class-conditional, and (c) text-conditional diffusion models. The geometric properties of the curves are then estimated using least-squares fitting.

where the curvature $\kappa(s)$ and the torsion $\tau(s)$ measure the deviations of a spatial curve $\mathbf{r}(s)$ from being a straight line and from being planar, respectively. In practice, we generally employ the following expressions for numerical calculation[6].

$$\kappa(s) = [\mathbf{r}'(s) \times \mathbf{r}''(s)] / \|\mathbf{r}'(s)\|^3, \quad \tau(s) = [(\mathbf{r}'(s) \times \mathbf{r}''(s)) \cdot \mathbf{r}'''(s)] / \|\mathbf{r}'(s) \times \mathbf{r}''(s)\|^2. \tag{15}$$

Specifically, for each discrete point on the spatial curve, we employ its surrounding points within a given window size to estimate the first-, second-, and third-order derivatives based on the third-order Taylor expansion (Lewiner *et al* 2005). With the help of least squares fitting using an appropriate window size, the torsion and curvature functions of each projected sampling trajectory can be reliably estimated with small reconstruction errors.

As shown in figure 6, the projected sampling trajectory remains nearly straight for a large portion ($\approx 80\%$) of the entire sampling process, with both curvature and torsion staying close to zero, consistent with the analysis in sections 3.1 and 3.2. For example, in class-conditional generation (figure 6(b)), the curvature and torsion gradually increase once the arc-length s exceeds around 7000. The torsion reaches its peak at around 8250, corresponding to a sampling time of around four, and then decreases back toward zero. Note that figure 6 serves as an important complement to figure 5, providing a more faithful description of the rotational structure of the 3D spatial curve throughout the sampling process. In contrast, the significant differences in axis magnitudes in figure 5 may visually distort the actual trajectory shape. Furthermore, according to the *fundamental theorem of curves* in differential geometry (Do Carmo 2016), the shape of any regular curve with non-zero curvature in 3D space is fully determined by its curvature

[6] These two equations also hold for the spatial curve $\mathbf{r}(\xi)$ parametrized by ξ.

and torsion. The highly similar evolution patterns of curvature and torsion observed in figure 6 provides strong evidence of trajectory regularity, suggesting that sampling trajectories are congruent across different diffusion models and generation conditions.

4. Understanding the deterministic sampling dynamics

In this section, we investigate several properties of deterministic sampling in diffusion models and provide explanations for the trajectory regularity observed in the previous section. We begin by noting that, beyond the explicit sampling trajectory, there exists an additional but often overlooked trajectory that determines each intermediate sampling point through a convex combination (section 4.1). Under the empirical data distribution, we establish a connection between the closed-form solutions of these trajectories and the classic mean-shift algorithm (section 4.2). We then present a detailed theoretical analysis of the sampling dynamics, revealing stepwise rotation and monotone likelihood increase as local behaviors, and characterizing the overall trajectory as a linear–nonlinear–linear mode-seeking path of approximately constant length as a global behavior. Finally, we revisit trajectory regularity under the Gaussian data assumption (section 4.3).

4.1. Implicit denoising trajectory

Given a parametric diffusion model with the *denoising output* $r_{\boldsymbol{\theta}}(\cdot;\cdot)$, the sampling trajectory is simulated by numerically solving the empirical PF-ODE (11). Meanwhile, an implicitly coupled sequence $\{r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}, t_n)\}_{n=1}^{N}$ is formed as a by-product. We designate this sequence, or simplified to $\{r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})\}_{n=1}^{N}$ if there is no ambiguity, as the *denoising trajectory*. We then rearrange the empirical PF-ODE (11) as $r_{\boldsymbol{\theta}}(\mathbf{x}_t;t) = \mathbf{x}_t - \sigma_t(\mathrm{d}\mathbf{x}_t/\mathrm{d}\sigma_t)$, and take the derivative of both sides to obtain the differential equation of the denoising trajectory

$$\mathrm{d}r_{\boldsymbol{\theta}}\left(\mathbf{x}_t;t\right)/\mathrm{d}\sigma_t = -\sigma_t\left[\mathrm{d}^2\mathbf{x}_t/\mathrm{d}\sigma_t^2\right]. \tag{16}$$

This equation, although not directly applicable for simulation, reveals that the denoising trajectory encapsulates the curvature information of the associated sampling trajectory. The following proposition reveals how these two trajectories are inherently related.

Proposition 1. *Given the probability flow ODE (11) and a current position $\hat{\mathbf{x}}_{t_{n+1}}$, $n \in [0, N-1]$ in the sampling trajectory, the next position $\hat{\mathbf{x}}_{t_n}$ predicted by a k-th order Taylor expansion with the time step size $\sigma_{t_{n+1}} - \sigma_{t_n}$ is*

$$\hat{\mathbf{x}}_{t_n} = \frac{\sigma_{t_n}}{\sigma_{t_{n+1}}}\hat{\mathbf{x}}_{t_{n+1}} + \frac{\sigma_{t_{n+1}} - \sigma_{t_n}}{\sigma_{t_{n+1}}}\mathcal{R}_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_{n+1}}\right), \tag{17}$$

which is a convex combination of $\hat{\mathbf{x}}_{t_{n+1}}$ and the generalized denoising output

$$\mathcal{R}_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_{n+1}}\right) = r_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_{n+1}}\right) - \sum_{i=2}^{k}\frac{1}{i!}\frac{\mathrm{d}^{(i)}\mathbf{x}_t}{\mathrm{d}\sigma_t^{(i)}}\Big|_{\hat{\mathbf{x}}_{t_{n+1}}}\sigma_{t_{n+1}}\left(\sigma_{t_n} - \sigma_{t_{n+1}}\right)^{i-1}. \tag{18}$$

https://doi.org/10.1088/1742-5468/ae17ac

In particular, we have $\mathcal{R}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}}) = r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}})$ *for the Euler method (*$k=1$*), and* $\mathcal{R}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}}) = r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}}) + \frac{\sigma_{t_n}-\sigma_{t_{n+1}}}{2}\frac{\mathrm{d}r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}})}{\mathrm{d}\sigma_t}$ *for second-order numerical methods (*$k=2$*).*

Corollary 1. *The denoising output* $r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}})$ *reflects the prediction made by a single Euler step from* $\hat{\mathbf{x}}_{t_{n+1}}$ *with the time step size* $\sigma_{t_{n+1}}$*.*

Corollary 2. *Each second-order ODE-based accelerated sampling method corresponds to a specific first-order finite difference of* $\mathrm{d}r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}})/\mathrm{d}\sigma_t$*.*

The ratio $\sigma_{t_n}/\sigma_{t_{n+1}}$ in (17) quantifies the relative preference for maintaining the current position versus transitioning to the generalized denoising output at t_{n+1}. Since the denoising output starts from a spurious mode and gradually converges toward a true mode, a reasonable strategy is to decrease this weight progressively during sampling. From this perspective, different time-scheduling functions designed for diffusion sampling, such as uniform, quadratic, or polynomial schedules (Song *et al* 2021a, Karras *et al* 2022, Lu *et al* 2022a, Chen *et al* 2024), essentially represent various weighting functions. This interpretation also motivates the direct search for proper weights to further enhance visual quality (section 5).

We primarily focus on the Euler method to simplify subsequent discussions, though these insights can be readily extended to higher-order methods. The trajectory behavior in the continuous-time scenario is similarly discernible through examining the sampling process with an infinitesimally small Euler step.

4.2. Theoretical analysis of the trajectory structure

In this section, we leverage the well-established closed-form solutions under the empirical data distribution (Karras *et al* 2022, Scarvelis *et al* 2023, Chen *et al* 2023a) to connect deterministic sampling dynamics with the classic mean-shift algorithm (Fukunaga and Hostetler 1975, Cheng 1995, Comaniciu and Meer 2002), and characterize their *local* and *global* behaviors.

As discussed in section 2.2, once a diffusion model converges to the optimum, i.e. $\forall t, r_{\boldsymbol{\theta}}(\mathbf{x}_t;t) \to r^{\star}_{\boldsymbol{\theta}}(\mathbf{x}_t;t) = \mathbb{E}(\mathbf{x}_0|\mathbf{x}_t)$, it captures the score $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)$ across different noise levels. The exact formula for the *optimal denoising output* is given in (9), under which both the sampling trajectory and the denoising trajectory admit closed-form solutions. In this case, the marginal density at each time step of the forward diffusion process (12) becomes a Gaussian KDE with bandwidth σ_t^2, i.e. $\hat{p}_t(\mathbf{x}_t) = \int p_{0t}(\mathbf{x}_t|\mathbf{y})\hat{p}_d(\mathbf{y}) = (1/|\mathcal{I}|)\sum_{i\in\mathcal{I}}\mathcal{N}\left(\mathbf{x}_t;\mathbf{y}_i,\sigma_t^2\mathbf{I}\right)$. Intuitively, the forward process can be viewed as an expansion in both magnitude and manifold: the training data samples leave the original small-magnitude low-rank manifold and spread onto a large-magnitude high-rank manifold. Consequently, the squared magnitude of a noisy sample is expected to exceed that of the original sample. As the dimension $d\to\infty$, this expansion occurs with probability one and the isotropic Gaussian noise becomes approximately uniformly distributed on the sphere (Vershynin 2018, Chen *et al* 2023a). In contrast, the backward process exhibits the opposite trend due to marginal preservation.

In particular, the closed-form solution (9) is highly reminiscent of the iterative formula used in mean shift (Fukunaga and Hostetler 1975, Cheng 1995, Comaniciu and Meer 2002, Yamasaki and Tanaka 2020). Mean shift is a non-parametric algorithm designed to locate modes of a density function, typically a KDE, via iterative gradient ascent with adaptive step sizes. Given a current position $\mathbf{x}$, mean shift with a Gaussian kernel and bandwidth h iteratively adds the vector $\mathbf{m}(\mathbf{x})-\mathbf{x}$, which points toward the direction of maximum increase of the KDE $p_h(\mathbf{x}) = (1/|\mathcal{I}|)\sum_{i=1}^{|\mathcal{I}|}\mathcal{N}(\mathbf{x};\mathbf{y}_i, h^2\mathbf{I})$, to itself, i.e. $\mathbf{x} \leftarrow [\mathbf{m}(\mathbf{x})-\mathbf{x}]+\mathbf{x}$. The *mean vector* is

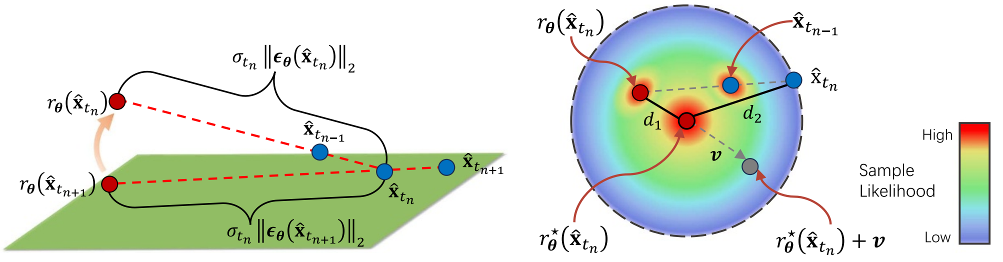
$$\mathbf{m}(\mathbf{x}, h) = \sum_i \frac{\exp\left(-\|\mathbf{x}-\mathbf{y}_i\|_2^2/2h^2\right)}{\sum_j \exp\left(-\|\mathbf{x}-\mathbf{y}_j\|_2^2/2h^2\right)}\mathbf{y}_i. \tag{19}$$

As a mode-seeking algorithm, mean shift has been particularly successful in clustering (Cheng 1995, Carreira-Perpinán 2015), image segmentation (Comaniciu and Meer 1999, 2002), and video tracking (Comaniciu *et al* 2000, 2003). By identifying the bandwidth σ_t in (9) with the bandwidth h in (19), we build a connection between the optimal denoising output of a diffusion model and annealed mean shift under the KDE-based data modeling. Moreover, the time-decreasing bandwidth ($\sigma_t \to 0$ as $t \to 0$) in (9) strongly parallels *annealed mean shift*, or *multi-bandwidth mean shift* (Shen *et al* 2005), a metaheuristic algorithm designed to escape local maxima where classical mean shift is susceptible to stuck, by monotonically decreasing the bandwidth in iterations. Analogous to (17), each Euler step in the optimal case equals a convex combination of the annealed mean vector and the current position, with the PF-ODE $\mathrm{d}\mathbf{x}_t/\mathrm{d}\sigma_t = (\mathbf{x}_t - r^\star_{\boldsymbol{\theta}}(\mathbf{x}_t;t))/\sigma_t = \boldsymbol{\epsilon}^\star_{\boldsymbol{\theta}}(\mathbf{x}_t;t)$.

The above analysis further implies that all ODE trajectories generated by an optimal diffusion model are uniquely determined and governed by a bandwidth-varying mean shift. In this setting, both the forward (encoding) process and backward (decoding) process depend solely on the data distribution and the noise distribution, independent of the model architecture or optimization algorithm. This property, previously referred to as uniquely identifiable encoding and empirically verified in (Song *et al* 2021c), is here shown to be theoretically connected to a global KDE-based mode-seeking algorithm (Shen *et al* 2005), and thus reveals the asymptotic sampling behavior of diffusion models as training converges to the optimum. Although optimal diffusion models essentially memorize the dataset and replay discrete data points during sampling, we argue that in practice, a slight score deviation from the optimum both preserves generative ability and substantially mitigates mode collapse (see appendix C.2).

4.2.1. Local properties. We first show that (1) the denoising output governs the rotation of the sampling trajectory, and (2) each sampling trajectory converges monotonically in terms of sample likelihood, with its coupled denoising trajectory consistently achieving higher likelihood.

Figure 7(a) illustrates two successive Euler steps according to proposition 1. The direction of the sampling trajectory (depicted as a polygonal chain with consecutive blue vertices) is controlled by the denoising outputs, while the vertex locations depend on the time schedule. In the optimal case, the sampling path follows a similar structure with the PF-ODE $\mathrm{d}\mathbf{x}_t = \boldsymbol{\epsilon}^\star_{\boldsymbol{\theta}}(\mathbf{x}_t;t)\mathrm{d}\sigma_t$. This equation defines a special vector field featuring an approximately constant magnitude $\|\boldsymbol{\epsilon}^\star_{\boldsymbol{\theta}}(\mathbf{x}_t;t)\|_2$ across all marginal distributions $p_t(\mathbf{x}_t)$.


$\sigma_{t_n}\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})\|_2$
$r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})$
$\hat{\mathbf{x}}_{t_{n-1}}$
$r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}})$
$\hat{\mathbf{x}}_{t_n}$
$\hat{\mathbf{x}}_{t_{n+1}}$
$\sigma_{t_n}\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}})\|_2$
$r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})$
$\hat{\mathbf{x}}_{t_{n-1}}$
$\hat{\mathbf{x}}_{t_n}$
d_1
d_2
$\boldsymbol{v}$
$r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})$
$r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})+\boldsymbol{v}$
High
Sample Likelihood
Low

(a) Stepwise rotation.

(b) Monotone likelihood increasing.

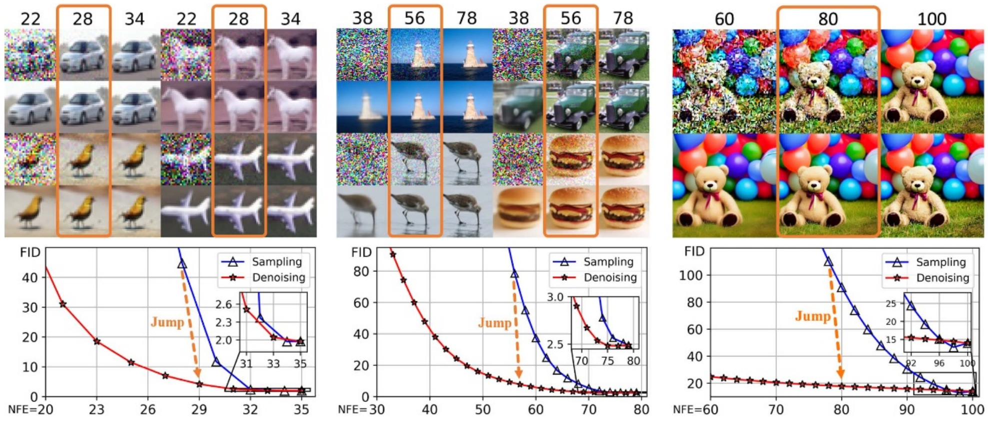
Figure 7. (a) An illustration of two consecutive Euler steps, starting from a current sample $\hat{\mathbf{x}}_{t_{n+1}}$. A single Euler step in the ODE-based sampling is a convex combination of the denoising output and the current position to determine the next position. Blue points form a piecewise linear sampling trajectory, while red points form the denoising trajectory governing the rotation direction. (b) We have three likelihood orders in the ODE-based diffusion sampling: (1) $p_h(r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})) \geqslant p_h(\hat{\mathbf{x}}_{t_n})$, (2) $p_h(\hat{\mathbf{x}}_{t_{n-1}}) \geqslant p_h(\hat{\mathbf{x}}_{t_n})$, and (3) $p_h(r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})) \geqslant p_h(\hat{\mathbf{x}}_{t_n})$. Note that $p_h(r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}))$ may not possess the highest likelihood within the sphere.

Proposition 2. *The magnitude of $\boldsymbol{\epsilon}^\star_{\boldsymbol{\theta}}(\mathbf{x}_t;t)$ concentrates around $\sqrt{d}$, where d denotes the data dimension. Consequently, the total length of the sampling trajectory is approximately $\sigma_T\sqrt{d}$, where σ_T denotes the maximum noise level.*

The above results also hold for practical diffusion models, with proofs and supporting empirical evidence provided in appendix B.6. We further deduce that the position of each intermediate point $\hat{\mathbf{x}}_{t_n}$, $n \in [1, N-1]$ in the sampling trajectory is primarily determined by the chosen time schedule, given that $\|r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}}) - \hat{\mathbf{x}}_{t_n}\|_2 = (\sigma_{t_n}/\sigma_{t_{n+1}})\|r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}}) - \hat{\mathbf{x}}_{t_{n+1}}\|_2 = \sigma_{t_n}\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}})\|_2 \approx \sigma_{t_n}\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})\|_2 = \|r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}) - \hat{\mathbf{x}}_{t_n}\|_2$. In this scenario, the denoising output $r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}})$ appears to be oscillating toward $r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})$ around $\hat{\mathbf{x}}_{t_n}$, akin to the motion of a simple gravity pendulum (Young *et al* 1996). The pendulum length contracts by the factor $\sigma_{t_n}/\sigma_{t_{n+1}}$ at each sampling step, starting from an initial length of roughly $\sigma_T\sqrt{d}$. This specific structure is shared across all trajectories. In practice, the oscillation amplitude is extremely small ($\approx 0^\circ$), and the entire sampling trajectory remains nearly confined to a two-dimensional plane. The minor deviations can be effectively represented using a small number of orthogonal bases, as discussed in section 3.

Next, we characterize the likelihood orders in the deterministic sampling process. To simplify notations, we denote the deviation of denoising output from the optimal counterpart as $d_1(\hat{\mathbf{x}}_{t_n}) = \|r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}) - r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})\|_2$ and the distance between the optimal denoising output and the current position as $d_2(\hat{\mathbf{x}}_{t_n}) = \|r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}) - \hat{\mathbf{x}}_{t_n}\|_2$.

Proposition 3. *In deterministic sampling with the Euler method, the sample likelihood is non-decreasing, i.e. $\forall n \in [1, N]$, we have $p_h(\hat{\mathbf{x}}_{t_{n-1}}) \geqslant p_h(\hat{\mathbf{x}}_{t_n})$ and $p_h(r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})) \geqslant p_h(\hat{\mathbf{x}}_{t_n})$ with respect to the Gaussian KDE $p_h(\mathbf{x}) = (1/|\mathcal{I}|)\sum_{i\in\mathcal{I}}\mathcal{N}(\mathbf{x};\mathbf{y}_i, h^2\mathbf{I})$ for any positive bandwidth h, under the assumption that all samples in the trajectory satisfy $d_1(\hat{\mathbf{x}}_{t_n}) \leqslant d_2(\hat{\mathbf{x}}_{t_n})$.*



(a) Unconditional generation (CIFAR-10). Total NFE = 35.

(b) Class-conditional generation (ImageNet). Total NFE = 79.

(c) Text-conditional generation with SDv1.5. Total NFE = 100.

Figure 8. Comparison of visual quality (top is sampling trajectory, bottom is denoising trajectory) and Fréchet Inception Distance (FID (Heusel *et al* 2017), lower is better) *w.r.t.* the number of score function evaluations (NFEs). The denoising trajectory converges much faster than the sampling trajectory in terms of FID and visual quality. Figures (a)(b) are generated by EDM (Karras *et al* 2022), and figure (c) is generated by SDv1.5 (Rombach *et al* 2022).

A visual illustration is provided in figure 7(b). The assumption requires that the learned denoising output $r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})$ falls within a sphere centered at the optimal denoising output $r^{\star}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})$ with a radius of $d_2(\hat{\mathbf{x}}_{t_n})$. This radius controls the maximum deviation of the learned denoising output and shrinks during the sampling process. In practice, such an assumption is relatively easy to satisfy for a well-trained diffusion model. Therefore, each sampling trajectory monotonically converges in terms of sample likelihood ($p_h(\hat{\mathbf{x}}_{t_{n-1}}) \geqslant p_h(\hat{\mathbf{x}}_{t_n})$), while its coupled denoising trajectory converges even faster ($p_h(r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})) \geqslant p_h(\hat{\mathbf{x}}_{t_n})$). Given an empirical data distribution, proposition 3 applies to any marginal distribution of the forward SDE $\{p_t(\mathbf{x})\}_{t=0}^{T}$, each of which is a KDE with varying positive bandwidth t. Moreover, with an infinitesimal step size, proposition 3 naturally extends to the continuous-time version. Finally, the standard monotone convergence property of mean shift is recovered when diffusion models are trained to optimality.

Corollary 3. *We have* $p_h(\mathbf{m}(\hat{\mathbf{x}}_{t_n})) \geqslant p_h(\hat{\mathbf{x}}_{t_n})$, *when* $r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}) = r^{\star}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}) = \mathbf{m}(\hat{\mathbf{x}}_{t_n})$.

Besides the monotone increase in sample likelihood, a similar trend is observed in image quality (figure 8). Both qualitative and quantitative results show that the denoising trajectory converges significantly faster than the sampling trajectory. This observation motivates a new technique, which we termed 'ODE-Jump'. The key idea is to directly transition from *any* sample at *any* time step of the sampling trajectory to its corresponding point on the denoising trajectory, and then returns the denoising output as the final synthetic result. Specifically, instead of following the full sequence $\hat{\mathbf{x}}_{t_N} \to \cdots \to \hat{\mathbf{x}}_{t_n} \to \cdots \to \hat{\mathbf{x}}_{t_0}$, we modify it to $\hat{\mathbf{x}}_{t_N} \to \cdots \to \hat{\mathbf{x}}_{t_n} \to r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})$. This reduces the total NFE from N to $N - n + 1$, assuming one NFE per step. This technique is highly flexible and simple to implement. It only requires monitoring the visual quality of intermediate denoising samples to determine an appropriate time to terminate the remaining steps. As an example, consider the sampling process with SDv1.5 in figure 8(c). By jumping from NFE $= 80$ of the sampling trajectory to NFE $= 81$ of the denoising trajectory, we obtain a substantial improvement in FID, while producing a visually comparable result to the final sample at NFE $= 100$ with significantly fewer NFEs. Additional results are shown in figure 11. Figure 8 also highlights the insensitivity of FID to subtle differences in image quality, a limitation also noted in previous work (Kirstain *et al* 2023, Podell *et al* 2024).

4.2.2. Global properties. We then show that (1) the sampling trajectory acts as a linear–nonlinear–linear mode-seeking path, and (2) the trajectory statistics undergo a dramatic change during a short phase transition period.

In the optimal case, the denoising output, also referred to as the annealed mean vector, starts from a spurious mode (approximately the dataset mean), i.e. $r^{\star}_{\boldsymbol{\theta}}(\mathbf{x}_t;t) \approx (1/|\mathcal{I}|)\sum_{i\in\mathcal{I}}\mathbf{y}_i$ when the bandwidth σ_t is sufficiently large. Meanwhile, the sampling trajectory is initially located in an approximately uni-modal Gaussian distribution with a *linear* score function:

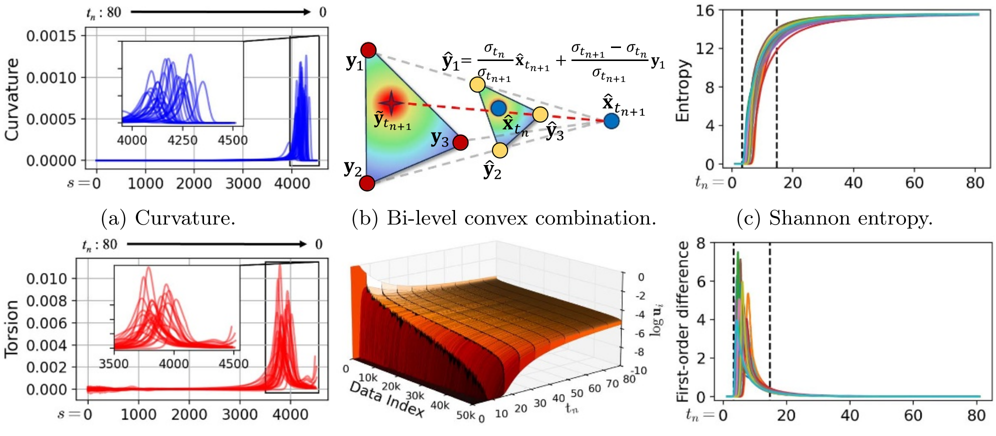
$$\text{The first linear stage:}\quad \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t) = \left(r^{\star}_{\boldsymbol{\theta}}(\mathbf{x}_t;t)-\mathbf{x}_t\right)/\sigma_t^2 \approx -\mathbf{x}_t/\sigma_t^2. \tag{20}$$

This approximation holds for large t, since the dataset mean has negligible norm relative to $\mathbf{x}_t$ by the concentration of measure (lemma 5). This justifies heuristic methods that replace the learned score with the Gaussian analytic score at the first step (Dockhorn *et al* 2022, Wang and Vastola 2024, Zhou *et al* 2024a). As σ_t monotonically decreases during sampling, the number of modes in the Gaussian KDE $\hat{p}_t(\mathbf{x}_t) = (1/|\mathcal{I}|)\sum_{i\in\mathcal{I}}\mathcal{N}\left(\mathbf{x}_t;\mathbf{y}_i,\sigma_t^2\mathbf{I}\right)$ increases (Silverman 1981), and the underlying distribution surface gradually shifts from a simple Gaussian to a complex multi-modal form. In this intermediate stage, the score function appears highly data-dependent and *nonlinear*, as multiple data points exert non-negligible influence. Finally, with a sufficiently small bandwidth σ_t, the sampling trajectory is attracted to a specific real-data mode, and the score function appears approximately *linear* again, i.e.

$$\text{The second linear stage:}\quad \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t) = \left(r^{\star}_{\boldsymbol{\theta}}(\mathbf{x}_t;t)-\mathbf{x}_t\right)/\sigma_t^2 \approx \left(\mathbf{y}_k-\mathbf{x}_t\right)/\sigma_t^2, \tag{21}$$

where $\mathbf{y}_k$ denotes the nearest data point to $\mathbf{x}_t$. In other words, the posterior distribution can again be well approximated by a Gaussian. This global linear–nonlinear–linear behavior allows the sampling trajectory to locate a true mode under mild conditions, similar to annealed mean shift (Shen *et al* 2005). Intriguingly, the total trajectory length is guaranteed to be about $\sigma_T\sqrt{d}$ (proposition 2), implying a shared structural property across trajectories originating from different initial conditions.

A straightforward piece of quantitative evidence supporting the above analysis comes from the trajectory statistics of generated sampling paths based on optimal denoising outputs, as discussed in section 3.3. The statistics presented in figures 9(a) and (d) exhibit a distinct three-stage pattern, with the first and second linear stages characterized by near-zero curvature and torsion. Critical transition points can be readily



(a) Curvature.

(b) Bi-level convex combination.

(c) Shannon entropy.

(d) Torsion.

(e) Evolution of coefficients.

(f) First-order difference.

Figure 9. Further analysis of deterministic sampling dynamics based on optimal denoising outputs, includes (a/d) the curvature and torsion functions; (b) an illustration of the bi-level convex combination used to infer the next position; (e) the evolution of convex combination coefficients; and (c/f) the corresponding Shannon entropy along the sampling trajectories.

identified by thresholding. For example, curvature values below $1e^{-5}$, which corresponds to average sampling times of 14.80 or 3.44, can be used to define linear trajectories in practice. We next delve into a more detailed analysis of the phase transition between the linear and nonlinear stages in the sampling dynamics. Given the current position $\hat{\mathbf{x}}_{t_{n+1}}$ and its corresponding optimal denoising output $r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}})$, the next position $\hat{\mathbf{x}}_{t_n}$ predicted by the Euler method according to (17) becomes

$$\begin{aligned}\hat{\mathbf{x}}_{t_n} &= \frac{\sigma_{t_n}}{\sigma_{t_{n+1}}}\hat{\mathbf{x}}_{t_{n+1}} + \frac{\sigma_{t_{n+1}}-\sigma_{t_n}}{\sigma_{t_{n+1}}} r^\star_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_{n+1}}\right)\\ &= \sum_i \underbrace{\frac{\exp\left(-\|\hat{\mathbf{x}}_{t_{n+1}}-\mathbf{y}_i\|_2^2/2\sigma_t^2\right)}{\sum_j \exp\left(-\|\hat{\mathbf{x}}_{t_{n+1}}-\mathbf{y}_j\|_2^2/2\sigma_t^2\right)}}_{\mathbf{u}_i\left(\hat{\mathbf{x}}_{t_{n+1}}\right)} \underbrace{\left(\frac{\sigma_{t_n}}{\sigma_{t_{n+1}}}\hat{\mathbf{x}}_{t_{n+1}} + \frac{\sigma_{t_{n+1}}-\sigma_{t_n}}{\sigma_{t_{n+1}}}\mathbf{y}_i\right)}_{\hat{\mathbf{y}}_i\left(\hat{\mathbf{x}}_{t_{n+1}}\right)}.\end{aligned} \tag{22}$$

This implies that $\hat{\mathbf{x}}_{t_n}$ lies within a convex hull, whose vertices $\hat{\mathbf{y}}_i$ are *convex combinations* of the current position $\hat{\mathbf{x}}_{t_{n+1}}$ and data points $\mathbf{y}_i$, with coefficients determined by the time schedule $(\sigma_{t_n}/\sigma_{t_{n+1}}, 1-\sigma_{t_n}/\sigma_{t_{n+1}})$, as illustrated in figure 9(b). In contrast, another convex combination, parameterized by coefficients $\mathbf{u}_i(\mathbf{x}_{t_{n+1}})$, quantifies the relative influence of individual data points and plays a central role in determining transition points within the linear–nonlinear–linear path. As shown in figure 9(e), where a logarithmic scale is used with a small bias term $1e^{-10}$ for numerical stability, the evolution of coefficients begins approximately uniform at $1/50\,000 \approx 10^{-4.7}$ and gradually converges toward a specific data point. Note that the influence of different data points evolves differently:

https://doi.org/10.1088/1742-5468/ae17ac

some decay monotonically, while others increase initially before declining. This behavior suggests the existence of hierarchical clusters in the dataset, potentially reflecting coarse-to-fine semantic structures in representing learning (Bengio *et al* 2013a). Moreover, there exists a non-eligible period (roughly the final 5% of the trajectory in figure 9(e)) during which one data point already dominates the trajectory's trend. This phenomenon becomes more pronounced in figures 9(c) and (f), where we introduce a Shannon entropy-based criterion, $\mathcal{H}(\mathbf{u}) = -\sum_i \mathbf{u}_i(\mathbf{x}_{t_{n+1}})\log \mathbf{u}_i(\mathbf{x}_{t_{n+1}})$, and its temporal derivative $\partial\mathcal{H}(\mathbf{u})/\partial t$ to quantify and visualize the evolving influence of data points throughout the sampling dynamics. Similar three distinct dynamical regimes and phase transitions during the generative diffusion process have also been observed in previous works (Biroli *et al* 2024, Raya and Ambrogioni 2024).

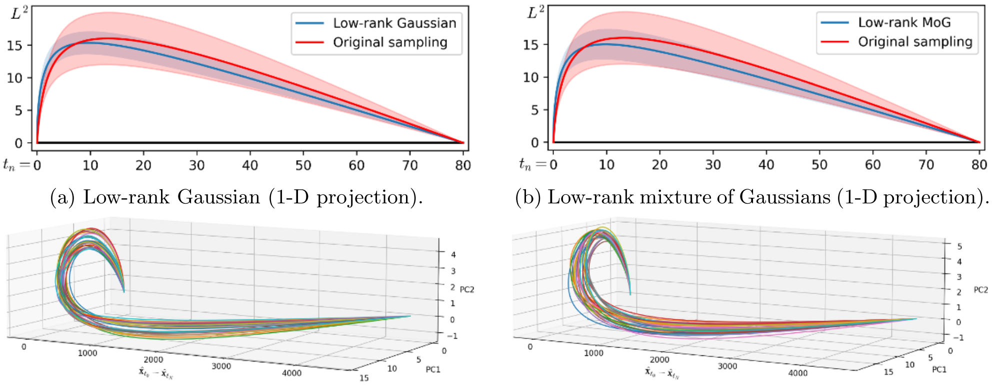
4.3. Regularity revisited under the Gaussian data assumption

Although we have analyzed closed-form solutions for the optimal sampling dynamics under the empirical data distribution (section 4.2), the resulting complex ODE hinders detailed theoretical analysis, particularly in the intermediate nonlinear regime. To gain further insight, we examine a simplified Gaussian data setting and demonstrate that trajectory regularity still emerges. These findings confirm that the observed structure is primarily an intrinsic property of deterministic sampling dynamics.

Proposition 4. *Suppose the data distribution is Gaussian* $p_{\mathrm{d}}(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, *where* $\boldsymbol{\mu} \in \mathbb{R}^d$, $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ *is positive semi-definite (PSD) with* $\mathrm{rank}(\boldsymbol{\Sigma}) = r \ll d$. *Let* $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}$ *denote the SVD, where* $\mathbf{U} \in \mathbb{R}^{d\times r}$ *contains eigenvectors* $\boldsymbol{u}_i$ *as columns, and* $\boldsymbol{\Lambda} \in \mathbb{R}^{r\times r}$ *is diagonal with eigenvalues* λ_i, $i \in [1, r]$. *In this setting, the PF-ODE solution* $\boldsymbol{x}_t$ *can be decomposed into the final sample* $\boldsymbol{x}_0$, *a scaled reverse displacement vector* $\mathbf{x}_{\mathrm{T}} - \mathbf{x}_0$, *and a trajectory residual* $\Delta_k(t)$:

$$\mathbf{x}_t = \mathbf{x}_0 + \frac{\sigma_t}{\sigma_{\mathrm{T}}}(\mathbf{x}_{\mathrm{T}} - \mathbf{x}_0) + \Delta_k(t), \qquad \Delta_k(t) = \sum_{k=1}^{r} \varphi_k(t)\, \mathbf{u}_k^{\mathrm{T}}(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu})\, \mathbf{u}_k,$$
$$\varphi_k(t) = \sqrt{\frac{\lambda_k + \sigma_t^2}{\lambda_k + \sigma_{\mathrm{T}}^2}} - \sqrt{\frac{\lambda_k}{\lambda_k + \sigma_{\mathrm{T}}^2}} - \frac{\sigma_t}{\sigma_{\mathrm{T}}}\left(1 - \sqrt{\frac{\lambda_k}{\lambda_k + \sigma_{\mathrm{T}}^2}}\right). \tag{23}$$

The squared norm of the trajectory residual, $\|\Delta_k(t)\|_2^2$, approximates the 1-D trajectory deviation and almost surely attains a unique maximum for $t \in [\sigma_0, \sigma_{\mathrm{T}}]$. Furthermore, it concentrates around its expectation $\mathbb{E}_{\mathbf{x}_{\mathrm{T}}}\left[\|\Delta_k(t)\|_2^2\right]$. Proofs are deferred to appendix B.8. Empirical verification is provided in figures 10, 13–15, 18–19, where we consider two cases: (1) fitting the entire dataset with a single Gaussian distribution, and (2) fitting each class with an individual Gaussian, yielding a Gaussian mixture model for the full dataset. Both simplified Gaussian data settings exhibit similar trajectory regularity, as we have observed in section 3.



(a) Low-rank Gaussian (1-D projection).

(b) Low-rank mixture of Gaussians (1-D projection).

(c) Low-rank Gaussian (3-D projection).

(d) Low-rank mixture of Gaussians (3-D projection).

Figure 10. Unconditional generation results on CIFAR-10 (32× 32). (a)–(b) One-dimensional trajectory deviation. (c)–(d) Three-dimensional trajectory visualization. Trajectory reconstruction and corresponding statistics are provided in figures 18 and 19.

5. Geometry-inspired time scheduling

In this section, as a simple illustration, we propose a new technique inspired by the geometric regularity of deterministic sampling in diffusion models to accelerate sampling and enhance sample quality. This technique is compatible with any numerical solver-based sampler and model architecture, easy to implement, and incurs negligible computational overhead.

5.1. Algorithm

A deterministic ODE-based numerical solver such as the Euler (Song *et al* 2021c) or Runge–Kutta (Liu *et al* 2022, Zhang and Chen 2023) relies on a pre-defined time schedule $\Gamma = \{t_0 = \epsilon, \cdots, t_N = T\}$ in the sampling process. Typically, given the initial time t_N and the final time t_0, the intermediate time steps from t_1 to t_{N-1} are determined by heuristic strategies such as uniform, quadratic (Song *et al* 2021a), log-SNR (Lu *et al* 2022a, 2022b), and polynomial functions (Karras *et al* 2022, Song *et al* 2023). In fact, the time schedule reflects our prior knowledge of the sampling trajectory shape. Under the constraint of the total *number of score function evaluations* (NFEs), an improved time schedule can reduce the local truncation error in each numerical step, and hopefully minimize the global truncation error. In this way, the sample quality generated by numerical methods could approach that of the exact solutions of the given empirical PF-ODE (11).

Our previous discussions in section 3 identified each sampling trajectory as a simple low-dimensional ‘boomerang’ curve. We thus leverage this geometric structure to re-allocate the intermediate timestamps according to the principle that assigning a larger time step size when the trajectory exhibits a relatively small curvature, while assigning a smaller time step size when the trajectory exhibits a relatively large curvature. Additionally, different trajectories share almost the same shape, regardless of the model architecture used or generation conditions, which helps us estimate the common structure of the sampling trajectory by using just a few 'warmup' samples. We name our approach to achieve the above goal as *geometry-inspired time scheduling* (GITS) and elaborate the details as follows.

The allocation of the intermediate timestamps can be formulated as an *integer programming problem* and solved using standard DP to search for an optimal time schedule (Cormen *et al* 2022)[7]. We first define a searching space denoted as Γ_{g}, which is a fine-grained grid including all possible intermediate timestamps. Then, we measure the trajectory curvature by the local truncation errors. More precisely, we define the cost from the current position $\mathbf{x}_{t_i}$ to the next position $\mathbf{x}_{t_j}$ as the difference between an Euler step and the ground-truth prediction, i.e. $c_{t_i\to t_j} := \mathcal{D}(\hat{\mathbf{x}}_{t_i\to t_j}, \mathbf{x}_{t_i\to t_j})$, where t_i and t_j are two intermediate timestamps from Γ_{g} and $t_i > t_j$. According to the empirical PF-ODE (11), the ground-truth prediction is calculated as $\mathbf{x}_{t_i\to t_j} = \mathbf{x}_{t_i} + \int_{t_i}^{t_j} \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t)\sigma_t' \mathrm{d}t$, and the Euler prediction is calculated as $\hat{\mathbf{x}}_{t_i\to t_j} = \mathbf{x}_{t_i} + (\sigma_{t_j} - \sigma_{t_i})\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_i})\sigma_{t_i}'$. The cost function $\mathcal{D}$ can be defined as the Euclidean distance in the original pixel space, or any other user-specified metric. Given all computed pairwise costs, which form a cost matrix, the problem reduces to a standard *minimum-cost path problem* and can be solved with DP. Since the global truncation error is not equal to the accumulation of local truncation errors at each step, we introduce a hyperparameter γ, analogous to the discount factor used in reinforcement learning (Sutton *et al* 1998), to compensate for this effect.

Dynamic programming is a fundamental concept widely used in computer science and many other fields (Cormen *et al* 2022). Watson *et al* (2021) was the first one leveraging DP to re-allocate the time schedule in diffusion models. However, our motivation differs significantly from that of this previous work. Watson *et al* (2021) exploited the fact that the evidence lower bound (ELBO) can be decomposed into separate KL terms and utilized DP to find the optimal discrete-time schedule that maximizes the training ELBO. However, this strategy was reported to worsen sample quality, as acknowledged by the authors. In contrast, we first discovered a strong trajectory regularity shared by all sampling trajectories, and then used several 'warmup' samples to estimate the trajectory curvature to determine a more effective time schedule for the sampling of diffusion models.

5.2. Experimental results

We adhere to the setup and experimental designs of the EDM framework (Karras *et al* 2022, Song *et al* 2023), with $f(t) = 0$, $g(t) = \sqrt{2t}$, and $\sigma_t = t$. Under this parameterization, the forward VE-SDE is expressed as $\mathrm{d}\mathbf{x}_t = \sqrt{2t}\,\mathrm{d}\mathbf{w}_t$, while the corresponding empirical PF-ODE is formulated as $\mathrm{d}\mathbf{x}_t/\mathrm{d}t = (\mathbf{x}_t - r_{\boldsymbol{\theta}}(\mathbf{x}_t; t))/t$. The temporal domain is segmented using a polynomial function $t_n = (t_0^{1/\rho} + \frac{n}{N}(t_N^{1/\rho} - t_0^{1/\rho}))^{\rho}$, where $t_0 = 0.002$, $t_N = 80$, $n \in [0, N]$, and $\rho = 7$. We initiate the DP experiments with 256 'warmup' samples randomly selected from Gaussian noise to create a more refined grid, and then calculate the associated cost matrix. The ground-truth predictions are generated by

[7] We also tried the Branch and Bound algorithm (Land and Doig 1960) and obtained similar results. Nevertheless, alternative approaches exist for determining the time schedule, such as using a trainable neural network (Frankel *et al* 2025, Tong *et al* 2025), by leveraging our discovered trajectory regularity.

Table 1. Sample quality comparison in terms of Fréchet Inception Distance (FID (Heusel *et al* 2017), lower is better) on four datasets (resolutions ranging from 32 × 32 to 256 × 256).

METHOD	NFE 5	6	8	10
CIFAR-10 32×32 (Krizhevsky and Hinton 2009)				
DDIM (Song *et al* 2021a)	49.66	35.62	22.32	15.69
DDIM + GITS (**ours**)	28.05	21.04	13.30	10.37
DPM-Solver-2 (Lu *et al* 2022a)	—	60.00	10.30	5.01
DPM-Solver++(3 M) (Lu *et al* 2022b)	24.97	11.99	4.54	3.00
DEIS-tAB3 (Zhang and Chen 2023)	14.39	9.40	5.55	4.09
UniPC (Zhao *et al* 2023)	23.98	11.14	3.99	2.89
AMED-Solver (Zhou *et al* 2024a)	—	7.04	5.56	4.14
AMED-Plugin (Zhou *et al* 2024a)	—	6.67	3.34	**2.48**
iPNDM (Zhang and Chen 2023)	13.59	7.05	3.69	2.77
iPNDM + GITS (**ours**)	**8.38**	**4.88**	**3.24**	**2.49**
FFHQ 64×64 (Karras *et al* 2019)				
DDIM (Song *et al* 2021a)	43.93	35.22	24.39	18.37
DDIM + GITS (**ours**)	29.80	23.67	16.60	13.06
DPM-Solver-2 (Lu *et al* 2022a)	—	83.17	22.84	9.46
DPM-Solver++(3 M) (Lu *et al* 2022b)	22.51	13.74	6.04	4.12
DEIS-tAB3 (Zhang and Chen 2023)	17.36	12.25	7.59	5.56
UniPC (Zhao *et al* 2023)	21.40	12.85	5.50	3.84
AMED-Solver (Zhou *et al* 2024a)	—	10.28	6.90	5.49
AMED-Plugin (Zhou *et al* 2024a)	—	9.54	5.28	3.66
iPNDM (Zhang and Chen 2023)	17.17	10.03	5.52	3.98
iPNDM + GITS (**ours**)	**11.22**	**7.00**	**4.52**	**3.62**
ImageNet 64×64 (Russakovsky *et al* 2015)				
DDIM (Song *et al* 2021a)	43.81	34.03	22.59	16.72
DDIM + GITS (**ours**)	24.92	19.54	13.79	10.83
DPM-Solver-2 (Lu *et al* 2022a)	—	44.83	12.42	6.84
DPM-Solver++(3 M) (Lu *et al* 2022b)	25.49	15.06	7.84	5.67
DEIS-tAB3 (Zhang and Chen 2023)	14.75	12.57	6.84	5.34
UniPC (Zhao *et al* 2023)	24.36	14.30	7.52	5.53
RES(M)[a] (Zhang *et al* 2023)	25.10	14.32	7.44	5.12
AMED-Solver (Zhou *et al* 2024a)	—	10.63	7.71	6.06
AMED-Plugin (Zhou *et al* 2024a)	—	12.05	7.03	5.01
iPNDM (Zhang and Chen 2023)	18.99	12.92	7.20	5.11
iPNDM + GITS (**ours**)	**10.79**	**8.43**	**5.82**	**4.48**

(Continued.)

Table 1. (Continued.)

METHOD	NFE			
	5	6	8	10
LSUN Bedroom 256×256 (Yu *et al* 2015) (pixel-space)				
DDIM (Song *et al* 2021a)	34.34	25.25	15.71	11.42
DDIM + GITS (**ours**)	22.04	16.54	11.20	9.04
DPM-Solver-2 (Lu *et al* 2022a)	—	80.59	23.26	9.61
DPM-Solver++(3 M) (Lu *et al* 2022b)	23.15	12.28	7.44	5.71
UniPC (Zhao *et al* 2023)	23.34	11.71	7.53	5.75
AMED-Solver (Zhou *et al* 2024a)	—	12.75	**6.95**	5.38
AMED-Plugin (Zhou *et al* 2024a)	—	11.58	7.48	5.70
iPNDM (Zhang and Chen 2023)	26.65	20.73	11.78	5.57
iPNDM + GITS (**ours**)	**15.85**	**10.41**	**7.31**	**5.28**

[a] Results reported by authors. More results are provided in table 8.

Table 2. Image generation results using Stable Diffusion v1.5 (two NFEs per sampling step).

METHOD	Step			
	5	6	7	8
DPM-Solver++(2 M) (Lu *et al* 2022b)	16.80	15.43	14.88	14.65
DPM-Solver++(2 M) + GITS (**ours**)	**15.53**	**13.18**	**12.32**	**12.17**

iPNDM (Zhang and Chen 2023), which employs a fourth-order multistep Runge–Kutta method with a lower-order warming start, using the polynomial time schedule with 60 NFEs. This yields a grid size of $|\Gamma_{\mathrm{g}}| = 61$. The default classifier-free guidance scale of 7.5 is used for Stable Diffusion (SDv1.5). We follow the standard FID and CLIP Score evaluation protocol for SDv1.5, using the reference statistics and 30k sampled captions from the MS-COCO validation set (Lin *et al* 2014). For other datasets, we compute FID based on 50k generated samples (Heusel *et al* 2017). All reported results for evaluated methods are obtained based on our developed open-source toolbox: https://github.com/zju-pi/diff-sampler.

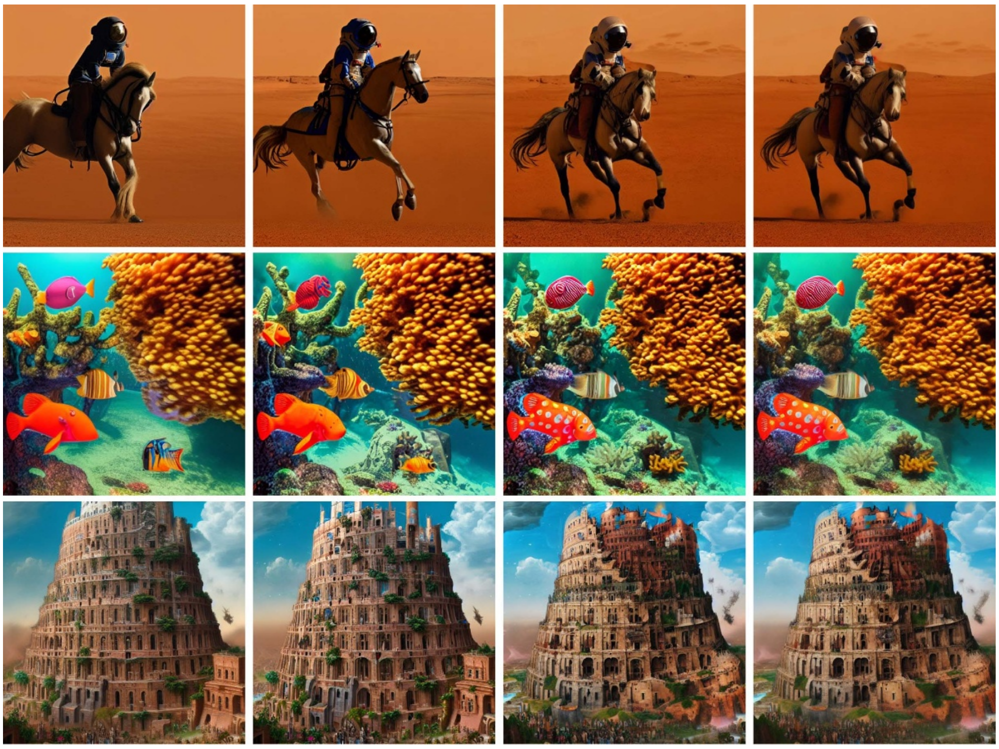
Image generation. As shown in tables 1 and 2, our simple time re-allocation strategy based on iPNDM (Zhang and Chen 2023) consistently outperforms all existing ODE-based accelerated sampling methods with a significant margin, especially in the few NFE cases. In particular, all time schedules in these datasets are searched based on the Euler method, i.e. DDIM (Song *et al* 2021a), but they are directly applicable to high-order methods such as iPNDM (Zhang and Chen 2023). The trajectory regularity we uncovered guarantees that the schedule determined through 256 ‘warmup’ samples is effective across all generated content. Furthermore, the experimental results suggest that identifying this trajectory regularity enhances our understanding of the mechanisms of diffusion models. This understanding opens avenues for developing tailored time schedules for more efficient diffusion sampling. Note that we did not adopt the analytical first step (AFS) that replaces the first numerical step with an analytical Gaussian score to save one NFE, proposed in Dockhorn *et al* (2022) and later used in Zhou *et al* (2024a), as we found AFS is particularly effective only for datasets with low-resolution images. DPM-Solver-2 (Lu *et al* 2022a) and AMED-Solver/Plugin (Zhou *et al* 2024a) are thus not applicable with NFE $=5$ (marked as '-') in table 1. Ablation studies on AFS are provided in table 8.

A concurrent work named AYS was recently proposed to optimize time schedules for sampling by minimizing the mismatch between the true backward SDE and its linear approximation, utilizing tools from stochastic calculus (Sabour *et al* 2024). In contrast, our GITS exploits the strong trajectory regularity inherent in diffusion models and yields time schedules via DP, requiring only a small number of 'warmup' samples. Our method also gets rid of the time-consuming Monte-Carlo computation in AYS (Sabour *et al* 2024) and therefore is several orders of magnitude faster. In figure 11, we compare image samples generated under different time schedules, using the publicly released colab code and its default setting[8]. The text prompts used are 'a photo of an astronaut riding a horse on Mars' (1st row); 'a whimsical underwater world inhabited by colorful sea creatures and coral reefs' (2nd row); 'a digital illustration of the Babel tower, 4k detailed, trending in artstation, fantasy vivid colors' (3rd row). The evaluated FID results for each schedule are 14.28 (uniform), 12.48 (AYS), and 12.01 (GITS). Besides, building on the significantly faster convergence of the denoising trajectory compared to the sampling trajectory, as discussed in section 4.2.1, we propose 'GITS-Jump' to further reduce sampling cost by 30% (from 10-step to 7-step), almost without degradation in image quality.

Time schedule comparison. From table 3, we can see that time schedules considerably affect the image generation performance. Compared with existing handcrafted schedules, the schedule we found better fits the underlying trajectory structure in the sampling of diffusion models and achieves smaller truncation errors with improved sample quality.

Running time. Our strategy is highly efficient and incurs a very low computational cost, without requiring access to the real dataset. The procedure starts with a small number of initial 'warmup' samples, followed by executing the given ODE-solver with both fine-grained and coarse-grained steps to construct the cost matrix for DP. Such a computation is performed only *once* per dataset, and it yields optimal time schedules for different NFE budgets simultaneously, thanks to the optimal substructure property (Cormen *et al* 2022). As reported in table 4, the entire algorithm takes less than or approximately one minute on datasets such as CIFAR-10, FFHQ, and ImageNet 64×64, and around 10 to 15 min for larger datasets such as LSUN Bedroom and LAION (Stable Diffusion), when evaluated on an NVIDIA A100 GPU.

[8] https://research.nvidia.com/labs/toronto-ai/AlignYourSteps/.



(a) Uniform.

(b) AYS.

(c) GITS.

(d) GITS +Jump(30%).

GITS	Step				
	6	**7**	8	9	10
Sampling trajectory	56.86/26.49	24.52/28.58	14.15/29.56	11.44/29.95	**12.01**/**30.11**
Denoising trajectory	20.55/29.77	**14.48**/**29.97**	13.16/30.06	12.45/30.09	**12.01**/**30.11**

Figure 11. *Top:* Visual comparison of samples generated by SDv1.5 using 10-step DPM-Solver++(2 M) under various time schedules: (a) uniform, (b) AYS-optimized (Sabour *et al* 2024), and (c) GITS-optimized. (d) Results from GITS+Jump, which further reduces the number of steps by 30%, are also presented for comparison. **Bottom**: FID and CLIP Scores (underlined) for GITS along the trajectories are reported.

Ablation studies. We provide ablation studies on the number of ‘warmup’ sample sizes and the grid size used for generating the fine-grained sampling trajectory in tables 5 and 6, respectively. The default experiments are conducted using iPNDM+GITS with the coefficient $\gamma = 1.15$ on CIFAR-10. We also provide a sensitivity analysis of the coefficient in table 8. It is shown that the number of ‘warmup’ samples is not a critical hyper-parameter, but reducing it generally increases the variance, as shown in table 6. Due to subtle differences among sampling trajectories (see figure 5), we recommend utilizing a reasonable number of ‘warmup’ samples to determine the optimal time schedule, such that this time schedule works well for all the generated samples.

Table 3. The comparison of FID results on CIFAR-10 across different time schedules.

	NFE			
TIME SCHEDULE	5	6	8	10
DDIM-uniform	36.98	28.22	19.60	15.45
DDIM-logsnr	53.53	38.20	24.06	16.43
DDIM-polynomial	49.66	35.62	22.32	15.69
DDIM + GITS (**ours**)	**28.05**	**21.04**	**13.30**	**10.09**
iPNDM-uniform	17.34	9.75	7.56	7.35
iPNDM-logsnr	19.87	10.68	4.74	2.94
iPNDM-polynomial	13.59	7.05	3.69	2.77
iPNDM + GITS (**ours**)	**8.38**	**4.88**	**3.24**	**2.49**

Table 4. Time (in seconds) used at different stages of GITS. 'warmup' samples are generated using 60 NFE, and the NFE budget for dynamic programming is set to 10.

DATASET	sample generation	cost matrix	dynamic programming	total time (s)
CIFAR-10 32×32	27.47	5.29	0.015	32.78
FFHQ 64×64	51.90	10.88	0.016	62.79
ImageNet 64×64	71.77	13.28	0.016	85.07
LSUN Bedroom	517.63	122.13	0.015	639.78
LAION (SDv1.5)	877.62	24.00	0.016	901.62

Table 5. Ablation study on the grid size of the dynamic programming-based time scheduling.

	NFE BUDGET						
GRID SIZE	4	5	6	7	8	9	10
11	20.88	10.15	5.11	4.63	3.16	2.78	2.77
21	16.22	9.87	4.83	3.76	3.39	3.20	2.81
41	15.34	9.34	4.83	5.54	3.01	2.66	2.53
61 (default)	15.10	8.38	4.88	5.11	3.24	2.70	2.49
81	15.74	8.57	5.09	5.38	3.10	2.93	2.38
101	15.03	8.72	5.02	5.19	3.12	2.81	2.41
iPNDM	24.82	13.59	7.05	5.08	3.69	3.17	2.77

Table 6. Ablation study on the 'warmup' sample size.

	SAMPLE SIZE							
NFE	1[a]	16	64	128	**256**	512	1024	2048
5	9.25	9.55 ± 0.75	9.57 ± 0.97	9.21 ± 0.44	8.84 ± 0.30	8.81 ± 0.04	8.89 ± 0.11	8.88 ± 0.12
6	5.12	5.36 ± 0.61	5.16 ± 0.28	4.99 ± 0.18	5.03 ± 0.25	5.20 ± 0.27	5.01 ± 0.19	4.92 ± 0.08
8	3.13	3.25 ± 0.13	3.22 ± 0.08	3.28 ± 0.10	3.27 ± 0.11	3.30 ± 0.11	3.29 ± 0.08	3.33 ± 0.10
10	2.41	2.46 ± 0.11	2.46 ± 0.05	2.45 ± 0.05	2.46 ± 0.04	2.45 ± 0.04	2.44 ± 0.05	2.44 ± 0.05

[a] indicates that a unique time schedule is searched for each of the 50k generated samples. This special case is more time-consuming while achieving similar results, owing to the strong trajectory regularity.

6. Related work and discussions

The popular VE SDEs (Song and Ermon 2019, Song *et al* 2021c) are taken as our main examples for analysis, which are equivalent to their VP counterparts according to Itô's lemma (see lemma 4 and appendix B.4). The equivalence has been established in their corresponding PF-ODE (rather than SDE) forms by using the change-of-variable formula (see proposition 1 of (Song *et al* 2021a) and proposition 3 of (Zhang and Chen 2023)). Karras *et al* (2022) also presented a series of operational steps to reframe different models within a single framework (see appendix C of (Karras *et al* 2022)). The equivalence guarantees the wide applicability of our conclusions, even though we focus on VE-SDEs. Besides, instead of training a noise-conditional score model with DSM (Vincent 2011, Song and Ermon 2019, Song *et al* 2021b) or training a noise-prediction model to estimate the added noise in each step (Ho *et al* 2020, Nichol and Dhariwal 2021, Vahdat *et al* 2021, Song *et al* 2021a, Bao *et al* 2022), we follow (Kingma *et al* 2021, Karras *et al* 2022) and train a denoising model that predicts the reconstructed data from its corrupted version. With the help of simplified empirical PF-ODE (11), we can characterize an implicit denoising trajectory, draw inspiration from classical non-parametric mean shift (Fukunaga and Hostetler 1975, Cheng 1995, Comaniciu and Meer 2002), and derive various trajectory properties.

Denoising trajectories have been observed since the renaissance of diffusion models (see figure 6 of Ho *et al* (2020)) and later in figure 3 of Kwon *et al* (2023), but they have not been systematically investigated, perhaps due to the indirect model parameterization. Karras *et al* (2022) were the first to note that the denoising output reflects the tangent of the sampling trajectory, consistent with our corollary 1. However, their work did not formulate it in differential-equation form nor examine how it controls the evolution of the sampling trajectory. In fact, Karras *et al* (2022) mentioned this property to argue that the sampling trajectory of (11) is approximately linear, owing to the slow variation of the denoising output, and validated this intuition using a 1-D toy example. In contrast, we establish the equivalence of linear diffusion models and provide an in-depth analysis of high-dimensional sampling trajectories with real data, highlighting their intrinsically low dimensionality and pronounced geometric regularity.

The mathematical foundations of the closed-form solution for the DSM objective, or equivalently, the DAE objective, were established more than half a century ago under the framework of *empirical Bayes* (Robbins 1956); see, for instance, Chapter 1 of (Efron 2010). Perhaps the earliest appearance of the closed-form solution (9) for a finite dataset within the literature of diffusion models is in appendix B.3 of (Karras *et al* 2022), where it was included for completeness and not connected to kernel density estimation (KDE) or any application. Subsequent works explicitly adopted the KDE-based interpretation (or, optimal denoising output) to analyze memorization and generalization in generative diffusion models (Gu *et al* 2023, Kadkhodaie *et al* 2023, Scarvelis *et al* 2023, Yi *et al* 2023, Chen *et al* 2023a, Kamb and Ganguli 2024, Li *et al* 2024, Niedoba *et al* 2024), listed here chronologically by their arXiv release dates. The early arXiv version[9] of our paper (Chen *et al* 2023a) was among the first, or at least concurrent with these studies. Importantly, our unique contribution lies in leveraging this well-established analytical formula to provide theoretical guarantees for the observed trajectory regularity and to extract additional insights from this approximate model in the context of diffusion-based generative models (section 4.2).

The trajectory regularity revealed in this paper is presented as an independent scientific discovery, supported by comprehensive empirical and theoretical analysis designed to reveal, characterize, and understand these principles. It is not intended merely as a prerequisite for specific algorithms; rather, it provides an intuitive yet grounded perspective on the underlying mechanics of diffusion models and helps explain the success of many widely used heuristic methods. (I) The observation that sampling trajectories follow a simple curvature and torsion function clarifies, for instance, why large steps can be safely taken at the beginning of sampling (Dockhorn *et al* 2022, Zhou *et al* 2024a) without incurring significant truncation errors, and why polynomial time schedules outperform uniform schedules during sampling. Moreover, training efficiency improves when a larger computational training budget is allocated to the intermediate non-linear region of the trajectory and fewer to the near-linear regions (Karras *et al* 2022, Chen 2023, Hang *et al* 2024), considering the trajectory shape. While previous work largely converged on these effective time/noise schedules through trial-and-error search (Karras *et al* 2022, Lu *et al* 2022a, Chen 2023, Hang *et al* 2024, Sabour *et al* 2024). (II) Our geometric perspective also provides a theoretical justification for the common heuristic of disabling classifier-free guidance (Ho and Salimans 2022, Karras *et al* 2024, Kynkäänniemi *et al* 2024) at the beginning or end of the sampling process with minimal performance degradation (Kynkäänniemi *et al* 2024, Castillo *et al* 2025). This phenomenon arises naturally, since the intermediate nonlinear region strongly influences trajectory orientation, whereas the early and late linear regions contribute little.

Finally, we describe a potential application of the discovered trajectory regularity for accelerating the sampling process. Different from most existing methods focusing on developing improved ODE-solvers (Song *et al* 2021a, Karras *et al* 2022, Liu *et al* 2022, Lu *et al* 2022a, Zhang and Chen 2023, Zhao *et al* 2023, Zhou *et al* 2024a) while selecting time schedules through a handcrafted or empirical tuning, we leverage the trajectory regularity of deterministic sampling dynamics to more effectively allocate discretized

[9] https://arxiv.org/abs/2305.19947.

time steps. Our method achieves acceleration by several orders of magnitude compared with distillation-based sampling methods (Luhman and Luhman 2021, Salimans and Ho 2022, Song *et al* 2023, Zheng *et al* 2023, Kim *et al* 2024, Zhou *et al* 2024b). Although Watson *et al* (2021) were the first to employ DP for optimizing time schedules based on the decomposable nature of the ELBO objective, their method was shown to degrade sample quality. Moreover, while various theoretical studies have explored convergence analysis and score estimation of diffusion models, none of them have examined the trajectory-level properties that govern the sampling dynamics (De Bortoli 2022, Pidstrigach 2022, Lee *et al* 2023, Chen *et al* 2023b, 2023c).

7. Conclusion

We reveal that a strong trajectory regularity consistently emerges in the deterministic sampling dynamics of diffusion-based generative models, regardless of the model implementation or generated content. This regularity is explained by characterizing and analyzing the implicit denoising trajectory, particularly its behavior under KDE-based data modeling. These insights into the underlying trajectory structure lead to an accelerated sampling method that enhances image synthesis quality with negligible computational overhead. We hope that the empirical and theoretical findings presented in this paper contribute to a deeper understanding of diffusion models and inspire further research into more efficient training paradigms and faster sampling algorithms.

Future works. We aim to explore deeper geometric regularities in sampling trajectories, characterize more precise structural patterns, and identify new applications inspired by these insights. Several promising directions are outlined below:

- The geometric regularity of sampling trajectories analyzed in this paper may have potential connections to the behavior of random walk paths simulated in the forward process of diffusion models. In the limit of infinite dimensions and trajectory length, random walk-based trajectories exhibit similarly intriguing low-dimensional structures, with the explained variance taking an analytic form. Furthermore, their projections onto PCA subspaces follow Lissajous curves (Antognini and Sohl-Dickstein 2018, Moore *et al* 2018). Extending existing theoretical results from the forward diffusion process to the backward sampling process remains an open problem.
- A distinct three-stage pattern emerges in the sampling dynamics when using optimal score functions. Concurrently with our earlier manuscripts (Chen *et al* 2023a, 2024), Biroli *et al* (2024) introduced concepts from statistical physics, such as symmetry breaking and phase transitions, to characterize sampling dynamics. They provided analytic solutions for critical points in a simplified setting (two well-separated Gaussian mixture classes), and discussed the trade-off between generalization and memorization. It is particularly intriguing to bridge these theoretical insights with realistic diffusion models, especially incorporating conditional signals into the framework.
- In practice, sampling in diffusion-based generative models is typically performed using general-purpose numerical solvers, sometimes augmented with learned solver

coefficients or sampling schedules in a data-driven way (Frankel *et al* 2025, Tong *et al* 2025). Our findings reveal that each integral curve of the gradient field defined by a diffusion model lies within an extremely low-dimensional subspace embedded in the high-dimensional data space, with a regular trajectory shape shared across all initial conditions. While we present a preliminary attempt to exploit this structure, further investigation in this direction holds great promise.

Acknowledgment

This work is supported by the University at Buffalo Office of Vice President for Research and Economic Development.

Appendix A. Further readings

A.1. Details about popular linear SDEs

In the literature, two specific forms of linear SDEs are widely used in large-scale diffusion models (Balaji *et al* 2022, Ramesh *et al* 2022, Rombach *et al* 2022, Saharia *et al* 2022, Peebles and Xie 2023, Podell *et al* 2024, Xie *et al* 2025), namely, the VP SDE and the VE SDE (Song *et al* 2021c, Karras *et al* 2022). They correspond to the continuous versions of previously established models, i.e. DDPMs (Ho *et al* 2020, Nichol and Dhariwal 2021) and NCSNs (Song and Ermon 2019, 2020), respectively. Next, we demonstrate that the original notations of VP-SDE, VE-SDE (Song *et al* 2021c), including recently proposed flow matching-based generative models (Albergo *et al* 2023, Heitz *et al* 2023, Lipman *et al* 2023, Neklyudov *et al* 2023, Liu *et al* 2023b, Esser *et al* 2024) can be recovered by properly setting the coefficients s_t and σ_t:

- VP-SDEs (Ho *et al* 2020, Nichol and Dhariwal 2021, Song *et al* 2021a, 2021c): By setting $s_t = \sqrt{\alpha_t}$, $\sigma_t = \sqrt{(1-\alpha_t)/\alpha_t}$, $\beta_t = -\mathrm{d}\log\alpha_t/\mathrm{d}t$, and $\alpha_t \in (0,1]$ as a decreasing sequence with $\alpha_0 = 1, \alpha_T \approx 0$, we have the transition kernel $p_{0t}(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t; \sqrt{\alpha_t}\mathbf{z}_0, (1-\alpha_t)\mathbf{I})$, or equivalently,

$$\mathbf{z}_t = \sqrt{\alpha_t}\;\mathbf{z}_0 + \sqrt{1-\alpha_t}\;\boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I}), \tag{24}$$

  with the forward linear SDE $\mathrm{d}\mathbf{z}_t = -\frac{1}{2}\beta_t\mathbf{z}_t\,\mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\mathbf{w}_t$.
- VE-SDEs (Song and Ermon 2019, Song *et al* 2021c): By setting $s_t = 1$, and $\sigma_0 \approx 0$, $\sigma_T \gg 1$ for an increasing sequence σ_t, we have the transition kernel $p_{0t}(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t; \mathbf{z}_0, \sigma_t^2\mathbf{I})$, or equivalently,

$$\mathbf{z}_t = \mathbf{z}_0 + \sigma_t\boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I}), \tag{25}$$

  with the forward linear SDE $\mathrm{d}\mathbf{z}_t = \sqrt{\mathrm{d}\sigma_t^2/\mathrm{d}t}\;\mathrm{d}\mathbf{w}_t$.
- A typical flow matching-based instantiation (Lipman *et al* 2023, Liu *et al* 2023b, Esser *et al* 2024) defines the transition kernel directly without relying on the

forward linear SDE, with $s_t = 1 - t/T$, $\sigma_t = t/(T-t)$, i.e. $p_{0t}(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t; (1 - t/T)\mathbf{z}_0, (t/T)^2\mathbf{I})$, or equivalently,

$$\mathbf{z}_t = (1 - t/T)\,\mathbf{z}_0 + t/T\boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})\,. \tag{26}$$

A.2. Details about score matching

The score function $\nabla_{\mathbf{z}_t} \log p_t(\mathbf{z}_t)$ (Hyvärinen 2005, Lyu 2009), which can be estimated with nonparametric score matching via KDE, implicit score matching via *integration by parts* formula (Hyvärinen 2005), sliced score matching via *Hutchinson's trace estimator* (Song *et al* 2019), or more typically, DSM via *mean squared regression* (Vincent 2011, Song and Ermon 2019, Karras *et al* 2022). The DSM objective function of training a score-estimation model $s_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$ is

$$\mathcal{L}_{\mathrm{DSM}}\left(\boldsymbol{\theta}; \lambda(t)\right) := \int_0^T \lambda(t) \mathbb{E}_{\mathbf{z}_0 \sim p_d} \mathbb{E}_{\mathbf{z}_t \sim p_{0t}(\mathbf{z}_t|\mathbf{z}_0)} \| \overbrace{\nabla_{\mathbf{z}_t} \log p_{\boldsymbol{\theta}}(\mathbf{z}_t; t)}^{s_{\boldsymbol{\theta}}(\mathbf{z}_t;t)} - \nabla_{\mathbf{z}_t} \log p_{0t}(\mathbf{z}_t|\mathbf{z}_0) \|_2^2 \mathrm{d}t. \tag{27}$$

The weighting function $\lambda(t)$ across different noise levels reflects our preference for visual quality or density estimation during model training (Song *et al* 2021b, Kim *et al* 2022). The optimal estimator $s^{\star}_{\boldsymbol{\theta}}(\mathbf{z}_t; t)$ equals $\nabla_{\mathbf{z}_t} \log p_t(\mathbf{z}_t)$, and therefore we can use the converged score-estimation model as an effective proxy for the ground-truth score function. lemma 1 shows that we can also estimate the conditional expectation $\mathbb{E}(\mathbf{z}_0|\mathbf{z}_t)$ instead, typically using a DAE (Vincent *et al* 2008, Bengio *et al* 2013b). In fact, the mathematical essentials of the deep connection between DAE and DSM were established more than half a century ago under the framework of *empirical Bayes* (Robbins 1956); see, for example, chapter 1 of a textbook (Efron 2010), or technical details given in appendix A of our early manuscript (Chen *et al* 2023a).

A.3. Details about numerical approximation

Given the empirical PF-ODE (10), generally, we have two formulas to calculate the exact solution from the current position $\mathbf{z}_{t_{n+1}}$ to the next position $\mathbf{z}_{t_n}$ ($t_0 \leqslant t_n < t_{n+1} \leqslant t_N$) in the ODE-based sampling to obtain the sampling trajectory from t_N to t_0. One is the direct integral from t_{n+1} to t_n

$$\mathbf{z}_{t_n} = \mathbf{z}_{t_{n+1}} + \int_{t_{n+1}}^{t_n} \frac{\mathrm{d}\mathbf{z}_t}{\mathrm{d}t} \mathrm{d}t = \mathbf{z}_{t_{n+1}} + \int_{t_{n+1}}^{t_n} \left(\frac{\mathrm{d} \log s_t}{\mathrm{d}t} \mathbf{z}_t - \frac{1}{2} s_t^2 \frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t} \nabla_{\mathbf{z}_t} \log p_t\left(\mathbf{z}_t\right) \right), \tag{28}$$

and another leverages the semi-linear structure in the PF-ODE to derive the following equation with the *variant of constants* formula (Lu *et al* 2022a, Zhang and Chen 2023)

$$\begin{aligned} \mathbf{z}_{t_n} &= \exp\left(\int_{t_{n+1}}^{t_n} f\left(t\right) \mathrm{d}t \right) \mathbf{z}_{t_{n+1}} - \int_{t_{n+1}}^{t_n} \left(\exp\left(\int_t^{t_n} f\left(r\right) \mathrm{d}r \right) \frac{g^2\left(t\right)}{2} \nabla_{\mathbf{z}_t} \log p_t\left(\mathbf{z}_t\right) \right) \mathrm{d}t \\ &= \frac{s_{t_n}}{s_{t_{n+1}}} \mathbf{z}_{t_{n+1}} - s_{t_n} \int_{t_{n+1}}^{t_n} \left(s_t \sigma_t \sigma_t' \nabla_{\mathbf{z}_t} \log p_t\left(\mathbf{z}_t\right) \right) \mathrm{d}t. \end{aligned} \tag{29}$$

The above integrals, whether in (28) or (29), involving the score function parameterized by a neural network, are generally intractable. Therefore, deterministic sampling in diffusion models centers on approximating these integrals with numerical methods in each step. In practice, various sampling strategies inspired by classic numerical methods have been proposed to solve the backward PF-ODE (10), including the Euler method (Song *et al* 2021a), Heun's method (Karras *et al* 2022), Runge–Kutta method (Song *et al* 2021c, Liu *et al* 2022, Lu *et al* 2022a), and linear multistep method (Liu *et al* 2022, Lu *et al* 2022b, Zhang and Chen 2023, Zhao *et al* 2023, Chen *et al* 2024, Zhou *et al* 2024a).

Appendix B. Proofs

In this section, we provide detailed proofs of the lemmas, propositions, and theorems presented in the main content.

B.1. Proof of lemma 1

Lemma 1. *Let the clean data be* $\mathbf{z}_0 \sim p_{\mathrm{d}}$*, and consider a transition kernel that adds Gaussian noise to the data,* $p_{0t}(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}\left(\mathbf{z}_t; s_t\mathbf{z}_0, s_t^2\sigma_t^2\mathbf{I}\right)$*. Then the score function is related to the posterior expectation by*

$$\nabla_{\mathbf{z}_t} \log p_t\left(\mathbf{z}_t\right) = \left(s_t\sigma_t\right)^{-2}\left(s_t\mathbb{E}\left(\mathbf{z}_0|\mathbf{z}_t\right) - \mathbf{z}_t\right), \tag{30}$$

or equivalently, by linearity of expectation,

$$\nabla_{\mathbf{z}_t} \log p_t\left(\mathbf{z}_t\right) = -\left(s_t\sigma_t\right)^{-1}\mathbb{E}_{p_{t0}(\mathbf{z}_0|\mathbf{z}_t)}\boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t = \left(s_t\sigma_t\right)^{-1}\left(\mathbf{z}_t - s_t\mathbf{z}_0\right). \tag{31}$$

Proof. We take derivative of $p_t(\mathbf{z}_t) = \int p_{\mathrm{d}}(\mathbf{z}_0)p_{0t}(\mathbf{z}_t|\mathbf{z}_0)\mathrm{d}\mathbf{z}_0$ with respect to $\mathbf{z}_t$,

$$\begin{aligned}
\nabla_{\mathbf{z}_t} p_t\left(\mathbf{z}_t\right) &= \int \frac{\left(s_t\mathbf{z}_0 - \mathbf{z}_t\right)}{s_t^2\sigma_t^2} p_{\mathrm{d}}\left(\mathbf{z}_0\right)p_{0t}\left(\mathbf{z}_t|\mathbf{z}_0\right)\mathrm{d}\mathbf{z}_0\\
s_t^2\sigma_t^2\nabla_{\mathbf{z}_t} p_t\left(\mathbf{z}_t\right) &= \int s_t\mathbf{z}_0 p_{\mathrm{d}}\left(\mathbf{z}_0\right)p_{0t}\left(\mathbf{z}_t|\mathbf{z}_0\right)\mathrm{d}\mathbf{z}_0 - \mathbf{z}_t p_t\left(\mathbf{z}_t\right)\\
s_t^2\sigma_t^2\frac{\nabla_{\mathbf{z}_t} p_t\left(\mathbf{z}_t\right)}{p_t\left(\mathbf{z}_t\right)} &= s_t\int \mathbf{z}_0 p_t\left(\mathbf{z}_0|\mathbf{z}_t\right)\mathrm{d}\mathbf{z}_0 - \mathbf{z}_t\\
\nabla_{\mathbf{z}_t}\log p_t\left(\mathbf{z}_t\right) &= \left(s_t\sigma_t\right)^{-2}\left(s_t\mathbb{E}\left(\mathbf{z}_0|\mathbf{z}_t\right) - \mathbf{z}_t\right).
\end{aligned} \tag{32}$$

We further have

$$\begin{aligned}
\nabla_{\mathbf{z}_t}\log p_t\left(\mathbf{z}_t\right) = \left(s_t\sigma_t\right)^{-2}\left(s_t\mathbb{E}\left(\mathbf{z}_0|\mathbf{z}_t\right) - \mathbf{z}_t\right) &= \left(s_t\sigma_t\right)^{-1}\mathbb{E}\left(\frac{s_t\mathbf{z}_0 - \mathbf{z}_t}{s_t\sigma_t}|\mathbf{z}_t\right)\\
&= -\left(s_t\sigma_t\right)^{-1}\mathbb{E}_{p_{t0}(\mathbf{z}_0|\mathbf{z}_t)}\boldsymbol{\epsilon}_t,
\end{aligned} \tag{33}$$

by linearity of expectation, where $\mathbf{z}_t = s_t\mathbf{z}_0 + s_t\sigma_t\boldsymbol{\epsilon}_t$, $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$ according to the transition kernel (2). □

B.2. Proof of lemma 2

Lemma 2. *The optimal estimator* $r^\star_{\boldsymbol{\theta}}(\mathbf{z}_t;t)$ *for the DAE objective, also known as the Bayesian least squares estimator or minimum mean square error (MMSE) estimator, is given by* $\mathbb{E}(\mathbf{z}_0|\mathbf{z}_t)$.

Proof. The solution is easily obtained by setting the derivative of $\mathcal{L}_{\mathrm{DAE}}$ equal to zero. For each noisy sample $\mathbf{z}_t$, we have

$$\begin{gathered}\nabla_{r_{\boldsymbol{\theta}}(\mathbf{z}_t;t)}\mathcal{L}_{\mathrm{DAE}}=0\\ \int p_{t0}(\mathbf{z}_0|\mathbf{z}_t)\left(r^\star_{\boldsymbol{\theta}}(\mathbf{z}_t;t)-\mathbf{z}_0\right)\mathrm{d}\mathbf{z}_0=0\\ \int p_{t0}(\mathbf{z}_0|\mathbf{z}_t)r^\star_{\boldsymbol{\theta}}(\mathbf{z}_t;t)\,\mathrm{d}\mathbf{z}_0=\int p_{t0}(\mathbf{z}_0|\mathbf{z}_t)\mathbf{z}_0\mathrm{d}\mathbf{z}_0\\ r^\star_{\boldsymbol{\theta}}(\mathbf{z}_t;t)=\mathbb{E}(\mathbf{z}_0|\mathbf{z}_t).\end{gathered}\tag{34}$$

□

B.3. Proof of lemma 3

Lemma 3. *Let* $\mathcal{D}:=\{\mathbf{y}_i\in\mathbb{R}^d\}_{i\in\mathcal{I}}$ *denote a dataset of* $|\mathcal{I}|$ *i.i.d. data points drawn from* p_{d}*. When training a DAE with the empirical data distribution* $\hat{p}_d$*, the optimal denoising output is a convex combination of original data points, namely*

$$r^\star_{\boldsymbol{\theta}}(\mathbf{z}_t;t)=\min_{r_{\boldsymbol{\theta}}}\mathbb{E}_{\mathbf{y}\sim\hat{p}_d}\mathbb{E}_{\mathbf{z}_t\sim p_{0t}(\mathbf{z}_t|\mathbf{y})}\|r_{\boldsymbol{\theta}}(\mathbf{z}_t;t)-\mathbf{y}\|_2^2=\sum_i\frac{\exp\left(-\|\mathbf{z}_t-\mathbf{y}_i\|_2^2/2\sigma_t^2\right)}{\sum_j\exp\left(-\|\mathbf{z}_t-\mathbf{y}_j\|_2^2/2\sigma_t^2\right)}\mathbf{y}_i,\tag{35}$$

where $\hat{p}_d(\mathbf{y})$ *is the sum of multiple Dirac delta functions, i.e.* $\hat{p}_d(\mathbf{y})=(1/|\mathcal{I}|)\sum_{i\in\mathcal{I}}\delta(\|\mathbf{y}-\mathbf{y}_i\|)$.

Proof. Based on lemmas 1 and 2, and the Gaussian KDE $\hat{p}_t(\mathbf{z}_t)=\int p_{0t}(\mathbf{z}_t|\mathbf{y})\hat{p}_d(\mathbf{y})=\frac{1}{|\mathcal{I}|}\sum_i\mathcal{N}\left(\mathbf{z}_t;\mathbf{y}_i,\sigma_t^2\mathbf{I}\right)$, the optimal denoising output is

$$\begin{aligned}r^\star_{\boldsymbol{\theta}}(\mathbf{z}_t;t)=\mathbb{E}(\mathbf{z}_0|\mathbf{z}_t)&=\mathbf{z}_t+\sigma_t^2\nabla_{\mathbf{z}_t}\log\hat{p}_t(\mathbf{z}_t)\\ &=\mathbf{z}_t+\sigma_t^2\sum_i\frac{\nabla_{\mathbf{z}_t}\mathcal{N}\left(\mathbf{z}_t;\mathbf{y}_i,\sigma_t^2\mathbf{I}\right)}{\sum_j\mathcal{N}\left(\mathbf{z}_t;\mathbf{y}_j,\sigma_t^2\mathbf{I}\right)}\\ &=\mathbf{z}_t+\sigma_t^2\sum_i\frac{\mathcal{N}\left(\mathbf{z}_t;\mathbf{y}_i,\sigma_t^2\mathbf{I}\right)}{\sum_j\mathcal{N}\left(\mathbf{z}_t;\mathbf{y}_j,\sigma_t^2\mathbf{I}\right)}\left(\frac{\mathbf{y}_i-\mathbf{z}_t}{\sigma_t^2}\right)\\ &=\mathbf{z}_t+\sum_i\frac{\mathcal{N}\left(\mathbf{z}_t;\mathbf{y}_i,\sigma_t^2\mathbf{I}\right)}{\sum_j\mathcal{N}\left(\mathbf{z}_t;\mathbf{y}_j,\sigma_t^2\mathbf{I}\right)}(\mathbf{y}_i-\mathbf{z}_t)\\ &=\sum_i\frac{\mathcal{N}\left(\mathbf{z}_t;\mathbf{y}_i,\sigma_t^2\mathbf{I}\right)}{\sum_j\mathcal{N}\left(\mathbf{z}_t;\mathbf{y}_j,\sigma_t^2\mathbf{I}\right)}\mathbf{y}_i\\ &=\sum_i\frac{\exp\left(-\|\mathbf{z}_t-\mathbf{y}_i\|_2^2/2\sigma_t^2\right)}{\sum_j\exp\left(-\|\mathbf{z}_t-\mathbf{y}_j\|_2^2/2\sigma_t^2\right)}\mathbf{y}_i,\end{aligned}\tag{36}$$

where each weight is calculated based on the time-scaled and normalized Euclidean distance between the input $\mathbf{z}_t$ and each data point $\mathbf{y}_i$ and the sum of coefficients equals one. □

B.4. Proof of lemma 4

Lemma 4. *The linear diffusion process defined as (3) can be transformed into its VE counterpart with the change of variables $\mathbf{x}_t = \mathbf{z}_t/s_t$, keeping the SNR function unchanged.*

Proof. We adopt the change-of-variables formula $\mathbf{x}_t = \boldsymbol{\phi}(t, \mathbf{z}_t) = \mathbf{z}_t/s_t$ with $\boldsymbol{\phi}: [0,T] \times \mathbb{R}^n \to \mathbb{R}^n$, and denote the ith dimension of $\mathbf{z}_t$, $\mathbf{x}_t$ and $\mathbf{w}_t$ as $\mathbf{z}_t[i]$, $\mathbf{x}_t[i]$, and $\mathbf{w}_t[i]$ respectively; $\boldsymbol{\phi} = [\phi_1, \cdots, \phi_i, \cdots, \phi_n]^{\mathrm{T}}$ with a twice differentiable scalar function $\phi_i(t,z) = z/s_t$ of two real variables t and z. Since each dimension of $\mathbf{z}_t$ is independent, we can apply Itô's lemma (Oksendal 2013) to each dimension with $\phi_i(t, \mathbf{z}_t[i])$ separately. We have

$$\frac{\partial \phi_i}{\partial t} = -\frac{z}{s_t^2}\frac{\mathrm{d}s_t}{\mathrm{d}t}, \quad \frac{\partial \phi_i}{\partial z} = \frac{1}{s_t}, \quad \frac{\partial^2 \phi_i}{\partial z^2} = 0, \quad \mathrm{d}\mathbf{z}_t[i] = \frac{\mathrm{d}\log s_t}{\mathrm{d}t}\mathbf{z}_t[i]\,\mathrm{d}t + s_t\sqrt{\frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t}}\mathrm{d}\mathbf{w}_t[i], \tag{37}$$

then

$$\begin{aligned}
\mathrm{d}\phi_i(t, \mathbf{z}_t[i]) &= \left(\frac{\partial \phi_i}{\partial t} + f(t)\,\mathbf{z}_t[i]\frac{\partial \phi_i}{\partial z} + \frac{g^2(t)}{2}\frac{\partial^2 \phi_i}{\partial z^2}\right)\mathrm{d}t + g(t)\frac{\partial \phi_i}{\partial z}\mathrm{d}\mathbf{w}_t[i] \\
&= \left(\frac{\partial \phi_i}{\partial t} + \frac{g^2(t)}{2}\frac{\partial^2 \phi_i}{\partial z^2}\right)\mathrm{d}t + \frac{\partial \phi_i}{\partial z}\mathrm{d}\mathbf{z}_t[i] \\
\mathrm{d}\mathbf{x}_t[i] &= -\frac{\mathbf{z}_t[i]}{s_t}\frac{\mathrm{d}\log s_t}{\mathrm{d}t}\mathrm{d}t + \frac{1}{s_t}\left(\frac{\mathrm{d}\log s_t}{\mathrm{d}t}\mathbf{z}_t[i]\,\mathrm{d}t + s_t\sqrt{\frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t}}\mathrm{d}w_t\right) \\
\mathrm{d}\mathbf{x}_t[i] &= \sqrt{\mathrm{d}\sigma_t^2/\mathrm{d}t}\;\mathrm{d}\mathbf{w}_t[i] \quad \Rightarrow \quad \mathrm{d}\mathbf{x}_t = \sqrt{\mathrm{d}\sigma_t^2/\mathrm{d}t}\;\mathrm{d}\mathbf{w}_t,
\end{aligned} \tag{38}$$

with the initial condition $\mathbf{x}_0 = \mathbf{z}_0 \sim p_{\mathrm{d}}$. Since σ_t in the above VE-SDE ($\mathbf{x}$-space) is exactly the same as that used in the original SDE ($\mathbf{z}$-space, (3)), the SNR remains unchanged. □

We also establish the connections between their score functions and sampling behaviors. Similarly, we have the score function ($t \in [0,T]$)

$$\begin{aligned}
\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t) &= \nabla_{\mathbf{x}_t}\log\int \mathcal{N}\left(\mathbf{x}_t; \mathbf{x}_0, \sigma_t^2\mathbf{I}\right)p_{\mathrm{d}}(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0 \\
&= \nabla_{\mathbf{z}_t/s_t}\log\int \mathcal{N}\left(\mathbf{z}_t/s_t; \mathbf{x}_0, \sigma_t^2\mathbf{I}\right)p_{\mathrm{d}}(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0 \\
&= s_t\nabla_{\mathbf{z}_t}\log\int s_t^d\mathcal{N}\left(\mathbf{z}_t; s_t\mathbf{z}_0, s_t^2\sigma_t^2\mathbf{I}\right)p_{\mathrm{d}}(\mathbf{z}_0)\,\mathrm{d}\mathbf{z}_0 = s_t\nabla_{\mathbf{z}_t}\log p_t(\mathbf{z}_t).
\end{aligned} \tag{39}$$

Corollary 4. *With the same numerical method, the results obtained by solving (28) or (29) are not equal in the general cases. But they become exactly the same by first transforming the formulas into the $\boldsymbol{x}$-space and then perform numerical approximation.*

Corollary 5. *With the same numerical method, the result obtained by solving (29) in the $\boldsymbol{z}$-space is exactly the same as the result obtained by solving (28) or (29) in the $\boldsymbol{x}$-space.*

Proof. Given the sample $\mathbf{z}_{t_n}$ obtained by solving the equation (29) in $\mathbf{z}$-space starting from $\mathbf{z}_{t_{n+1}}$, we demonstrate that $\mathbf{z}_{t_n}/s_{t_n}$ is exactly equal to sampling with the equation (28) in $\mathbf{x}$-space starting from $\mathbf{x}_{t_{n+1}} = \mathbf{z}_{t_{n+1}}/s_{t_{n+1}}$ to $\mathbf{x}_{t_n}$. We have

$$\begin{aligned}\mathbf{z}_{t_n} &= s_{t_n}\left(\frac{\mathbf{z}_{t_{n+1}}}{s_{t_{n+1}}} - \int_{t_{n+1}}^{t_n} s_t\sigma_t\sigma_t'\nabla_{\mathbf{z}_t}\log p_t\left(\mathbf{z}_t\right)\mathrm{d}t\right)\\ &= s_{t_n}\left(\mathbf{x}_{t_{n+1}} + \int_{t_{n+1}}^{t_n} -\sigma_t\nabla_{\mathbf{x}_t}\log p_t\left(\mathbf{x}_t\right)\sigma_t'\mathrm{d}t\right) = s_{t_n}\left(\mathbf{x}_{t_{n+1}} + \int_{t_{n+1}}^{t_n}\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\mathrm{d}t\right) = s_{t_n}\mathbf{x}_{t_n}.\end{aligned} \tag{40}$$

□

B.5. Proof of proposition 1

All following proofs are conducted in the context of a VE-SDE $\mathrm{d}\mathbf{x}_t = \sqrt{2t}\,\mathrm{d}\mathbf{w}_t$, i.e. $\sigma_t = t$ for notation simplicity, and the sampling trajectory always starts from $\hat{\mathbf{x}}_{t_N} \sim \mathcal{N}(0, T^2\mathbf{I})$ and ends at $\hat{\mathbf{x}}_{t_0}$. The PF-ODEs of the sampling trajectory and denoising trajectory are $\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t;t) = \frac{\mathbf{x}_t - r_{\boldsymbol{\theta}}(\mathbf{x}_t;t)}{t}$, and $\frac{\mathrm{d}r_{\boldsymbol{\theta}}(\mathbf{x}_t;t)}{\mathrm{d}t} = -t\frac{\mathrm{d}^2\mathbf{x}_t}{\mathrm{d}t^2}$, respectively.

Proposition 1. *Given the probability flow ODE (11) and a current position $\hat{\mathbf{x}}_{t_{n+1}}$, $n \in [0, N-1]$ in the sampling trajectory, the next position $\hat{\mathbf{x}}_{t_n}$ predicted by a k-th order Taylor expansion with the time step size $\sigma_{t_{n+1}} - \sigma_{t_n}$ is*

$$\hat{\mathbf{x}}_{t_n} = \frac{\sigma_{t_n}}{\sigma_{t_{n+1}}}\hat{\mathbf{x}}_{t_{n+1}} + \frac{\sigma_{t_{n+1}} - \sigma_{t_n}}{\sigma_{t_{n+1}}}\mathcal{R}_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_{n+1}}\right), \tag{41}$$

which is a convex combination of $\hat{\mathbf{x}}_{t_{n+1}}$ and the generalized denoising output

$$\mathcal{R}_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_{n+1}}\right) = r_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_{n+1}}\right) - \sum_{i=2}^{k}\frac{1}{i!}\frac{\mathrm{d}^{(i)}\mathbf{x}_t}{\mathrm{d}\sigma_t^{(i)}}\bigg|_{\hat{\mathbf{x}}_{t_{n+1}}}\sigma_{t_{n+1}}\left(\sigma_{t_n} - \sigma_{t_{n+1}}\right)^{i-1}. \tag{42}$$

In particular, we have $\mathcal{R}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}}) = r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}})$ for the Euler method ($k=1$), and $\mathcal{R}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}}) = r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}}) + \frac{\sigma_{t_n} - \sigma_{t_{n+1}}}{2}\frac{\mathrm{d}r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}})}{\mathrm{d}\sigma_t}$ for second-order numerical methods ($k=2$).

Proof. The k-th order Taylor expansion at $\hat{\mathbf{x}}_{t_{n+1}}$ is

$$\begin{aligned}\hat{\mathbf{x}}_{t_n} &= \sum_{i=0}^{k}\frac{1}{i!}\frac{\mathrm{d}^{(i)}\mathbf{x}_t}{\mathrm{d}t^{(i)}}\bigg|_{\hat{\mathbf{x}}_{t_{n+1}}}\left(t_n - t_{n+1}\right)^i\\ &= \hat{\mathbf{x}}_{t_{n+1}} + \left(t_n - t_{n+1}\right)\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}\bigg|_{\hat{\mathbf{x}}_{t_{n+1}}} + \sum_{i=2}^{k}\frac{1}{i!}\frac{\mathrm{d}^{(i)}\mathbf{x}_t}{\mathrm{d}t^{(i)}}\bigg|_{\hat{\mathbf{x}}_{t_{n+1}}}\left(t_n - t_{n+1}\right)^i\\ &= \hat{\mathbf{x}}_{t_{n+1}} + \frac{t_n - t_{n+1}}{t_{n+1}}\left(\hat{\mathbf{x}}_{t_{n+1}} - r_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_{n+1}}\right)\right) + \sum_{i=2}^{k}\frac{1}{i!}\frac{\mathrm{d}^{(i)}\mathbf{x}_t}{\mathrm{d}t^{(i)}}\bigg|_{\hat{\mathbf{x}}_{t_{n+1}}}\left(t_n - t_{n+1}\right)^i\\ &= \frac{t_n}{t_{n+1}}\hat{\mathbf{x}}_{t_{n+1}} + \frac{t_{n+1} - t_n}{t_{n+1}}\mathcal{R}_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_{n+1}}\right),\end{aligned} \tag{43}$$

where $\mathcal{R}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}}) = r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}}) - \sum_{i=2}^{k} \frac{1}{i!} \frac{\mathrm{d}^{(i)}\mathbf{x}_t}{\mathrm{d}t^{(i)}}\Big|_{\hat{\mathbf{x}}_{t_{n+1}}} t_{n+1}(t_n - t_{n+1})^{i-1}$. As for the first-order approximation ($k = 1$), we have $\mathcal{R}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}}) = r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}})$. As for the second-order approximation ($k = 2$), we have

$$\mathcal{R}_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_{n+1}}\right) = r_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_{n+1}}\right) - \frac{t_{n+1}\left(t_n - t_{n+1}\right)}{2} \frac{\mathrm{d}^2\mathbf{x}_t}{\mathrm{d}t^2}\Big|_{\hat{\mathbf{x}}_{t_{n+1}}} = r_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_{n+1}}\right) + \frac{t_n - t_{n+1}}{2} \frac{\mathrm{d}r_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_{n+1}}\right)}{\mathrm{d}t}. \tag{44}$$

□

Corollary 6. *The denoising output $r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_{n+1}})$ reflects the prediction made by a single Euler step from $\hat{\mathbf{x}}_{t_{n+1}}$ with the time step size t_{n+1}.*

Proof. The prediction of such an Euler step equals to

$$\hat{\mathbf{x}}_{t_{n+1}} + \left(0 - t_{n+1}\right)\left(\hat{\mathbf{x}}_{t_{n+1}} - r_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_{n+1}}\right)\right)/t_{n+1} = r_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_{n+1}}\right). \tag{45}$$

□

B.6. Proof of proposition 2

Lemma 5 (See section 3.1 in (Vershynin 2018)). *Given a high-dimensional isotropic Gaussian noise $\mathbf{z} \sim \mathcal{N}(0; \sigma^2\mathbf{I}_d)$, $\sigma > 0$, we have $\mathbb{E}\|\mathbf{z}\|^2 = \sigma^2 d$, and with high probability, z stays within a 'thin spherical shell': $\|\mathbf{z}\| = \sigma\sqrt{d} \pm O(1)$.*

Proof. We denote $\mathbf{z}_i$ as the ith dimension of random variable $\mathbf{z}$, then the expectation and variance is $\mathbb{E}[\mathbf{z}_i] = 0$, $\mathbb{V}[\mathbf{z}_i] = \sigma^2$, respectively. The fourth central moment is $\mathbb{E}\left[\mathbf{z}_i^4\right] = 3\sigma^4$. Additionally,

$$\begin{gathered}\mathbb{E}\left[\mathbf{z}_i^2\right] = \mathbb{V}\left[\mathbf{z}_i\right] + \mathbb{E}\left[\mathbf{z}_i\right]^2 = \sigma^2, \qquad \mathbb{E}\left[\|\mathbf{z}\|^2\right] = \mathbb{E}\left[\sum_{i=1}^{d} \mathbf{z}_i^2\right] = \sum_{i=1}^{d} \mathbb{E}\left[\mathbf{z}_i^2\right] = \sigma^2 d, \\ \mathbb{V}\left[\|\mathbf{z}\|^2\right] = \mathbb{E}\left[\|\mathbf{z}\|^4\right] - \left(\mathbb{E}\left[\|\mathbf{z}\|^2\right]\right)^2 = 2d\sigma^4,\end{gathered} \tag{46}$$

Then, we have

$$\mathbb{E}\left[\|\mathbf{x} + \mathbf{z}\|^2 - \|\mathbf{x}\|^2\right] = \mathbb{E}\left[\|\mathbf{z}\|^2 + 2\mathbf{x}^{\mathrm{T}}\mathbf{z}\right] = \mathbb{E}\left[\|\mathbf{z}\|^2\right] = \sigma^2 d. \tag{47}$$

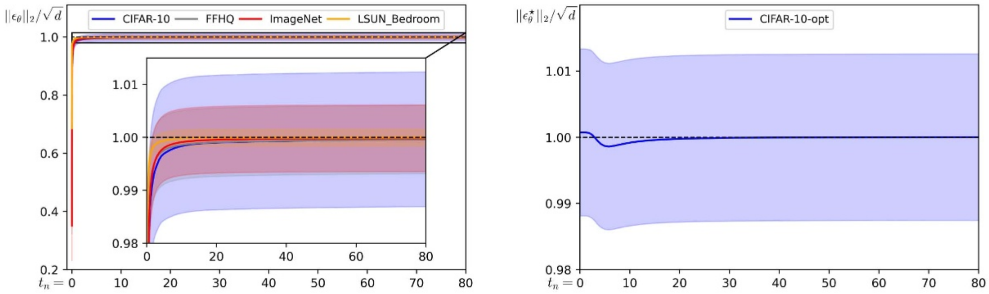
Furthermore, the standard deviation of $\|\mathbf{z}\|^2$ is $\sigma^2\sqrt{2d}$, which means

$$\|\mathbf{z}\|^2 = \sigma^2 d \pm \sigma^2\sqrt{2d} = \sigma^2 d \pm O\left(\sqrt{d}\right), \qquad \|\mathbf{z}\| \ = \sigma\sqrt{d} \pm O(1), \tag{48}$$

holds with high probability. □

Proposition 2. *The magnitude of $\boldsymbol{\epsilon}^{\star}_{\boldsymbol{\theta}}(\mathbf{x}_t; t)$ concentrates around $\sqrt{d}$, where d denotes the data dimension. Consequently, the total length of the sampling trajectory is approximately $\sigma_{\mathrm{T}}\sqrt{d}$, where σ_{T} denotes the maximum noise level.*

Proof. We next provide a sketch of proof. Suppose the data distribution lies in a smooth real low-dimensional manifold with the intrinsic dimension as m. According to the *Whitney embedding theorem* (Whitney 1936), it can be smoothly embedded in a real $2m$ Euclidean space. We then decompose each $\boldsymbol{\epsilon}^{\star}_{\boldsymbol{\theta}} \in \mathbb{R}^d$ vector as $\boldsymbol{\epsilon}^{\star}_{\boldsymbol{\theta},\|}$ and $\boldsymbol{\epsilon}^{\star}_{\boldsymbol{\theta},\perp}$, which



(a) The L^2 norm of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$.

(b) The L^2 norm of $\boldsymbol{\epsilon}^\star_{\boldsymbol{\theta}}$.

Figure 12. The optimal noise prediction satisfies $\|\boldsymbol{\epsilon}^\star_{\boldsymbol{\theta}}\|_2 \approx \sqrt{d}$ throughout the entire sampling process, as guaranteed by theoretical results. The actual noise prediction $\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\|_2$ also remains $\sqrt{d}$ for most time steps, but exhibits a noticeable shrinkage in the final stage, when the time step approaches zero. This norm shrinkage almost does not affect the trajectory length, as the discretized time steps in the final stage are extremely small.

are parallel and perpendicular to the $2m$ Euclidean space, respectively. Therefore, we have $\|\boldsymbol{\epsilon}^\star_{\boldsymbol{\theta}}\|_2 = \|\boldsymbol{\epsilon}^\star_{\boldsymbol{\theta},\|} + \boldsymbol{\epsilon}^\star_{\boldsymbol{\theta},\perp}\|_2 \geqslant \|\boldsymbol{\epsilon}^\star_{\boldsymbol{\theta},\perp}\|_2 \approx \sqrt{d-2m}$.

We provide a upper bound for the $\|\boldsymbol{\epsilon}^\star_{\boldsymbol{\theta}}\|_2$ below

$$\begin{aligned}\mathbb{E}_{p_t(\mathbf{x}_t)}\|\boldsymbol{\epsilon}^\star_{\boldsymbol{\theta}}\|_2 &= \mathbb{E}_{p_t(\mathbf{x}_t)}\|\frac{\mathbf{x}_t - r^\star_{\boldsymbol{\theta}}(\mathbf{x}_t)}{\sigma_t}\|_2 = \mathbb{E}_{p_t(\mathbf{x}_t)}\|\frac{\mathbf{x}_t - \mathbb{E}(\mathbf{x}_0|\mathbf{x}_t)}{\sigma_t}\|_2\\ &= \mathbb{E}_{p_t(\mathbf{x}_t)}\|\mathbb{E}\left(\frac{\mathbf{x}_t - \mathbf{x}_0}{\sigma_t}|\mathbf{x}_t\right)\|_2 = \mathbb{E}_{p_t(\mathbf{x}_t)}\|\mathbb{E}_{p_{t0}(\mathbf{x}_0|\mathbf{x}_t)}\boldsymbol{\epsilon}\|_2\\ &\leqslant \mathbb{E}_{p_t(\mathbf{x}_t)}\mathbb{E}_{p_{t0}(\mathbf{x}_0|\mathbf{x}_t)}\|\boldsymbol{\epsilon}\|_2 = \mathbb{E}_{p_0(\mathbf{x}_0)}\mathbb{E}_{p_{0t}(\mathbf{x}_t|\mathbf{x}_0)}\|\boldsymbol{\epsilon}\|_2\\ &\approx \sqrt{d} \qquad\qquad (\text{concentration of measure, lemma 5}).\end{aligned} \tag{49}$$

Additionally, the variance of $\|\boldsymbol{\epsilon}^\star_{\boldsymbol{\theta}}\|_2$ is relatively small.

$$\begin{aligned}\mathrm{Var}_{p_t(\mathbf{x}_t)}\|\boldsymbol{\epsilon}^\star_{\boldsymbol{\theta}}\|_2 &= \mathrm{Var}_{p_t(\mathbf{x}_t)}\|\mathbb{E}_{p_{t0}(\mathbf{x}_0|\mathbf{x}_t)}\boldsymbol{\epsilon}\|_2 = \mathbb{E}_{p_t(\mathbf{x}_t)}\|\mathbb{E}_{p_{t0}(\mathbf{x}_0|\mathbf{x}_t)}\boldsymbol{\epsilon}\|_2^2 - \left[\mathbb{E}_{p_t(\mathbf{x}_t)}\|\mathbb{E}_{p_{t0}(\mathbf{x}_0|\mathbf{x}_t)}\boldsymbol{\epsilon}\|_2\right]^2\\ &\leqslant \mathbb{E}_{p_t(\mathbf{x}_t)}\mathbb{E}_{p_{t0}(\mathbf{x}_0|\mathbf{x}_t)}\|\boldsymbol{\epsilon}\|_2^2 - (d-2m) = \mathbb{E}_{p_0(\mathbf{x}_0)}\mathbb{E}_{p_{0t}(\mathbf{x}_0|\mathbf{x}_t)}\|\boldsymbol{\epsilon}\|_2^2 - (d-2m)\\ &= d - (d-2m)\\ &= 2m\end{aligned} \tag{50}$$

Therefore, the standard deviation of $\|\boldsymbol{\epsilon}^\star_{\boldsymbol{\theta}}\|_2$ is upper bounded by $\sqrt{2m}$. Since $d \gg m$, we can conclude that in the optimal case, the magnitude of vector field is approximately constant, i.e. $\|\boldsymbol{\epsilon}^\star_{\boldsymbol{\theta}}\|_2 \approx \sqrt{d}$. The total sampling trajectory length is $\sum_{n=0}^{N-1}(\sigma_{t_{n+1}} - \sigma_{t_n})\|\boldsymbol{\epsilon}^\star_{\boldsymbol{\theta}}(\mathbf{x}_{t_{n+1}})\|_2 \approx \sigma_T\sqrt{d}$. Empirical verification is provided in figure 12. □

B.7. Proof of proposition 3

Proposition 3. *In deterministic sampling with the Euler method, the sample likelihood is non-decreasing, i.e.* $\forall n \in [1, N]$, *we have* $p_h(\hat{\mathbf{x}}_{t_{n-1}}) \geqslant p_h(\hat{\mathbf{x}}_{t_n})$ *and* $p_h(r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})) \geqslant p_h(\hat{\mathbf{x}}_{t_n})$

Table 7. PCA ratios computed by directly performing dimension reduction of the original sampling trajectory. This differs from PCA ratios in the main context that measures the reconstruction of the orthogonal complement of the displacement vector of each trajectory.

Dataset	Top-1	Top-2	Top-3
CIFAR-10 (32×32)	$99.997\,29\pm 0.001\,35\%$	$99.999\,98\pm 0.000\,18\%$	$99.999\,994\pm 0.000\,027\%$
FFHQ (64×64)	$99.998\,43\pm 0.000\,95\%$	$99.999\,99\pm 0.000\,16\%$	$99.999\,994\pm 0.000\,024\%$
ImageNet (64×64)	$99.998\,05\pm 0.001\,21\%$	$99.999\,98\pm 0.000\,19\%$	$99.999\,994\pm 0.000\,033\%$
SDv1.5, MS-COCO (64×64)	$99.941\,01\pm 0.012\,89\%$	$99.994\,32\pm 0.001\,50\%$	$99.999\,195\pm 0.000\,475\%$
LSUN Bedroom (256×256)	$99.999\,53\pm 0.000\,83\%$	$99.999\,99\pm 0.000\,13\%$	$99.999\,994\pm 0.000\,025\%$

in terms of the Gaussian KDE $p_h(\mathbf{x}) = (1/|\mathcal{I}|)\sum_{i\in\mathcal{I}}\mathcal{N}(\mathbf{x};\mathbf{y}_i,h^2\mathbf{I})$ *with any positive bandwidth h, assuming all samples in the sampling trajectory satisfy* $d_1(\hat{\mathbf{x}}_{t_n})\leqslant d_2(\hat{\mathbf{x}}_{t_n})$.

Proof. We first prove that given a random vector $\mathbf{v}$ falling within a sphere centered at the optimal denoising output $r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})$ with a radius of $\|r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})-\hat{\mathbf{x}}_{t_n}\|_2$, i.e. $\|r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})-\hat{\mathbf{x}}_{t_n}\|_2\geqslant\|\mathbf{v}\|_2$, the sample likelihood is non-decreasing from $\hat{\mathbf{x}}_{t_n}$ to $r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})+\mathbf{v}$, i.e. $p_h(r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})+\mathbf{v})\geqslant p_h(\hat{\mathbf{x}}_{t_n})$. Then, we provide two settings for $\mathbf{v}$ to finish the proof.

The increase of sample likelihood from $\hat{\mathbf{x}}_{t_n}$ to $r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})+\mathbf{v}$ in terms of $p_h(\mathbf{x})$ is

$$
\begin{aligned}
&p_h\left(r^\star_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_n}\right)+\mathbf{v}\right)-p_h\left(\hat{\mathbf{x}}_{t_n}\right)=\frac{1}{|\mathcal{I}|}\sum_i\left[\mathcal{N}\left(r^\star_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_n}\right)+\mathbf{v};\mathbf{y}_i,h^2\mathbf{I}\right)-\mathcal{N}\left(\hat{\mathbf{x}}_{t_n};\mathbf{y}_i,h^2\mathbf{I}\right)\right]\\
&\overset{\text{(i)}}{\geqslant}\frac{1}{2h^2|\mathcal{I}|}\sum_i\mathcal{N}\left(\hat{\mathbf{x}}_{t_n};\mathbf{y}_i,h^2\mathbf{I}\right)\left[\|\hat{\mathbf{x}}_{t_n}-\mathbf{y}_i\|_2^2-\|r^\star_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_n}\right)+\mathbf{v}-\mathbf{y}_i\|_2^2\right]\\
&=\frac{1}{2h^2|\mathcal{I}|}\sum_i\mathcal{N}\left(\hat{\mathbf{x}}_{t_n};\mathbf{y}_i,h^2\mathbf{I}\right)\left[\|\hat{\mathbf{x}}_{t_n}\|_2^2-2\hat{\mathbf{x}}_{t_n}^T\mathbf{y}_i-\|r^\star_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_n}\right)+\mathbf{v}\|_2^2+2\left(r^\star_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_n}\right)+\mathbf{v}\right)^T\mathbf{y}_i\right]\\
&\overset{\text{(ii)}}{=}\frac{1}{2h^2|\mathcal{I}|}\sum_i\mathcal{N}\left(\hat{\mathbf{x}}_{t_n};\mathbf{y}_i,h^2\mathbf{I}\right)\left[\|\hat{\mathbf{x}}_{t_n}\|_2^2-2\hat{\mathbf{x}}_{t_n}^T r^\star_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_n}\right)-\|r^\star_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_n}\right)+\mathbf{v}\|_2^2+2\left(r^\star_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_n}\right)+\mathbf{v}\right)^T r^\star_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_n}\right)\right]\\
&=\frac{1}{2h^2|\mathcal{I}|}\sum_i\mathcal{N}\left(\hat{\mathbf{x}}_{t_n};\mathbf{y}_i,h^2\mathbf{I}\right)\left[\|\hat{\mathbf{x}}_{t_n}\|_2^2-2\hat{\mathbf{x}}_{t_n}^{\mathrm{T}}r^\star_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_n}\right)+\|r^\star_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_n}\right)\|_2^2-\|\mathbf{v}\|_2^2\right]\\
&=\frac{1}{2h^2|\mathcal{I}|}\sum_i\mathcal{N}\left(\hat{\mathbf{x}}_{t_n};\mathbf{y}_i,h^2\mathbf{I}\right)\left[\|r^\star_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_n}\right)-\hat{\mathbf{x}}_{t_n}\|_2^2-\|\mathbf{v}\|_2^2\right]\geqslant 0,
\end{aligned}
\tag{51}
$$

where (i) uses the definition of convex function $f(\mathbf{x}_2)\geqslant f(\mathbf{x}_1)+f'(\mathbf{x}_1)(\mathbf{x}_2-\mathbf{x}_1)$ with $f(\mathbf{x})=\exp\left(-\frac{1}{2}\|\mathbf{x}\|_2^2\right)$, $\mathbf{x}_1=(\hat{\mathbf{x}}_{t_n}-\mathbf{y}_i)/h$ and $\mathbf{x}_2=(r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})+\mathbf{v}-\mathbf{y}_i)/h$; (ii) uses the relationship between two consecutive steps $\hat{\mathbf{x}}_{t_n}$ and $r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})$ in mean shift, i.e.

$$
r^\star_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_n}\right)=\mathbf{m}\left(\hat{\mathbf{x}}_{t_n}\right)=\sum_i\frac{\exp\left(-\|\hat{\mathbf{x}}_{t_n}-\mathbf{y}_i\|_2^2/2h^2\right)}{\sum_j\exp\left(-\|\hat{\mathbf{x}}_{t_n}-\mathbf{y}_j\|_2^2/2h^2\right)}\mathbf{y}_i,
\tag{52}
$$

which implies the following equation also holds

$$\sum_i \mathcal{N}\left(\hat{\mathbf{x}}_{t_n};\mathbf{y}_i,h^2\mathbf{I}\right)\mathbf{x}_i = \sum_i \mathcal{N}\left(\hat{\mathbf{x}}_{t_n};\mathbf{y}_i,h^2\mathbf{I}\right) r^{\star}_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_{t_n}\right). \tag{53}$$

Since $\|r^{\star}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}) - \hat{\mathbf{x}}_{t_n}\|_2 \geqslant \|\mathbf{v}\|_2$, or equivalently, $\|r^{\star}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}) - \hat{\mathbf{x}}_{t_n}\|_2^2 \geqslant \|\mathbf{v}\|_2^2$, we conclude that the sample likelihood monotonically increases from $\hat{\mathbf{x}}_{t_n}$ to $r^{\star}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}) + \mathbf{v}$ unless $\hat{\mathbf{x}}_{t_n} = r^{\star}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}) + \mathbf{v}$, in terms of the KDE $p_h(\mathbf{x}) = \frac{1}{|\mathcal{I}|}\sum_i \mathcal{N}(\mathbf{x};\mathbf{y}_i,h^2\mathbf{I})$ with any positive bandwidth h. We next provide two settings for $\mathbf{v}$, which trivially satisfy the condition $\|r^{\star}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}) - \hat{\mathbf{x}}_{t_n}\|_2 \geqslant \|\mathbf{v}\|_2$, and have the following corollaries:

- $p_h(r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})) \geqslant p_h(\hat{\mathbf{x}}_{t_n})$, when $\mathbf{v} = r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}) - r^{\star}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n})$.
- $p_h(\hat{\mathbf{x}}_{t_{n-1}}) \geqslant p_h(\hat{\mathbf{x}}_{t_n})$, when $\mathbf{v} = r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}) - r^{\star}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}) + \frac{t_{n-1}}{t_n}(\hat{\mathbf{x}}_{t_n} - r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_n}))$.

□

B.8. Proof of proposition 4

Proposition 4. *Suppose the data distribution is Gaussian* $p_{\mathrm{d}}(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$, *where* $\boldsymbol{\mu} \in \mathbb{R}^d$, $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ *is PSD with* $\mathrm{rank}(\boldsymbol{\Sigma}) = r \ll d$. *Let* $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}$ *denote the singular value decomposition (SVD), where* $\mathbf{U} \in \mathbb{R}^{d\times r}$ *contains eigenvectors* $\boldsymbol{u}_i$ *as columns, and* $\boldsymbol{\Lambda} \in \mathbb{R}^{r\times r}$ *is diagonal with eigenvalues* λ_i, $i \in [1,r]$. *In this setting, the PF-ODE solution* $\boldsymbol{x}_t$ *can be decomposed into the final sample* $\boldsymbol{x}_0$, *a scaled reverse displacement vector* $\mathbf{x}_{\mathrm{T}} - \mathbf{x}_0$, *and a trajectory residual* $\Delta_k(t)$*:*

$$\begin{aligned} &\mathbf{x}_t = \mathbf{x}_0 + \frac{\sigma_t}{\sigma_{\mathrm{T}}}\left(\mathbf{x}_{\mathrm{T}} - \mathbf{x}_0\right) + \Delta_k\left(t\right), \qquad \Delta_k\left(t\right) = \sum_{k=1}^{r}\varphi_k\left(t\right)\mathbf{u}_k^{\mathrm{T}}\left(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu}\right)\mathbf{u}_k, \\ &\varphi_k\left(t\right) = \sqrt{\frac{\lambda_k + \sigma_t^2}{\lambda_k + \sigma_{\mathrm{T}}^2}} - \sqrt{\frac{\lambda_k}{\lambda_k + \sigma_{\mathrm{T}}^2}} - \frac{\sigma_t}{\sigma_{\mathrm{T}}}\left(1 - \sqrt{\frac{\lambda_k}{\lambda_k + \sigma_{\mathrm{T}}^2}}\right). \end{aligned} \tag{54}$$

Proof. The score function is $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t) = (\boldsymbol{\Sigma} + \sigma_t^2\mathbf{I})^{-1}(\boldsymbol{\mu} - \mathbf{x}_t)$, and the corresponding linear PF-ODE is $\mathrm{d}\mathbf{x}_t/\mathrm{d}\sigma_t = \sigma_t(\boldsymbol{\Sigma} + \sigma_t^2\mathbf{I})^{-1}(\mathbf{x}_t - \boldsymbol{\mu})$. The solution of $\mathbf{x}_t$

$$\begin{aligned} \mathbf{x}_t &= \boldsymbol{\mu} + \exp\left(\int_{\mathrm{T}}^{t}\sigma_s\left(\boldsymbol{\Sigma} + \sigma_s^2\mathbf{I}\right)^{-1}\mathrm{d}s\right)\left(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu}\right) \\ &\overset{\text{(i)}}{=} \boldsymbol{\mu} + \exp\left(\int_{\mathrm{T}}^{t}\frac{1}{\sigma_s}\left[\mathbf{I} - \mathbf{U}\,\mathrm{diag}\left[\frac{\lambda_k}{\lambda_k + \sigma_s^2}\right]\mathbf{U}^T\right]\mathrm{d}s\right)\left(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu}\right) \\ &\overset{\text{(ii)}}{=} \boldsymbol{\mu} + \exp\left(\int_{\mathrm{T}}^{t}\mathbf{Q}\begin{bmatrix}\mathrm{diag}\left(\frac{\sigma_s}{\lambda_k + \sigma_s^2}\right) & 0 \\ 0 & \frac{1}{\sigma_s}\mathbf{I}\end{bmatrix}\mathbf{Q}^{\mathrm{T}}\mathrm{d}s\right)\left(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu}\right) \\ &\overset{\text{(iii)}}{=} \boldsymbol{\mu} + \mathbf{Q}\exp\left(\int_{\mathrm{T}}^{t}\begin{bmatrix}\mathrm{diag}\left(\frac{\sigma_s}{\lambda_k + \sigma_s^2}\right) & 0 \\ 0 & \frac{1}{\sigma_s}\mathbf{I}\end{bmatrix}\mathrm{d}s\right)\mathbf{Q}^{\mathrm{T}}\left(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu}\right) \\ &= \boldsymbol{\mu} + \frac{\sigma_t}{\sigma_{\mathrm{T}}}\left(\mathbf{I} - \mathbf{U}\mathbf{U}^{\mathrm{T}}\right)\left(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu}\right) + \sum_{k=1}^{r}\sqrt{\frac{\lambda_k + \sigma_t^2}{\lambda_k + \sigma_{\mathrm{T}}^2}}\mathbf{u}_k\mathbf{u}_k^{\mathrm{T}}\left(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu}\right) \end{aligned}$$

$$
\begin{aligned}
&= \boldsymbol{\mu} + \frac{\sigma_t}{\sigma_{\mathrm{T}}}\left(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu}\right) + \left(1 - \frac{\sigma_t}{\sigma_{\mathrm{T}}}\right)\left(\sum_{k=1}^{r}\sqrt{\frac{\lambda_k}{\lambda_k + \sigma_{\mathrm{T}}^2}}\mathbf{u}_k\mathbf{u}_k^{\mathrm{T}}\left(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu}\right)\right) + \Delta_k\left(t\right) \\
&= \mathbf{x}_0 + \frac{\sigma_t}{\sigma_{\mathrm{T}}}\left(\mathbf{x}_{\mathrm{T}} - \mathbf{x}_0\right) + \Delta_k\left(t\right),
\end{aligned}
\tag{55}
$$

where (i) applies the Woodbury identity, (ii) denotes $\mathbf{Q} = [\mathbf{U}, \mathbf{U}_{\perp}] \in \mathbb{R}^{d\times d}$, $\mathbf{Q}\mathbf{Q}^{\mathrm{T}} = \mathbf{I}$, (iii) exchanges the order of operators given commuting matrices (simultaneous diagonalization). The trajectory deviation $\Delta_k(t)$:

$$
\begin{aligned}
\Delta_k\left(t\right) &= \mathbf{U}\operatorname{diag}\left[-\frac{\sigma_t}{\sigma_{\mathrm{T}}} + \sqrt{\frac{\lambda_k + \sigma_t^2}{\lambda_k + \sigma_{\mathrm{T}}^2}} + \left(\frac{\sigma_t}{\sigma_{\mathrm{T}}} - 1\right)\sqrt{\frac{\lambda_k}{\lambda_k + \sigma_{\mathrm{T}}^2}}\right]\mathbf{U}^{\mathrm{T}}\left(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu}\right) \\
&= \sum_{k=1}^{r}\varphi_k\left(t\right)\mathbf{u}_k^{\mathrm{T}}\left(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu}\right)\mathbf{u}_k, \quad \varphi_k\left(t\right) = -\frac{\sigma_t}{\sigma_{\mathrm{T}}} \\
&\quad + \sqrt{\frac{\lambda_k + \sigma_t^2}{\lambda_k + \sigma_{\mathrm{T}}^2}} + \left(\frac{\sigma_t}{\sigma_{\mathrm{T}}} - 1\right)\sqrt{\frac{\lambda_k}{\lambda_k + \sigma_{\mathrm{T}}^2}}.
\end{aligned}
\tag{56}
$$

□

Since $\mathbf{u}_i^{\mathrm{T}}\mathbf{u}_j = 0$ for $i \neq j$, we have the squared norm of the trajectory residual as follows

$$
\begin{aligned}
h\left(t\right) := \|\Delta_k\left(t\right)\|_2^2 &= \left(\sum_{i=1}^{r}\varphi_i\left(t\right)\mathbf{u}_i\mathbf{u}_i^{\mathrm{T}}\left(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu}\right)\right)\left(\sum_{j=1}^{r}\varphi_j\left(t\right)\mathbf{u}_j\mathbf{u}_j^{\mathrm{T}}\left(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu}\right)\right) \\
&= \sum_{k=1}^{r}\varphi_k\left(t\right)^2\left(\mathbf{u}_k^{\mathrm{T}}\left(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu}\right)\right)^2.
\end{aligned}
\tag{57}
$$

Since $\mathbf{x}_{\mathrm{T}} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma} + \sigma_{\mathrm{T}}^2\mathbf{I})$, then

$$
\begin{aligned}
\mathbf{U}^{\mathrm{T}}\left(\mathbf{x}_{\mathrm{T}} - \boldsymbol{\mu}\right) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{U}^{\mathrm{T}}\left(\boldsymbol{\Sigma} + \sigma_{\mathrm{T}}^2\mathbf{I}\right)\mathbf{U}\right) &= \mathcal{N}\left(\mathbf{0}, \mathbf{U}^{\mathrm{T}}\left(\boldsymbol{\Sigma} + \sigma_{\mathrm{T}}^2\mathbf{I}_n\right)\mathbf{U}\right) \\
&= \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Lambda} + \sigma_{\mathrm{T}}^2\mathbf{I}_d\right).
\end{aligned}
\tag{58}
$$

$$
h\left(t\right) = \sum_{k=1}^{r}\varphi_k\left(t\right)^2 v_k^2, \quad v_k \sim \mathcal{N}\left(0, \lambda_k + \sigma_{\mathrm{T}}^2\right).
\tag{59}
$$

We denote $s_k(t) = \sqrt{\frac{\lambda_k + \sigma_t^2}{\lambda_k + \sigma_{\mathrm{T}}^2}}$, then $s_k(T) = 1$, and

$$
\begin{aligned}
s_k'\left(t\right) &= \frac{\sigma_t}{\left(\lambda_k + \sigma_{\mathrm{T}}^2\right)s_k\left(t\right)}, \quad s_k''\left(t\right) = \frac{\lambda_k}{\left(\lambda_k + \sigma_t^2\right)^{3/2}\sqrt{\lambda_k + \sigma_{\mathrm{T}}^2}} \\
&= \frac{\lambda_k}{\left(\lambda_k + \sigma_{\mathrm{T}}^2\right)s_k\left(t\right)^3} > 0 \qquad \left(\lambda_k > 0\right),
\end{aligned}
\tag{60}
$$

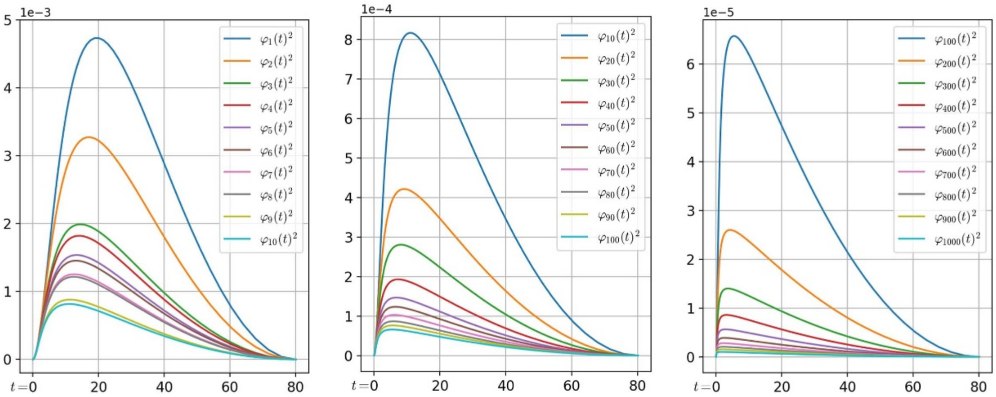


Figure 13. Variation of $\varphi_k(t)^2$ with respect to k and t.

$$
\begin{aligned}
\varphi_k(t) &= s_k(t) - \frac{\sigma_t}{\sigma_\mathrm{T}} + \left(\frac{\sigma_t}{\sigma_\mathrm{T}} - 1\right) s_k(0) = s_k(t) - s_k(0) - \frac{\sigma_t}{\sigma_\mathrm{T}}\left(1 - s_k(0)\right), \\
\varphi_k'(t) &= \frac{\sigma_t}{\left(\lambda_k + \sigma_\mathrm{T}^2\right) s_k(t)} - \frac{1 - s_k(0)}{\sigma_\mathrm{T}} = s_k'(t) - \frac{1 - s_k(0)}{\sigma_\mathrm{T}}, \\
\varphi_k''(t) &= s_k''(t) > 0 \quad \text{for each} \quad k.
\end{aligned}
\tag{61}
$$

Therefore, φ is a strict convex function for $t \in [0,T]$ and must have *one unique minimum*. By setting $\varphi_k'(t_\mathrm{min}) = 0$, we have $t_\mathrm{min} = \sqrt{\frac{\sqrt{\lambda_k(\lambda_k+\sigma_\mathrm{T}^2)}-\lambda_k}{2}}$, and

$$
\begin{aligned}
s_k(t_\mathrm{min}) &= \frac{\sqrt{\left(\sqrt{\lambda_k}\right)\left(\sqrt{\lambda_k} + \sqrt{\lambda_k + \sigma_\mathrm{T}^2}\right)}}{\sqrt{2}\sqrt{\lambda_k + \sigma_\mathrm{T}^2}}, \\
\varphi_k(t_\mathrm{min}) &= \sqrt{\frac{\lambda_k}{\lambda_k + \sigma_\mathrm{T}^2}} \left(\sqrt{\frac{2\sqrt{\lambda_k}}{\sqrt{\lambda_k} + \sqrt{\lambda_k + \sigma_\mathrm{T}^2}}} - 1\right).
\end{aligned}
\tag{62}
$$

The visualization of $\varphi_k(t)^2$ and $h_k(t)$ with respect to kth eigenvalue ($k \in [1, 1000]$) and t ($t \in [0, 80]$) are provided in figures 13 and 14. Since $\Delta_k(t)$ is approximately orthogonal to the displacement vector $\mathbf{x}_0 - \mathbf{x}_\mathrm{T}$, the differences between trajectory deviations and trajectory residuals are minor, as shown in figure 15.

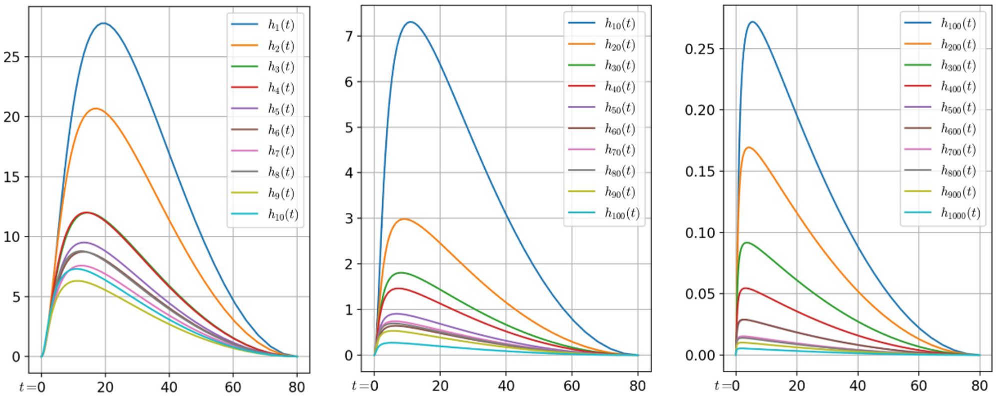


Figure 14. Variation of $h_k(t) = \varphi_k(t)^2(\mathbf{u}_k^\mathrm{T}(\mathbf{x}_\mathrm{T} - \mu))^2$ with respect to k and t.

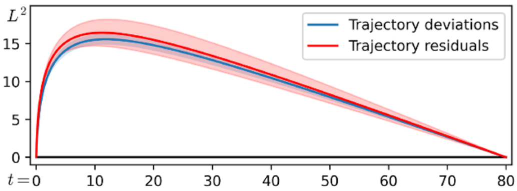
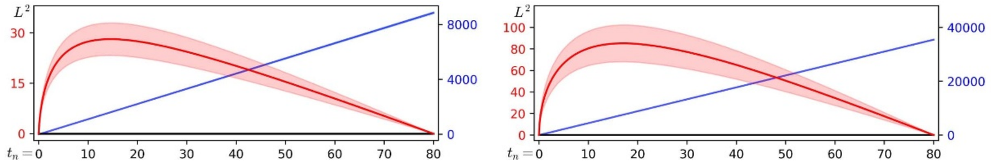


Figure 15. Comparison between trajectory deviations and trajectory residuals using low-rank Gaussian approximation.



(a) FFHQ (64×64).

(b) LSUN Bedroom (256×256).

Figure 16. Trajectory deviation (1-D projection) on unconditional generation.

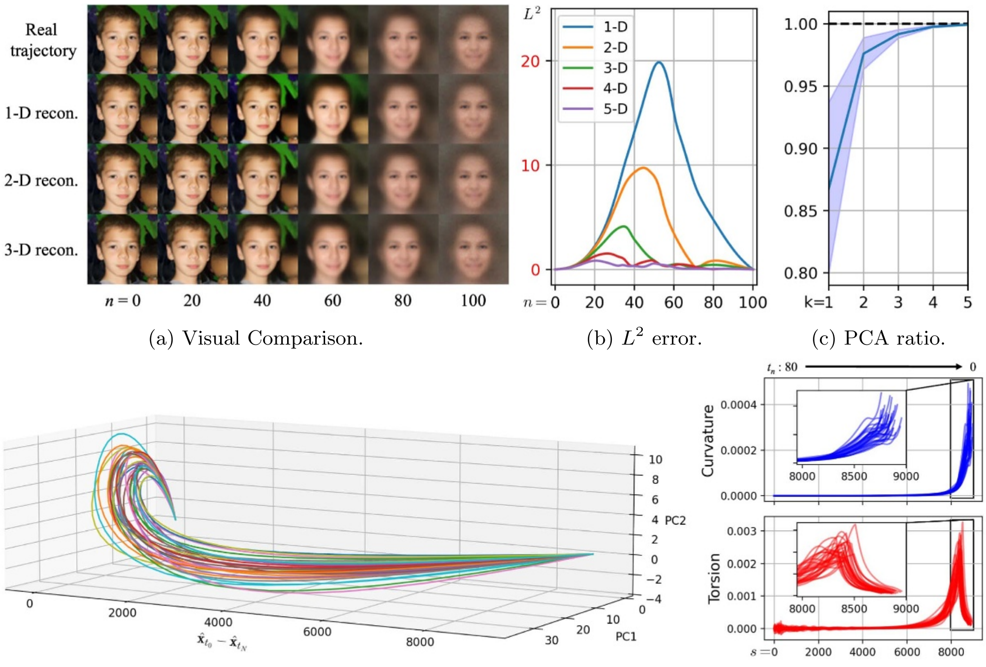
Appendix C. Additional results

C.1. Visualization of sampling trajectories

Figures 16(a) and (b) provide more experiments about the 1-D trajectory projection on FFHQ and LSUN Bedroom. Figure 17 provides more results about Multi-D projections on FFHQ. Figures 18 and 19 provide more results on CIFAR-10 using a Gaussian or mixture of Gaussians model. Figure 20 visualizes more generated samples on three datasets.

C.2. Diagnosis of score deviation

In this section, we simulate four new trajectories based on the optimal denoising output $r^\star_{\boldsymbol{\theta}}(\cdot)$ to monitor the score deviation from the optimum. We denote the *optimal sampling trajectory* as $\{\hat{\mathbf{x}}^\star_{t_n}\}_{n=0}^N$, where we generate samples as the standard sampling



(a) Visual Comparison.

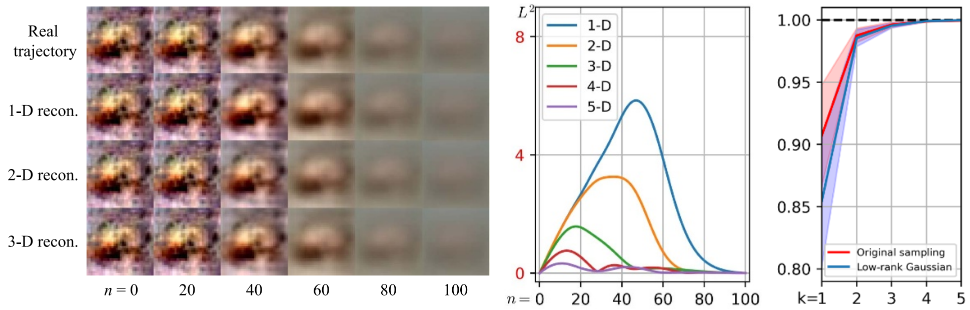
(b) L^2 error.

(c) PCA ratio.

(d) Unconditional generation, FFHQ (64× 64)

(e) Curvature/Torsion.

Figure 17. Trajectory reconstruction, visualization and statistics on FFHQ. Figure (a) is generated by EDM (Karras *et al* 2022).



(a) Visual Comparison.

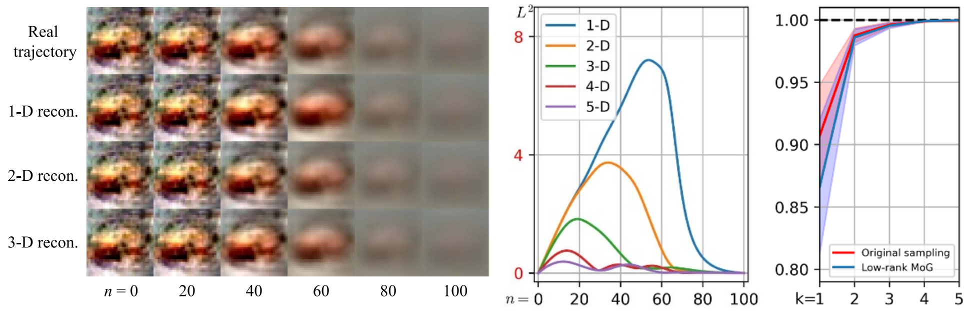
(b) L^2 error.

(c) PCA ratio.

Figure 18. Trajectory reconstruction, visualization and statistics on CIFAR-10 using low-rank Gaussian.

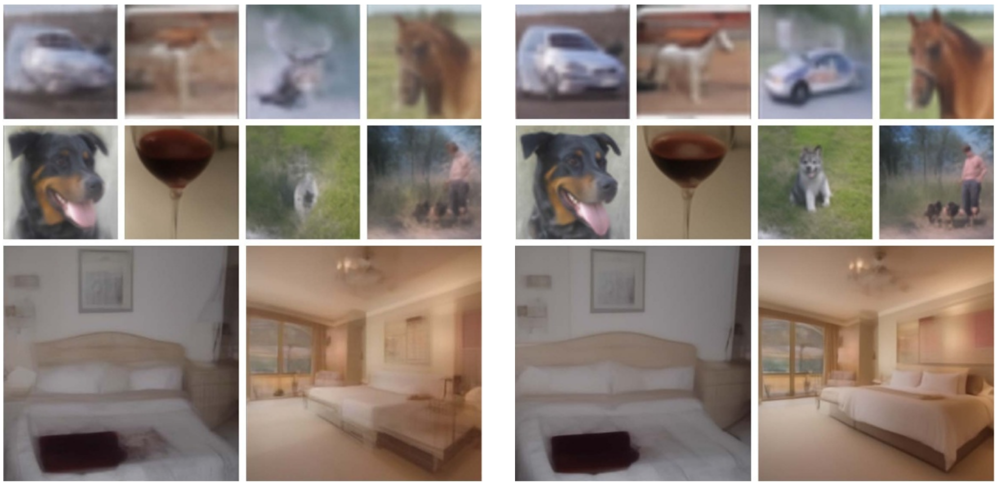
trajectory $\{\hat{\mathbf{x}}_{t_n}\}_{n=0}^{N}$ with the same time schedule $\Gamma=\{t_0=\epsilon,\cdots,t_N=T\}$, but adopt optimal denoising output $r^\star_{\boldsymbol{\theta}}(\cdot)$ rather than denoising output $r_{\boldsymbol{\theta}}(\cdot)$ for score estimation. The other three trajectories are simulated by tracking the (optimal) denoising output of each sample in $\{\hat{\mathbf{x}}^\star_t\}$ or $\{\hat{\mathbf{x}}_t\}$, and designated as $\{r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}^\star_t)\}$, $\{r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}^\star_t)\}$, $\{r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t)\}$. According to (17) and $t_0=0$, we have $\hat{\mathbf{x}}^\star_{t_0}=r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}^\star_{t_1})$, and similarly, $\hat{\mathbf{x}}_{t_0}=r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_1})$. As $t\to 0$, $r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}^\star_t)$ and $r^\star_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t)$ serve as the approximate nearest neighbors of $\hat{\mathbf{x}}^\star_t$ and $\hat{\mathbf{x}}_t$ to the real data, respectively.

We calculate the deviation of denoising output to quantify the score deviation across all time steps using the L^2 distance, though they should differ by a factor t^2, and have



(a) Visual Comparison. (b) L^2 error. (c) PCA ratio.

Figure 19. Trajectory reconstruction, visualization and statistics on CIFAR-10 using low-rank mixture of Gaussians.



(a) DDIM, NFE = 5. (b) DDIM + GITS, NFE = 5.

Figure 20. The visual comparison of samples generated by DDIM and DDIM + GITS (1st row: CIFAR-10, 2nd row: ImageNet 64×64, 3rd row: LSUN Bedroom). Figures are generated by EDM (Karras *et al* 2022).

the following observation: *The learned score is well-matched to the optimal score in the large-noise region, otherwise, they may diverge or almost coincide depending on different regions.* In fact, our learned score has to moderately diverge from the optimum to guarantee the generative ability. Otherwise, the ODE-based sampling reduces to an approximate (single-step) annealed mean shift for global mode-seeking, and simply replays the dataset. As shown in figure 21, the nearest sample of $\hat{\mathbf{x}}^{\star}_{t_0}$ to the real data is almost the same as itself, which indicates the optimal sampling trajectory has a very limited ability to synthesize novel samples. Empirically, score deviation in a small region is sufficient to bring forth a decent generative ability.

From the comparison of $\{r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}^{\star}_t)\}$, $\{r^{\star}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}^{\star}_t)\}$ sequences in figures 21 and 22, we can clearly see that *along the optimal sampling trajectory*, the deviation between the learned denoising output $r_{\boldsymbol{\theta}}(\cdot)$ and its optimal counterpart $r^{\star}_{\boldsymbol{\theta}}(\cdot)$ behaves differently in three

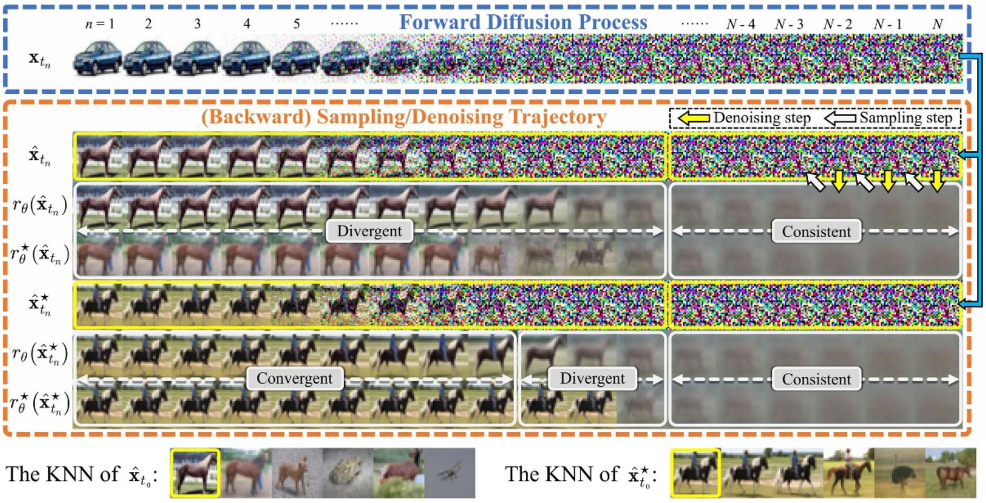

Forward Diffusion Process
$n=1$
2
3
4
5
……
……
$N-4$
$N-3$
$N-2$
$N-1$
N
$\mathbf{x}_{t_n}$
(Backward) Sampling/Denoising Trajectory
Denoising step
Sampling step
$\hat{\mathbf{x}}_{t_n}$
$r_\theta(\hat{\mathbf{x}}_{t_n})$
$r_\theta^\star(\hat{\mathbf{x}}_{t_n})$
Divergent
Consistent
$\hat{\mathbf{x}}_{t_n}^\star$
$r_\theta(\hat{\mathbf{x}}_{t_n}^\star)$
$r_\theta^\star(\hat{\mathbf{x}}_{t_n}^\star)$
Convergent
Divergent
Consistent
The KNN of $\hat{\mathbf{x}}_{t_0}$:
The KNN of $\hat{\mathbf{x}}_{t_0}^\star$:

Figure 21. *Top*: We visualize a forward diffusion process of a randomly-selected image to obtain its encoding $\hat{\mathbf{x}}_{t_N}$ (first row) and simulate multiple trajectories starting from this encoding (other rows). *Bottom*: The k-nearest neighbors ($k=5$) of $\hat{\mathbf{x}}_{t_0}$ and $\hat{\mathbf{x}}_{t_0}^\star$ to real samples in the dataset. Figures are generated by EDM (Karras *et al* 2022).

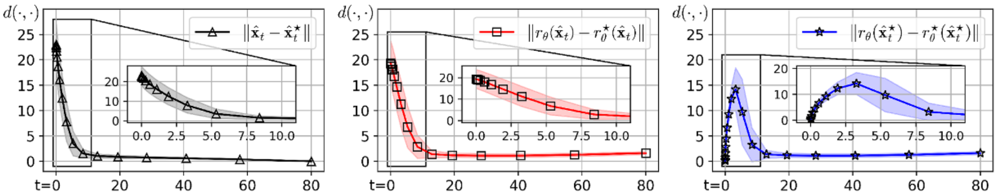


Figure 22. The deviation (Euclidean distance) of outputs from their corresponding optima.

successive regions: The deviation starts off as almost negligible (about $10 < t \leqslant 80$), gradually increases (about $3 < t \leqslant 10$), and then drops down to a low level once again (about $0 \leqslant t \leqslant 3$). This phenomenon was also validated by a recent work (Xu *et al* 2023) with a different perspective. We further observe that *along the sampling trajectory*, this phenomenon disappears and the score deviation keeps increasing (see $\{r_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t)\}$, $\{r_{\boldsymbol{\theta}}^\star(\hat{\mathbf{x}}_t)\}$ sequences in figures 21 and 22). Additionally, samples in the latter half of $\{r_{\boldsymbol{\theta}}^\star(\hat{\mathbf{x}}_t)\}$ appear almost the same as the nearest sample of $\hat{\mathbf{x}}_{t_0}$ to the real data, as shown in figure 21. This indicates that our score-based model strives to explore novel regions, and synthetic samples in the sampling trajectory are quickly attracted to a real-data mode but do not fall into it.

C.3. Comparision of time schedule and sample quality

Time schedule. The uniform schedule is commonly used with the DDPM (Ho *et al* 2020) backbone. Following EDMs (Karras *et al* 2022), we rewrite this schedule from its original range $[\epsilon_s, 1]$ to $[t_0, t_N]$, where $\epsilon_s = 0.001$, $t_0 = 0.002$ and $t_N = 80$.

Table 8. Sample quality in terms of FID (Heusel *et al* 2017) on four datasets (resolutions ranging from 32 × 32 to 256 × 256).

			NFE							
METHOD	Coeff	AFS^a	3	4	5	6	7	8	9	10
CIFAR-10 32×32 (Krizhevsky and Hinton 2009)										
DDIM (Song *et al* 2021a)	—	×	93.36	66.76	49.66	35.62	27.93	22.32	18.43	15.69
DDIM + GITS	1.10	×	88.68	46.88	32.50	22.04	16.76	13.93	11.57	10.09
DDIM + GITS (default)	1.15	79.67	×	43.07	28.05	21.04	16.35	13.30	11.62	10.37
DDIM + GITS	1.20	×	77.22	43.16	29.06	22.69	18.91	14.22	12.03	11.38
iPNDM (Zhang and Chen 2023)	—	×	47.98	24.82	13.59	7.05	5.08	3.69	3.17	2.77
iPNDM + GITS	1.10	×	51.31	17.19	12.90	5.98	6.62	4.36	3.59	3.14
iPNDM + GITS (default)	1.15	×	43.89	15.10	8.38	4.88	5.11	3.24	2.70	2.49
iPNDM + GITS	1.20	×	42.06	15.85	9.33	7.13	5.95	3.28	2.81	2.71
iPNDM + GITS	1.10	✓	34.22	11.99	12.44	6.08	6.20	3.53	3.48	2.91
iPNDM + GITS	1.15	✓	29.63	11.23	8.08	4.86	4.46	2.92	2.46	**2.27**
iPNDM + GITS	1.20	✓	**25.98**	**10.11**	**6.77**	**4.29**	**3.43**	**2.70**	**2.42**	2.28
FFHQ 64×64 (Karras *et al* 2019)										
DDIM (Song *et al* 2021a)	—	×	78.21	57.48	43.93	35.22	28.86	24.39	21.01	18.37
DDIM + GITS	1.10	×	62.70	43.12	31.01	24.62	20.35	17.19	14.71	13.01
DDIM + GITS (default)	1.15	×	60.84	40.81	29.80	23.67	19.41	16.60	14.46	13.06
DDIM + GITS	1.20	×	59.64	40.56	30.29	23.88	20.07	17.36	15.40	14.05
iPNDM (Zhang and Chen 2023)	—	×	45.98	28.29	17.17	10.03	7.79	5.52	4.58	3.98
iPNDM + GITS	1.10	×	34.82	18.75	13.07	7.79	8.30	4.76	5.36	3.47
iPNDM + GITS (default)	1.15	×	33.09	17.04	11.22	7.00	6.72	4.52	4.33	3.62
iPNDM + GITS	1.20	×	31.70	16.87	10.83	7.10	6.37	5.78	4.81	4.39
iPNDM + GITS	1.10	✓	33.19	19.88	12.90	8.29	7.50	4.26	4.95	**3.13**
iPNDM + GITS	1.15	✓	30.39	15.78	10.15	6.86	5.97	**4.09**	**3.76**	3.24
iPNDM + GITS	1.20	✓	**26.41**	**13.59**	**8.85**	**6.39**	**5.36**	4.91	3.89	3.51
ImageNet 64×64 (Russakovsky *et al* 2015)										
DDIM (Song *et al* 2021a)	—	×	82.96	58.43	43.81	34.03	27.46	22.59	19.27	16.72
DDIM + GITS	1.10	×	60.11	36.23	27.31	20.82	16.41	14.16	11.95	10.84
DDIM + GITS (default)	1.15	×	57.06	35.07	24.92	19.54	16.01	13.79	12.17	10.83
DDIM + GITS	1.20	×	54.24	34.27	24.67	19.46	16.66	14.15	13.41	11.87
iPNDM (Zhang and Chen 2023)	—	×	58.53	33.79	18.99	12.92	9.17	7.20	5.91	5.11
iPNDM + GITS	1.10	×	36.18	19.64	13.18	9.58	7.68	6.44	5.24	4.59
iPNDM + GITS (default)	1.15	×	34.47	18.95	10.79	8.43	6.83	5.82	4.96	4.48
iPNDM + GITS	1.20	×	32.70	18.59	11.04	9.23	7.18	6.20	5.50	5.08
iPNDM + GITS	1.10	✓	31.50	21.50	13.73	10.74	7.99	6.88	5.29	4.64
iPNDM + GITS	1.15	✓	28.01	18.28	10.28	8.68	6.76	5.90	4.81	**4.40**
iPNDM + GITS	1.20	✓	**26.41**	**16.41**	**9.85**	**8.39**	**6.44**	**5.64**	**4.79**	4.47

(Continued.)

Table 8. (Continued.)

METHOD	Coeff	AFS[a]	NFE 3	4	5	6	7	8	9	10
LSUN Bedroom 256×256 (Yu *et al* 2015) (pixel-space)										
DDIM (Song *et al* 2021a)	—	×	86.13	54.45	34.34	25.25	19.49	15.71	13.26	11.42
DDIM + GITS	1.05	×	81.77	36.89	27.46	18.78	13.60	12.23	10.29	8.77
DDIM + GITS (default)	1.10	×	61.85	35.12	22.04	16.54	13.58	11.20	9.82	9.04
DDIM + GITS	1.15	×	60.11	31.02	23.65	17.18	13.42	12.61	10.89	10.57
iPNDM (Zhang and Chen 2023)	—	×	80.99	43.90	26.65	20.73	13.80	11.78	8.38	5.57
iPNDM + GITS	1.05	×	59.02	24.71	19.08	12.77	**8.19**	**6.67**	**5.58**	**4.83**
iPNDM + GITS (default)	1.10	×	45.75	22.98	**15.85**	**10.41**	8.63	7.31	6.01	5.28
iPNDM + GITS	1.15	×	**44.78**	**21.67**	17.29	11.52	9.59	8.82	7.22	5.97

[a] After obtaining the DP schedule, we could further optimize the first time step with AFS, using the same 'warmup' samples. The default setting in our main submission does not use AFS and keeps the coefficient in DP as 1.1 for LSUN Bedroom and 1.15 otherwise. Although the performance can be further improved by carefully tuning the coefficient and using AFS as shown above.

We first uniformly sample τ_n $(n \in [0,N])$ from $[\epsilon_s, 1]$ and then calculate t_n by $t_n = \sqrt{\exp(\frac{1}{2}\beta_d {\tau_n}^2 + \beta_{\min}\tau_n) - 1}$, where $\beta_d = \frac{2}{\epsilon_s - 1}\frac{\log(1+{t_0}^2)}{\epsilon_s} - \log(1+{t_N}^2)$, $\beta_{\min} = \log(1+{t_N}^2) - \frac{1}{2}\beta_d$.

The logSNR time schedule is proposed for fast sampling in DPM-Solver (Lu *et al* 2022a). We first uniformly sample λ_n $(n \in [0,N])$ from $[\lambda_{\min}, \lambda_{\max}]$ where $\lambda_{\min} = -\log t_N$ and $\lambda_{\max} = -\log t_0$. The logSNR schedule is given by $t_n = e^{-\lambda_n}$.

The polynomial time schedule $t_n = (t_0^{1/\rho} + \frac{n}{N}(t_N^{1/\rho} - t_0^{1/\rho}))^\rho$ is proposed in EDM (Karras *et al* 2022), where $t_0 = 0.002$, $t_N = 80$, $n \in [0,N]$, and $\rho = 7$.

The optimized time schedules for SDv1.5 in figure 11 include

- AYS (Sabour *et al* 2024): [999, 850, 736, 645, 545, 455, 343, 233, 124, 24, 0].
- GITS: [999, 783, 632, 483, 350, 233, 133, 67, 33, 17, 0].

Furthermore, we do observe strong similarity in the optimal time schedules across different models and datasets, although the exact trajectory shapes vary slightly due to the influence of the specific models and datasets. (see figures 3, 5, 16, 17). We then conducted cross-dataset experiments by directly applying a time schedule optimized on one dataset (e.g. CIFAIR-10) to others (e.g. FFHQ and ImageNet). The results are reported in table 10, where each column corresponds to the dataset used to optimize the time schedule. We can see that the results within each row remain relatively stable, which confirms that trajectory regularity is consistent across different datasets.

Table 9. Comparison of various time schedules on CIFAR-10.

NFE	TIME SCHEDULE	FID
Uniform		
3	[80.0000, 6.9503, 1.2867, 0.0020]	50.44
4	[80.0000, 11.7343, 2.8237, 0.8565, 0.0020]	18.73
5	[80.0000, 16.5063, 4.7464, 1.7541, 0.6502, 0.0020]	17.34
6	[80.0000, 20.9656, 6.9503, 2.8237, 1.2867, 0.5272, 0.0020]	9.75
7	[80.0000, 25.0154, 9.3124, 4.0679, 2.0043, 1.0249, 0.4447, 0.0020]	12.50
8	[80.0000, 28.6496, 11.7343, 5.4561, 2.8237, 1.5621, 0.8565, 0.3852, 0.0020]	7.56
9	[80.0000, 31.8981, 14.1472, 6.9503, 3.7419, 2.1599, 1.2867, 0.7382, 0.3401, 0.0020]	10.60
10	[80.0000, 34.8018, 16.5063, 8.5141, 4.7464, 2.8237, 1.7541, 1.0985, 0.6502, 0.3047, 0.0020]	7.35
LogSNR		
3	[80.0000, 2.3392, 0.0684, 0.0020]	88.38
4	[80.0000, 5.6569, 0.4000, 0.0283, 0.0020]	35.59
5	[80.0000, 9.6090, 1.1542, 0.1386, 0.0167, 0.0020]	19.87
6	[80.0000, 13.6798, 2.3392, 0.4000, 0.0684, 0.0117, 0.0020]	10.68
7	[80.0000, 17.6057, 3.8745, 0.8527, 0.1876, 0.0413, 0.0091, 0.0020]	6.56
8	[80.0000, 21.2732, 5.6569, 1.5042, 0.4000, 0.1064, 0.0283, 0.0075, 0.0020]	4.74
9	[80.0000, 24.6462, 7.5929, 2.3392, 0.7207, 0.2220, 0.0684, 0.0211, 0.0065, 0.0020]	3.53
10	[80.0000, 27.7258, 9.6090, 3.3302, 1.1542, 0.4000, 0.1386, 0.0480, 0.0167, 0.0058, 0.0020]	2.94
Polynomial ($\rho = 7$)		
3	[80.0000, 9.7232, 0.4700, 0.0020]	47.98
4	[80.0000, 17.5278, 2.5152, 0.1698, 0.0020]	24.82
5	[80.0000, 24.4083, 5.8389, 0.9654, 0.0851, 0.0020]	13.59
6	[80.0000, 30.1833, 9.7232, 2.5152, 0.4700, 0.0515, 0.0020]	7.05
7	[80.0000, 34.9922, 13.6986, 4.6371, 1.2866, 0.2675, 0.0352, 0.0020]	5.08
8	[80.0000, 39.0167, 17.5278, 7.1005, 2.5152, 0.7434, 0.1698, 0.0261, 0.0020]	3.69
9	[80.0000, 42.4152, 21.1087, 9.7232, 4.0661, 1.5017, 0.4700, 0.1166, 0.0204, 0.0020]	3.17
10	[80.0000, 45.3137, 24.408 312.3816, 5.8389, 2.5152, 0.9654, 0.3183, 0.0851, 0.0167, 0.0020]	2.77
GITS (ours)		
3	[80.0000, 3.8811, 0.9654, 0.0020]	43.89
4	[80.0000, 5.8389, 1.8543, 0.4700, 0.0020]	15.10
5	[80.0000, 6.6563, 2.1632, 0.8119, 0.2107, 0.0020]	8.38
6	[80.0000, 10.9836, 3.8811, 1.5840, 0.5666, 0.1698, 0.0020]	4.88
7	[80.0000, 12.3816, 3.8811, 1.5840, 0.5666, 0.1698, 0.0395, 0.0020]	3.76
8	[80.0000, 10.9836, 3.8811, 1.8543, 0.9654, 0.4700, 0.2107, 0.0665, 0.0020]	3.24
9	[80.0000, 12.3816, 4.4590, 2.1632, 1.1431, 0.5666, 0.2597, 0.1079, 0.0300, 0.0020]	2.70
10	[80.0000, 12.3816, 4.4590, 2.1632, 1.1431, 0.5666, 0.3183, 0.1698, 0.0665, 0.0225, 0.0020]	2.49

Table 10. Comparison of FID results. Each column represents the dataset used for searching the optimized time schedule. The optimized time schedules may vary during different experiments due to the randomly sampled batch for optimization.

TIME SCHEDULE	CIFAR-10	FFHQ	ImageNet
NFE = 5			
CIFAR10	8.78	8.88	8.21
FFHQ	11.22	11.12	10.61
ImageNet	11.40	11.44	11.02
NFE = 6			
CIFAR10	5.07	4.89	4.73
FFHQ	7.28	7.00	6.90
ImageNet	8.85	8.61	8.43
NFE = 8			
CIFAR10	3.20	3.32	3.39
FFHQ	4.88	4.52	4.59
ImageNet	6.02	5.79	5.94
NFE = 10			
CIFAR10	2.43	2.40	2.61
FFHQ	3.77	3.62	3.59
ImageNet	4.57	4.36	4.70

References

Achilli B, Ambrogioni L, Lucibello C, Mézard M and Ventura E 2025 Memorization and generalization in generative diffusion under the manifold hypothesis *J. Stat. Mech.* 073401

Alain G and Bengio Y 2014 What regularized auto-encoders learn from the data-generating distribution *J. Mach.: Learn. Res.* **15** 3563–93

Albergo M S, Boffi N M and Vanden-Eijnden E 2023 Stochastic interpolants: a unifying framework for flows and diffusions (arXiv:2303.08797)

Anderson B D O 1982 Reverse-time diffusion equation models *Stoch. Process. Appl.* **12** 313–26

Antognini J and Sohl-Dickstein J 2018 Pca of high dimensional random walks with comparison to neural network training *Advances in Neural Information Processing Systems* vol 31

Bahri Y, Kadmon J, Pennington J, Schoenholz S S, Sohl-Dickstein J and Ganguli S 2020 Statistical mechanics of deep learning *Annu. Rev. Condens. Matter Phys.* **11** 501–28

Balaji Y *et al* 2022 ediffi: text-to-image diffusion models with an ensemble of expert denoisers (arXiv:2211.01324)

Bao F, Chongxuan Li, Zhu J and Zhang B 2022 Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models *Int. Conf. on Learning Representations*

Bengio Y, Courville A and Vincent P 2013a Representation learning: a review and new perspectives *IEEE Trans. Pattern Anal. Mach. Intell.* **35** 1798–828

Bengio Y, Yao Li, Alain G and Vincent P 2013b Generalized denoising auto-encoders as generative models *Advances in Neural Information Processing Systems* pp 899–907

Biroli G, Bonnaire T, De Bortoli V and Mézard M 2024 Dynamical regimes of diffusion models *Nat. Commun.* **15** 9957

Biroli G and Mézard M 2023 Generative diffusion in very large dimensions *J. Stat. Mech.* 093402

Biroli G and Mézard M 2024 Kernel density estimators in large dimensions (arXiv:2408.05807)

Blattmann A, Rombach R, Ling H, Dockhorn T, Kim S W, Fidler S and Kreis K 2023 Align your latents: high-resolution video synthesis with latent diffusion models *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 22563–75

Carmo M P D 2016 *Differential Geometry of Curves and Surfaces: Revised and Updated 2nd edn* (Courier Dover Publications)

Carreira-Perpinán M A 2015 A review of mean-shift algorithms for clustering (arXiv:1503.00687)

Castillo A, Kohler J, Pérez J C, Pérez J P, Pumarola A, Ghanem B, Arbeláez P and Thabet A 2025 Adaptive guidance: training-free acceleration of conditional diffusion models *Proc. AAAI Conf. on Artificial Intelligence* vol 39 pp 1962–70

Chen D, Zhou Z, Mei J-P, Shen C, Chen C and Wang C 2023a A geometric perspective on diffusion models (arXiv:2305.19947)

Chen D, Zhou Z, Wang C, Shen C and Lyu S 2024 On the trajectory regularity of ode-based diffusion sampling *Int. Conf. on Machine Learning* pp 7905–34

Chen M, Huang K, Zhao T and Wang M 2023b Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data *Int. Conf. on Machine Learning* pp 4672–712

Chen S, Daras G and Dimakis A 2023c Restoration-degradation beyond linear diffusions: a non-asymptotic analysis for ddim-type samplers *Int. Conf. on Machine Learning* pp 4462–84

Chen T 2023 On the importance of noise scheduling for diffusion models (arXiv:2301.10972)

Chen T, Liu G-H and Theodorou E 2022 Likelihood training of schrödinger bridge using forward-backward sdes theory *Int. Conf. on Learning Representations*

Cheng Y 1995 Mean shift, mode seeking and clustering *IEEE Trans. Pattern Anal. Mach. Intell.* **17** 790–9

Comaniciu D and Meer P 1999 Mean shift analysis and applications *Proc. Int. Conf. on Computer Vision* pp 1197–203

Comaniciu D and Meer P 2002 Mean shift: a robust approach toward feature space analysis *IEEE Trans. Pattern Anal. Mach. Intell.* **24** 603–19

Comaniciu D, Ramesh V and Meer P 2000 Real-time tracking of non-rigid objects using mean shift *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* pp 142–9

Comaniciu D, Ramesh V and Meer P 2003 Kernel-based object tracking *IEEE Trans. Pattern Anal. Mach. Intell.* **25** 564–77

Cormen T H, Leiserson C E, Rivest R L and Stein C 2022 *Introduction to Algorithms* (MIT Press)

De Bortoli V 2022 Convergence of denoising diffusion models under the manifold hypothesis *Trans. Mach. Learn. Res.* (available at: https://openreview.net/forum?id=MhK5aXo3gB) (arXiv:2208.05314)

Dhariwal P and Nichol A 2021 Diffusion models beat gans on image synthesis *Advances in Neural Information Processing Systems* pp 8780–94

Dockhorn T, Vahdat A and Kreis K 2022 Genie: Higher-order denoising diffusion solvers *Advances in Neural Information Processing Systems* pp 30150–66

Efron B 2010 *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction* (Cambridge University Press)

Esser P *et al* 2024 Scaling rectified flow transformers for high-resolution image synthesis *Int. Conf. on Machine Learning*

Feller W 1949 On the theory of stochastic processes, with particular reference to applications *Proc. 1st Berkeley Symp. on Mathematical Statistics and Probability* pp 403–32

Frankel E, Chen S, Li J, Koh P W, Ratliff L J and Oh S 2025 S4s: solving for a diffusion model solver *Int. Conf. on Machine Learning*

Fukunaga K and Hostetler L 1975 The estimation of the gradient of a density function, with applications in pattern recognition *IEEE Trans. Inf. Theory* **21** 32–40

Ghio D, Dandi Y, Krzakala F and Zdeborová L 2024 Sampling with flows, diffusion and autoregressive neural networks from a spin-glass perspective *Proc. Natl Acad. Sci.* **121** e2311810121

Golub G H and Van Loan C F 2013 *Matrix Computations* vol 3 (The Johns Hopkins University Press)

Gu X, Du C, Pang T, Li C, Lin M and Wang Y 2023 On memorization in diffusion models (arXiv:2310.02664)

Hang T, Gu S, Geng X and Guo B 2024 Improved noise schedule for diffusion training (arXiv:2407.03297)

Heitz E, Belcour L and Chambon T 2023 Iterative α-(de) blending: a minimalist deterministic diffusion model *ACM SIGGRAPH 2023 Conf. Proc.* pp 1–8

Heusel M, Ramsauer H, Unterthiner T, Nessler B and Hochreiter S 2017 GANs trained by a two time-scale update rule converge to a local Nash equilibrium *Advances in Neural Information Processing Systems* pp 6626–37

Ho J, Jain A and Abbeel P 2020 Denoising diffusion probabilistic models *Advances in Neural Information Processing Systems* pp 6840–51

Ho J and Salimans T 2022 Classifier-free diffusion guidance (arXiv:2207.12598)

Ho J, Salimans T, Gritsenko A A, Chan W, Norouzi M and Fleet D J 2022 Video diffusion models *Advances in Neural Information Processing Systems* pp 8633–46

Huang R, Huang J, Yang D, Ren Y, Liu L, Li M, Ye Z, Liu J, Yin X and Zhao Z 2023 Make-an-audio: text-to-audio generation with prompt-enhanced diffusion models *Int. Conf. on Machine Learning* (PMLR) pp 13916–32

Hurley J R and Cattell R B 1962 The Procrustes program: producing direct rotation to test a hypothesized factor structure *Behav. Sci.* **7** 258

Hyvärinen A 2005 Estimation of non-normalized statistical models by score matching *J. Mach.: Learn. Res.* **6** 695–709

Ikeda K, Uda T, Okanohara D and Ito S 2025 Speed-accuracy relations for diffusion models: wisdom from nonequilibrium thermodynamics and optimal transport *Phys. Rev. X* **15** 031031

Jarzynski C 1997 Equilibrium free-energy differences from nonequilibrium measurements: a master-equation approach *Phys. Rev. E* **56** 5018

Kadkhodaie Z, Guth F, Simoncelli E P and Mallat S 2023 Generalization in diffusion models arises from geometry-adaptive harmonic representations (arXiv:2310.02557)

Kamb M and Ganguli S 2024 An analytic theory of creativity in convolutional diffusion models (arXiv:2412.20292)

Karras T, Aittala M, Aila T and Laine S 2022 Elucidating the design space of diffusion-based generative models *Advances in Neural Information Processing Systems* pp 26565–77

Karras T, Aittala M, Kynkäänniemi T, Lehtinen J, Aila T and Laine S 2024 Guiding a diffusion model with a bad version of itself *Advances in Neural Information Processing Systems* pp 52996–3021

Karras T, Laine S and Aila T 2019 A style-based generator architecture for generative adversarial networks *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 4401–10

Kim D, Lai C-H, Liao W-H, Murata N, Takida Y, Uesaka T, Yutong H, Mitsufuji Y and Ermon S 2024 Consistency trajectory models: learning probability flow ode trajectory of diffusion *Int. Conf. on Learning Representations*

Kim D, Shin S, Song K, Kang W and Moon I-C 2022 Soft truncation: a universal training technique of score-based diffusion model for high precision score estimation *Int. Conf. on Machine Learning* pp 11201–28

Kingma D P, Salimans T, Poole B and Ho J 2021 Variational diffusion models *Advances in Neural Information Processing Systems* pp 21696–707

Kirstain Y, Polyak A, Singer U, Matiana S, Penna J and Levy O 2023 Pick-a-pic: an open dataset of user preferences for text-to-image generation *Advances in Neural Information Processing Systems* vol 36 pp 36652–63

Kong Z, Ping W, Huang J, Zhao K and Catanzaro B 2021 Diffwave: a versatile diffusion model for audio synthesis *Int. Conf. on Learning Representations*

Krizhevsky A and Hinton G 2009 Learning multiple layers of features from tiny images *Technical Report*

Kwon M, Jeong J and Uh Y 2023 Diffusion models already have a semantic latent space *Int. Conf. on Learning Representations*

Kynkäänniemi T, Aittala M, Karras T, Laine S, Aila T and Lehtinen J 2024 Applying guidance in a limited interval improves sample and distribution quality in diffusion models *Advances in Neural Information Processing Systems* pp 122458–83

Land A H and Doig A G 1960 An automatic method of solving discrete programming problems *Econometrica* **28** 497–520

Lee H, Jianfeng L and Tan Y 2023 Convergence of score-based generative modeling for general data distributions *Int. Conf. on Algorithmic Learning Theory* pp 946–85

Lewiner T, Gomes J D, Lopes H and Craizer M 2005 Curvature and torsion estimators based on parametric curve fitting *Comput. Graph.* **29** 641–55

Li X, Dai Y and Qu Q 2024 Understanding generalizability of diffusion models requires rethinking the hidden Gaussian structure (arXiv:2410.24060)

Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P and Zitnick C L 2014 Microsoft coco: Common objects in context *Proc. European Conf. on Computer Vision* pp 740–55

Lipman Y, Chen R T Q, Ben-Hamu H, Nickel M and Le M 2023 Flow matching for generative modeling *Int. Conf. on Learning Representations*

Liu G-H, Vahdat A, Huang D-A, Theodorou E A, Nie W and Anandkumar A 2023a I2sb: image-to-image schrödinger bridge *Int. Conf. on Machine Learning* pp 22042–62

Liu L, Ren Y, Lin Z and Zhao Z 2022 Pseudo numerical methods for diffusion models on manifolds *Int. Conf. on Learning Representations*

Liu X, Gong C and Liu Q 2023b Flow straight and fast: learning to generate and transfer data with rectified flow *Int. Conf. on Learning Representations*

Lu C, Zhou Y, Bao F, Chen J, Li C and Zhu J 2022a Dpm-solver: a fast ode solver for diffusion probabilistic model sampling in around 10 steps *Advances in Neural Information Processing Systems* pp 5775–87

Lu C, Zhou Y, Bao F, Chen J, Li C and Zhu J 2022b Dpm-solver++: fast solver for guided sampling of diffusion probabilistic models (arXiv:2211.01095)

Luhman E and Luhman T 2021 Knowledge distillation in iterative generative models for improved sampling speed (arXiv:2101.02388)

Lyu S 2009 Interpretation and generalization of score matching *Proc. 25th Conf. on Uncertainty in Artificial Intelligence* pp 359–66

Moore J, Ahmed H and Antia R 2018 High dimensional random walks can appear low dimensional: application to influenza H_3N_2 evolution *J. Theor. Biol.* **447** 56–64

Morris C N 1983 Parametric empirical Bayes inference: theory and applications *J. Am. Stat. Assoc.* **78** 47–55

Neklyudov K, Brekelmans R, Severo D and Makhzani A 2023 Action matching: learning stochastic dynamics from samples *Int. Conf. on Machine Learning* (PMLR) pp 25858–89

Nichol A Q and Dhariwal P 2021 Improved denoising diffusion probabilistic models *Int. Conf. on Machine Learning* pp 8162–71

Nichol A Q, Dhariwal P, Ramesh A, Shyam P, Mishkin P, Mcgrew B, Sutskever I and Chen M 2022 Glide: towards photorealistic image generation and editing with text-guided diffusion models *Int. Conf. on Machine Learning* pp 16784–804

Niedoba M, Zwartsenberg B, Murphy K and Wood F 2024 Towards a mechanistic explanation of diffusion model generalization (arXiv:2411.19339)

Oksendal B 2013 *Stochastic Differential Equations: an Introduction With Applications* (Springer Science & Business Media)

Peebles W and Xie S 2023 Scalable diffusion models with transformers *Proc. IEEE/CVF Int. Conf. on Computer Vision* pp 4195–205

Pidstrigach J 2022 Score-based generative models detect manifolds *Advances in Neural Information Processing Systems* pp 35852–65

Podell D, English Z, Lacey K, Blattmann A, Dockhorn T, Müller J, Penna J and Rombach R 2024 Sdxl: Improving latent diffusion models for high-resolution image synthesis *Int. Conf. on Learning Representations*

Ramesh A, Dhariwal P, Nichol A, Chu C and Chen M 2022 Hierarchical text-conditional image generation with clip latents (arXiv:2204.06125)

Raphan M and Simoncelli E P 2011 Least squares estimation without priors or supervision *Neural Comput.* **23** 374–420

Raya G and Ambrogioni L 2024 Spontaneous symmetry breaking in generative diffusion models *J. Stat. Mech.* 104025

Robbins H E 1956 An empirical Bayes approach to statistics *Proc. 3rd Berkeley Symp. on Mathematical Statistics and Probability*

Rombach R, Blattmann A, Lorenz D, Esser P and Ommer B 2022 High-resolution image synthesis with latent diffusion models *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 10684–95

Ruiz N, Li Y, Jampani V, Pritch Y, Rubinstein M and Aberman K 2023 Dreambooth: fine tuning text-to-image diffusion models for subject-driven generation *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 22500–10

Russakovsky O *et al* 2015 Imagenet large scale visual recognition challenge *Int. J. Comput. Vis.* **115** 211–52

Sabour A, Fidler S and Kreis K 2024 Align your steps: optimizing sampling schedules in diffusion models *Int. Conf. on Machine Learning*

Saharia C *et al* 2022 Photorealistic text-to-image diffusion models with deep language understanding *Advances in Neural Information Processing Systems* pp 36479–94

Salimans T and Jonathan H 2022 Progressive distillation for fast sampling of diffusion models *Int. Conf. on Learning Representations*

Särkkä S and Solin A 2019 *Applied Stochastic Differential Equations* vol 10 (Cambridge University Press)

Scarvelis C, de Ocáriz Borde H S and Solomon J 2023 Closed-form diffusion models (arXiv:2310.12395)

Schönemann P H 1966 A generalized solution of the orthogonal procrustes problem *Psychometrika* **31** 1–10

Shen C, Brooks M J and van den Hengel A 2005 Fast global kernel density mode seeking with application to localisation and tracking *Proc. Int. Conf. on Computer Vision* pp 1516–23

Silverman B W 1981 Using kernel density estimates to investigate multimodality *J. R. Stat. Soc. B* **43** 97–99

Sohl-Dickstein J, Weiss E, Maheswaranathan N and Ganguli S 2015 Deep unsupervised learning using nonequilibrium thermodynamics *Int. Conf. on Machine Learning* pp 2256–65

Song J, Meng C and Ermon S 2021a Denoising diffusion implicit models *Int. Conf. on Learning Representations*

Song Y, Dhariwal P, Chen M and Sutskever I 2023 Consistency models *Int. Conf. on Machine Learning* pp 32211–52

Song Y, Durkan C, Murray I and Ermon S 2021b Maximum likelihood training of score-based diffusion models *Advances in Neural Information Processing Systems* pp 1415–28

Song Y and Ermon S 2019 Generative modeling by estimating gradients of the data distribution *Advances in Neural Information Processing Systems* pp 11895–907

Song Y and Ermon S 2020 Improved techniques for training score-based generative models *Advances in Neural Information Processing Systems* pp 12438–48

Song Y, Garg S, Shi J and Ermon S 2019 Sliced score matching: a scalable approach to density and score estimation *Uncertainty in Artificial Intelligence* (PMLR) pp 574–84

Song Y, Sohl-Dickstein J, Kingma D P, Kumar A, Ermon S and Poole B 2021c Score-based generative modeling through stochastic differential equations *Int. Conf. on Learning Representations*

Stratonovich R 1968 *Conditional Markov Processes and Their Application to the Theory of Optimal Control* (American Elsevier Publishing)

Sutton R S *et al* 1998 *Reinforcement Learning: An Introduction* vol 1 (MIT Press Cambridge)

Tong V, Hoang T-D, Liu A, Van den Broeck G and Niepert M 2025 Learning to discretize denoising diffusion odes *Int. Conf. on Learning Representations*

Vahdat A, Kreis K and Kautz J 2021 Score-based generative modeling in latent space *Advances in Neural Information Processing Systems* pp 11287–302

Vershynin R 2018 *High-Dimensional Probability: An Introduction With Applications in Data Science* vol 47 (Cambridge University Press)

Vincent P 2011 A connection between score matching and denoising autoencoders *Neural Comput.* **23** 1661–74

Vincent P, Larochelle H, Bengio Y and Manzagol P-A 2008 Extracting and composing robust features with denoising autoencoders *Int. Conf. on Machine Learning* pp 1096–103

Wang B and Vastola J 2024 The unreasonable effectiveness of gaussian score approximation for diffusion models and its applications *Trans. Mach. Learn. Res.* (available at: https://openreview.net/forum?id=I0uknSHM2j) (arXiv:2412.09726)

Watson D, Ho J, Norouzi M and Chan W 2021 Learning to efficiently sample from diffusion probabilistic models (arXiv:2106.03802)

Whitney H 1936 Differentiable manifolds *Ann. Math.* **37** 645–80

Xie E *et al* 2025 Sana: efficient high-resolution image synthesis with linear diffusion transformers *Int. Conf. on Learning Representations*

Xu Y, Tong S and Jaakkola T S 2023 Stable target field for reduced variance score estimation in diffusion models *Int. Conf. on Learning Representations*

Yamasaki R and Tanaka T 2020 Properties of mean shift *IEEE Trans. Pattern Anal. Mach. Intell.* **42** 2273–86

Yi M, Sun J and Li Z 2023 On the generalization of diffusion model (arXiv:2305.14712)

Young H D, Freedman R A, Sandin T R and Ford A L 1996 *University Physics* vol 9 (Addison-wesley Reading)

Yu F, Seff A, Zhang Y, Song S, Funkhouser T and Xiao J 2015 Lsun: construction of a large-scale image dataset using deep learning with humans in the loop (arXiv:1506.03365)

Yu Z and Huang H 2025 Nonequilbrium physics of generative diffusion models *Phys. Rev. E* **111** 014111

Zhang Q and Chen Y 2021 Diffusion normalizing flow *Advances in Neural Information Processing Systems* pp 16280–91

Zhang Q and Chen Y 2023 Fast sampling of diffusion models with exponential integrator *Int. Conf. on Learning Representations*

Zhang Q, Song J and Chen Y 2023 Improved order analysis and design of exponential integrator for diffusion models sampling (arXiv:2308.02157)

Zhao W, Bai L, Rao Y, Zhou J and Lu J 2023 Unipc: a unified predictor-corrector framework for fast sampling of diffusion models *Advances in Neural Information Processing Systems* pp 49842–69

Zheng H, Nie W, Vahdat A, Azizzadenesheli K and Anandkumar A 2023 Fast sampling of diffusion models via operator learning *Int. Conf. on Machine Learning* pp 42390–402

Zhou Z, Chen D, Wang C and Chen C 2024a Fast ode-based sampling for diffusion models in around 5 steps *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 7777–86

Zhou Z, Chen D, Wang C, Chen C and Lyu S 2024b Simple and fast distillation of diffusion models *Advances in Neural Information Processing Systems* vol 37 pp 40831–60