Accelerating Convergence of Score-Based Diffusion Models, Provably

Gen Li [*] [1] Yu Huang [*] [2] Timofey Efimov [3] Yuting Wei [2] Yuejie Chi [3] Yuxin Chen [2]

Abstract

Score-based diffusion models, while achieving remarkable empirical performance, often suffer from low sampling speed, due to extensive function evaluations needed during the sampling phase. Despite a flurry of recent activities towards speeding up diffusion generative modeling in practice, theoretical underpinnings for acceleration techniques remain severely limited. In this paper, we design novel training-free algorithms to accelerate popular deterministic (i.e., DDIM) and stochastic (i.e., DDPM) samplers. Our accelerated deterministic sampler converges at a rate $O(\frac{1}{T^2})$ with T the number of steps, improving upon the $O(\frac{1}{T})$ rate for the DDIM sampler; and our accelerated stochastic sampler converges at a rate $O(\frac{1}{T})$, outperforming the rate $O(\frac{1}{\sqrt{T}})$ for the DDPM sampler. The design of our algorithms leverages insights from higher-order approximation, and shares similar intuitions as popular high-order ODE solvers like the DPM-Solver-2. Our theory accommodates ℓ_2-accurate score estimates, and does not require log-concavity or smoothness on the target distribution.

1. Introduction

Initially introduced by Sohl-Dickstein et al. (2015) and subsequently gaining momentum through the works Ho et al. (2020); Song et al. (2021), diffusion models have risen to the forefront of generative modeling. Remarkably, score-based diffusion models have demonstrated superior performance across various domains like computer vision, natural language processing, medical imaging, and bioinformatics (Croitoru et al., 2023; Yang et al., 2023; Kazerouni et al., 2023; Guo et al., 2023), outperforming earlier generative methods such as GANs (Goodfellow et al., 2020) and VAEs (Kingma and Welling, 2014) on multiple fronts (Dhariwal and Nichol, 2021).

1.1. Score-based diffusion models

On a high level, diffusion-based generative modeling begins by considering a forward Markov diffusion process that progressively diffuses a data distribution into noise:

$$X_0 \overset{\text{add noise}}{\longrightarrow} X_1 \overset{\text{add noise}}{\longrightarrow} X_2 \overset{\text{add noise}}{\longrightarrow} \cdots \overset{\text{add noise}}{\longrightarrow} X_T, \qquad (1)$$

where $X_0 \sim p_{\mathsf{data}}$ is drawn from the target data distribution in $\mathbb{R}^d$, and X_T resembles pure noise (e.g., with a distribution close to $\mathcal{N}(0, I_d)$). The pivotal step then lies in learning to construct a reverse Markov process

$$Y_0 \overset{\text{use scores}}{\longleftarrow} Y_1 \overset{\text{use scores}}{\longleftarrow} Y_2 \overset{\text{use scores}}{\longleftarrow} \cdots \overset{\text{use scores}}{\longleftarrow} Y_T, \qquad (2)$$

which starts from purse noise $Y_T \sim \mathcal{N}(0, I_d)$ and maintains distributional proximity throughout in the sense that $Y_t \overset{\mathrm{d}}{\approx} X_t$ $(t \leq T)$. To accomplish this goal, Y_{t-1} in each step is typically obtained from Y_t with the aid of (Stein) score functions — namely, $\nabla_X \log p_{X_t}(X)$, with p_{X_t} denoting the distribution of X_t — where the score functions are pre-trained by means of score matching techniques (e.g., Hyvärinen (2005); Ho et al. (2020); Hyvärinen (2007); Vincent (2011); Song and Ermon (2019); Pang et al. (2020)).

The mainstream approaches for constructing the reverse-time process (2) can roughly be divided into two categories, as described below.

- *Stochastic (or SDE-based) samplers.* A widely adopted strategy involves exploiting both the score function and some injected random noise when generating each Y_{t-1}; that is, Y_{t-1} is taken to be a function of Y_t and some independent noise Z_t. A prominent example of this kind is the Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020), to be detailed in Section 2. Notably, this approach has intimate connections with certain stochastic differential equations (SDEs), which can be elucidated via celebrated SDE results concerning the existence of reverse-time diffusion processes (Anderson, 1982; Haussmann and Pardoux, 1986).

[*] Equal contribution [1] Department of Statistics, The Chinese University of Hong Kong, Hong Kong [2] Department of Statistics and Data Science, Wharton School, University of Pennsylvania, Philadelphia, PA 19104, USA [3] Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Correspondence to: Yuxin Chen <yuxinc@wharton.upenn.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

- *Deterministic (or ODE-based) samplers.* In contrast, another approach is purely deterministic (except for the generation of Y_T), constructing Y_{t-1} as a function of the previously computed steps (e.g., Y_t) without injecting any additional noise. This approach was introduced by Song et al. (2021), as inspired by the existence of ordinary differential equations (ODEs) — termed *probability flow ODEs* or *diffusion ODEs* — exhibiting the same marginal distributions as the above-mentioned reverse-time diffusion process. A notable example in this category is often referred to as the Denoising Diffusion Implicit Model (DDIM) (Song et al., 2020).

In practice, it is often observed that DDIM converges more rapidly than DDPM, although the final data instances produced by DDPM (given sufficient runtime) might enjoy better diversity compared to the output of DDIM.

1.2. Non-asymptotic convergence theory and acceleration

Despite the astounding empirical success, theoretical analysis for diffusion-based generative modeling is still in its early stages of development. Treating the score matching step as a blackbox and exploiting only (crude) information about the score estimation error, a recent strand of works have explored the convergence rates of the data generating process (i.e., the reverse Markov process) in a non-asymptotic fashion, in an attempt to uncover how fast sampling can be performed (e.g., Lee et al. (2022; 2023); Chen et al. (2022; 2023a;c;b); Li et al. (2023); Benton et al. (2023b;a); Liang et al. (2024)). In what follows, let us give a brief overview of the state-of-the-art results in this direction. Here and throughout, the iteration complexity of a sampler refers to the number of steps T needed to attain ε accuracy in the sense that $\mathsf{TV}(p_{X_1}, p_{Y_1}) \leq \varepsilon$, where $\mathsf{TV}(\cdot,\cdot)$ represents the total-variation (TV) distance between two distributions, and p_{X_1} (resp. p_{Y_1}) stands for the distribution of X_1 (resp. Y_1).

- *Convergence rate of stochastic samplers.* Assuming Lipschitz continuity (or smoothness) of the score functions across all steps, Chen et al. (2022) proved that the iteration complexity of the DDPM sampler is proportional to $1/\varepsilon^2$. The Lipschitz assumption is then relaxed by Chen et al. (2023a); Benton et al. (2023a); Li et al. (2023), revealing that the scaling $1/\varepsilon^2$ is achievable for a fairly general family of data distributions.

- *Convergence rate of deterministic samplers.* As alluded to previously, deterministic samplers often exhibit faster convergence in both practice and theory. For instance, Chen et al. (2023c) provided the first polynomial convergence guarantees for the probability flow ODE sampler under exact scores, whereas Li et al. (2023) demonstrated that its iteration complexity scales proportionally to $1/\varepsilon$. Note that the theory in Li et al. (2023) also accommodates score estimation errors. Additionally, it is noteworthy that an iteration complexity proportional to $1/\varepsilon$ has also been established by Chen et al. (2023b) for a variant of the probability flow ODE sampler, although the sampler studied therein incorporates a stochastic corrector step in each iteration.

Acceleration? While the theoretical studies outlined above have offered non-asymptotic convergence guarantees for both the stochastic and deterministic samplers, one might naturally wonder whether there is potential for achieving faster rates. In practice, the evaluation of Stein scores in each step often entails computing the output of a large neural network, thereby calling for new solutions to reduce the number of score evaluations without compromising sampling fidelity. Indeed, this has inspired a large strand of recent works focused on speeding up diffusion generative modeling. Towards this end, one prominent approach is *distillation*, which attempts to distill a pre-trained diffusion model into another model (e.g., progressive distillation, consistency model) that can be executed in significantly fewer steps (Luhman and Luhman, 2021; Salimans and Ho, 2021; Meng et al., 2023; Song et al., 2023). However, while distillation-based techniques have achieved outstanding empirical performance, they often necessitate additional training processes, imposing high computational burdens beyond score matching. In contrast, an alternative route towards acceleration is "training-free," which directly invokes the pre-trained diffusion model (particularly the pre-trained score functions) for sampling without requiring additional training processes. Examples of training-free accelerated samplers include the DPM-Solver (Lu et al., 2022a), the DPM-Solver++ (Lu et al., 2022b), DEIS (Zhang and Chen, 2022), UniPC (Zhao et al., 2023), the SA-Solver (Xue et al., 2023), among others, which leverage faster solvers for ODE and SDE using only the pre-trained score functions. Nevertheless, non-asymptotic convergence analyses for these methods remain largely absent, making it challenging to rigorize the degrees of acceleration compared to the non-accelerated results (Lee et al., 2023; Chen et al., 2022; 2023a; Li et al., 2023; Benton et al., 2023a). All of this leads to the following question that we aim to explore in this work:

> *Can we design a training-free deterministic (resp. stochastic) sampler that converges provably faster than the DDIM (resp. DDPM)?*

1.3. Our contributions

In this paper, we answer the above question in the affirmative. Our main contributions can be summarized as follows.

- In the deterministic setting, we demonstrate how to speed up the ODE-based sampler (i.e., the DDIM-type sampler). The proposed sampler, which exploits some sort of momentum term to adjust the update rule, leverages insights from higher-order ODE approximation in discrete time and shares similar intuitions with the fast ODE-based sampler DPM-Solver-2 (Lu et al., 2022a). We establish non-asymptotic convergence guarantees for the accelerated DDIM-type sampler, showing that its iteration complexity scales proportionally to $1/\sqrt{\varepsilon}$ (up to log factor). This substantially improves upon the prior convergence theory for the original DDIM sampler (Li et al., 2023) (which has an iteration complexity proportional to $1/\varepsilon$).

- In the stochastic setting, we propose a novel sampling procedure to accelerate the SDE-based sampler (i.e., the DDPM-type sampler). For this new sampler, we establish an iteration complexity bound proportional to $1/\varepsilon$ (modulo some log factor), thus unveiling the superiority of the proposed sampler compared to the original DDPM sampler (recall that the original DDPM sampler has an iteration complexity proportional to $1/\varepsilon^2$ (Li et al., 2023; Chen et al., 2023a; 2022)).

In addition, two aspects of our theory are worth emphasizing: (i) our theory accommodates ℓ_2-accurate score estimates, rather than requiring ℓ_∞ score estimation accuracy; (ii) our theory covers a fairly general family of target data distributions, without imposing stringent assumptions like log-concavity and smoothness on the target distributions.

1.4. Other related works

We now briefly discuss additional related works in the prior art.

Convergence of score-based generative models (SGMs). For stochastic samplers of SGMs, the convergence guarantees were initially provided by early works including but not limited to De Bortoli et al. (2021); Liu et al. (2022b); Pidstrigach (2022); Block et al. (2020); De Bortoli (2022); Wibisono and Yang (2022); Gao et al. (2023), which often faced issues of either being not quantitative or suffering from the curse of dimensionality. More recent research has advanced this field by relaxing the assumptions on the score function and achieving polynomial convergence rates (Lee et al., 2022; 2023; Chen et al., 2022; 2023a;b; Li et al., 2023; Benton et al., 2023a; Liang et al., 2024; Tang and Zhao, 2024b). Furthermore, theoretical insights into probability flow-based ODE samplers, though less abundant, have been explored in recent works (Chen et al., 2023c; Li et al., 2023; Chen et al., 2023b; Benton et al., 2023b; Gao and Zhu, 2024). Additionally, Tang and Zhao (2024a) provided a continuous-time sampling error guarantee for a novel class of contraction diffusion models. Gao and Zhu (2024) studies the convergence properties for general probability flow ODEs w.r.t. Wasserstein distances. Most recently, Chen and Ying (2024) makes a step towards the convergence analysis of discrete state space diffusion model. Note that this body of research primarily aims to quantify the proximity between distributions generated by SGMs and the ground truth distributions, assuming availability of an accurate score estimation oracle. Interestingly, a very recent research by Li et al. (2024c) reveals that even SGMs with empirically optimized score functions might underperform due to strong memorization effects. Moreover, some works delve into other aspects of the theoretical understanding of diffusion models. Furthermore, Wu et al. (2024) investigated how diffusion guidance combined with DDPM and DDIM samplers influences the conditional sampling quality.

Fast sampling in diffusion models. A recent strand of works to achieve few-step sampling — or even one-step sampling — falls under the category of training-based samplers, primarily focused on knowledge distillation (Meng et al., 2023; Salimans and Ho, 2021; Song et al., 2023). This method aims to distill a pre-trained diffusion model into another model that can be executed in significantly fewer steps. The recent work (Li et al., 2024b) provided a first attempt towards theoretically understanding the sampling efficiency of consistency models. Another line of works aims to design training-free samplers (Lu et al., 2022a;b; Zhao et al., 2023; Zhang and Chen, 2022; Liu et al., 2022a; Zhang et al., 2022), which addresses the efficiency issue by developing faster solvers for the reverse-time SDE or ODE without requiring other information beyond the pre-trained SGMs. In addition, Li et al. (2023); Liang et al. (2024) introduced accelerated samplers that require additional training pertaining to estimating Hessian information at each step. Furthermore, combining GAN with diffusion has shown to be an effective strategy to speed up the sampling process (Wang et al., 2022; Xiao et al., 2021).

1.5. Notation

Before continuing, we find it helpful to introduce some notational conventions to be used throughout this paper. Capital letters are often used to represent random variables/vectors/processes, while lowercase letters denote deterministic variables. When considering two probability measures P and Q, we define their total-variation (TV) distance as $\mathsf{TV}(P,Q) := \frac{1}{2}\int |\mathrm{d}P - \mathrm{d}Q|$, and the Kullback-Leibler (KL) divergence as $\mathsf{KL}(P\,\|\,Q) := \int \big(\log\frac{\mathrm{d}P}{\mathrm{d}Q}\big)\mathrm{d}P$. We use $p_X(\cdot)$ and $p_{X\,|\,Y}(\cdot\,|\,\cdot)$ to denote the probability density function of a random vector X, and the conditional probability of X given Y, respectively. For matrices, $\|A\|$ and $\|A\|_{\mathrm{F}}$ refer to the spectral norm and Frobenius norm of a matrix A, respectively. For vector-valued functions f, we use J_f or $\frac{\partial f}{\partial x}$ to represent the Jacobian matrix of f. Given two functions $f(d,T)$ and $g(d,T)$, we employ the notation $f(d,T) \lesssim g(d,T)$ or $f(d,T) = O(g(d,T))$

(resp. $f(d,T) \gtrsim g(d,T)$) to indicate the existence of a universal constant $C_1 > 0$ such that for all d and T, $f(d,T) \leq C_1 g(d,T)$ (resp. $f(d,T) \geq C_1 g(d,T)$). The notation $f(d,T) \asymp g(d,T)$ indicates that both $f(d,T) \lesssim g(d,T)$ and $f(d,T) \gtrsim g(d,T)$ hold at once.

2. Problem settings

In this section, we formulate the problem, and introduce a couple of key assumptions.

2.1. Model and sampling process

Forward process. Consider the forward Markov process (1) in discrete time that starts from the target data distribution $X_0 \sim p_{\mathsf{data}}$ in $\mathbb{R}^d$ and proceeds as follows:

$$X_t = \sqrt{1-\beta_t} X_{t-1} + \sqrt{\beta_t}\, W_t, \qquad t = 1, \cdots, T, \quad (3)$$

where the W_t's are independently drawn from $\mathcal{N}(0, I_d)$. This process is said to be "variance-preserving," in the sense that the covariance $\mathsf{Cov}(X_t) = I_d$ holds throughout if $\mathsf{Cov}(X_0) = I_d$. Taking

$$\overline{\alpha}_t \coloneqq \prod_{k=1}^{t} \alpha_k \qquad \text{with } \alpha_t \coloneqq 1 - \beta_t \quad (4)$$

for every $1 \leq t \leq T$, one can write

$$X_t = \sqrt{\overline{\alpha}_t} X_0 + \sqrt{1-\overline{\alpha}_t}\, \overline{W}_t \quad \text{for } \overline{W}_t \sim \mathcal{N}(0, I_d). \quad (5)$$

Throughout the paper, we shall use $q_t(\cdot)$ or $p_{X_t}(\cdot)$ interchangeably to denote the probability density function (PDF) of X_t. While we shall concentrate on the discrete-time process in the current paper, we shall note that the forward process has also been commonly studied in the continuous-time limit through the following diffusion process for $0 \leq t \leq T$

$$\mathrm{d}X_t = -\frac{1}{2}\beta(t) X_t \mathrm{d}t + \sqrt{\beta(t)} \mathrm{d}W_t, \quad X_0 \sim p_{\text{data}} \quad (6)$$

for some function $\beta(t)$ related to the learning rate, where W_t is the standard Brownian motion.

Score functions and score estimates. A key ingredient that plays a pivotal role in the sampling process is the (Stein) score function, defined as the log marginal density of the forward process.

Definition 2.1 (Score function)**.** The score function, denoted by $s_t^\star : \mathbb{R}^d \to \mathbb{R}^d (1 \leq t \leq T)$, is defined as

$$\begin{aligned} s_t^\star(X) &\coloneqq \nabla \log q_t(X) \\ &= -\frac{1}{1-\overline{\alpha}_t} \int_{x_0} p_{X_0 \,|\, X_t}(x_0 \,|\, x)\big(x - \sqrt{\overline{\alpha}_t} x_0\big) \mathrm{d}x_0. \end{aligned} \quad (7)$$

Here, the last identity follows from standard properties about Gaussians; see, e.g., Chen et al. (2022). In most applications, we have no access to perfect score functions; instead, what we have available are certain estimates for the score functions, to be denoted by $\{s_t(\cdot)\}_{1 \leq t \leq T}$ throughout.

Data generation process. The sampling process is performed via careful construction of the reverse process (2) to ensure distributional proximity. Working backward from $t = T, \ldots, 1$, we assume throughout that $Y_T \sim \mathcal{N}(0, I_d)$.

- *Deterministic sampler.* A deterministic sampler typically chooses Y_{t-1} for each t to be a function of $\{Y_t, \ldots, Y_T\}$. For instance, the following construction

$$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(Y_t + \frac{1-\alpha_t}{2} s_t(Y_t) \right), \ t = T, \ldots, 1 \quad (8)$$

  can be viewed as a DDIM-type sampler in discrete time. Note that the DDIM sampler is intimately connected with the following ODE — called the probability flow ODE or the diffusion ODE — in the continuous-time limit:

$$\mathrm{d}\widetilde{Y}_t = -\frac{1}{2}\beta(t) \left(\widetilde{Y}_t + \nabla \log q_t\big(\widetilde{Y}_t\big) \right) \mathrm{d}t, \ \widetilde{Y}_T \sim q_T \quad (9)$$

  which enjoys matching marginal distributions as the forward diffusion process (6) in the sense that $\widetilde{Y}_t \overset{\mathrm{d}}{=} X_t$ for all $0 \leq t \leq T$ (Song et al., 2021).

- *Stochastic sampler.* In contrast to the deterministic case, each Y_{t-1} is a function of not only $\{Y_t, \ldots, Y_T\}$ but also an additional independent noise $Z_t \sim \mathcal{N}(0, I_d)$. One example is the following sampler:

$$Y_{t-1} = \frac{1}{\sqrt{\alpha_t}} \Big(Y_t + (1-\alpha_t) s_t(Y_t) \Big) + \sqrt{1-\alpha_t} Z_t, \quad (10)$$

  which is closely related to the DDPM sampler in discrete time. The design of DDPM draws inspiration from a well-renowned result in the SDE literature (Anderson, 1982; Haussmann and Pardoux, 1986); namely, there exists a reverse-time SDE

$$\mathrm{d}\widehat{Y}_t = -\frac{1}{2}\beta(t) \left(\widehat{Y}_t + 2\nabla \log q_t\big(\widehat{Y}_t\big) \right) \mathrm{d}t + \sqrt{\beta(t)} \mathrm{d}\widehat{Z}_t \quad (11)$$

  with $\widehat{Y}_T \sim q_T$ that exhibits the same marginals — $\widehat{Y}_t \overset{\mathrm{d}}{=} X_t$ for all t — as the forward diffusion process (6). Here, $\widehat{Z}_t$ indicates a backward standard Brownian motion.

2.2. Assumptions

Before moving on to our algorithms and theory, let us introduce several assumptions that shall be used multiple times in this paper. To begin with, we impose the following assumption on the target data distribution.

Assumption 2.2. Suppose that X_0 is a continuous random vector, and obeys

$$\mathbb{P}\big(\|X_0\|_2 \leq R = T^{c_R} \mid X_0 \sim p_{\mathsf{data}}\big) = 1 \tag{12}$$

for some arbitrarily large constant $c_R > 0$.

In words, the size of X_0 is allowed to grow polynomially (with arbitrarily large constant degree) in the number of steps, which suffices to accommodate the vast majority of practical applications.

Next, we specify the learning rates $\{\beta_t\}$ (or $\{\alpha_t\}$) employed in the forward process (3). Throughout this paper, we select the same learning rate schedule as in Li et al. (2023), namely,

$$\beta_1 = 1 - \alpha_1 = \frac{1}{T^{c_0}}, \tag{13a}$$

$$\begin{aligned}\beta_t &= 1 - \alpha_t & (13b)\\ &= \frac{c_1 \log T}{T} \min\left\{\beta_1\left(1 + \frac{c_1 \log T}{T}\right)^t, 1\right\}, \quad t > 1\end{aligned}$$

for some large enough numerical constants $c_0, c_1 > 0$. In short, there are two phases here: at first β_t grows exponentially fast, and then stays unchanged after surpassing some threshold. This also resembles the learning rate choices recommended by Benton et al. (2023a).

Moreover, let us also introduce two assumptions regarding the accuracy of the score estimates $\{s_t\}$, which are adopted in Li et al. (2023). Here and throughout, we denote by

$$J_{s_t^\star} = \frac{\partial s_t^\star}{\partial x} \qquad \text{and} \qquad J_{s_t} = \frac{\partial s_t}{\partial x}, \tag{14}$$

the Jacobian matrices of $s_t^\star(\cdot)$ and $s_t(\cdot)$, respectively.

Assumption 2.3. Suppose that the mean squared estimation error of the score estimates $\{s_t\}_{1\leq t\leq T}$ obeys

$$\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{X\sim q_t}\left[\|s_t(X) - s_t^\star(X)\|_2^2\right] \leq \varepsilon_{\mathsf{score}}^2.$$

Assumption 2.4. Suppose that $s_t(\cdot)$ is continuously differentiable for each $1 \leq t \leq T$, and that the Jacobian matrices associated with the score estimates $\{s_t\}_{1\leq t\leq T}$ satisfy

$$\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{X\sim q_t}\left[\|J_{s_t}(X) - J_{s_t^\star}(X)\|\right] \leq \varepsilon_{\mathsf{Jacobi}}.$$

In short, Assumption 2.3 is concerned with the ℓ_2 score estimation error averaged across all steps, whereas Assumption 2.4 is about the average discrepancy in the associated Jacobian matrices. It is worth noting that Assumption 2.4 will only be imposed when analyzing the convergence of deterministic samplers, and is completely unnecessary for the stochastic counterpart.

3. Algorithm and main theory

In this section, we put forward two accelerated samplers — an ODE-based algorithm and an SDE-based algorithm — and present convergence theory to confirm the acceleration compared with prior DDIM and DDPM approaches. Due to space limitations, all proofs of our main theory are provided in the arXiv version Li et al. (2024a).

3.1. Accelerated ODE-based sampler

The first algorithm we propose is an accelerated variant of the ODE-based deterministic sampler. Specifically, starting from $Y_T \sim \mathcal{N}(0, I_d)$, the proposed discrete-time sampler adopts the following update rule:

$$Y_t^- = \Phi_t(Y_t), \quad Y_{t-1} = \Psi_t(Y_t, Y_t^-) \quad \text{for } t = T, \cdots, 1 \tag{15a}$$

where the mappings $\Phi_t(\cdot)$ and $\Psi_t(\cdot,\cdot)$ are chosen to be

$$\Phi_t(x) = \sqrt{\alpha_{t+1}}\bigg(x - \frac{1-\alpha_{t+1}}{2}s_t(x)\bigg), \tag{15b}$$

$$\begin{aligned}\Psi_t(x,y) = \frac{1}{\sqrt{\alpha_t}}\bigg(x &+ \frac{1-\alpha_t}{2}s_t(x) & (15c)\\ &+ \frac{(1-\alpha_t)^2}{4(1-\alpha_{t+1})}\big(s_t(x) - \sqrt{\alpha_{t+1}}s_{t+1}(y)\big)\bigg),\end{aligned}$$

and we remind the reader that s_t is the score estimate. In contrast to the original DDIM-type solver (8), the proposed accelerated sampler enjoys two distinguishing features:

- In each iteration t, the proposed sampler computes a mid-point $Y_t^- = \Phi_t(Y_t)$ (cf. (15b)). As it turns out, this mid-point is designed as a prediction of the probability flow ODE at time $t+1$ using Y_t.
- In contrast to (8), the proposed update rule $Y_{t-1} = \Psi_t(Y_t, Y_t^-)$ (see (15c)) includes an additional term that is a properly scaled version of $s_t(Y_t) - \sqrt{\alpha_{t+1}}s_{t+1}(Y_t^-)$. In some sense, this term can be roughly viewed as exploiting "momentum" in adjusting the original sampling rule.

Theoretical guarantees. Let us proceed to present our convergence theory and its implications for the proposed deterministic sampler.

Theorem 3.1. *Suppose that Assumptions 2.2, 2.3 and 2.4 hold. Then the proposed sampler (15) with the learning rate schedule (14) satisfies*

$$\begin{aligned}\mathsf{TV}\big(q_1, p_1\big) \leq C_1\frac{d^6\log^6 T}{T^2} &+ C_1\sqrt{d\log^3 T}\varepsilon_{\mathsf{score}}\\ &+ C_1(d\log T)\varepsilon_{\mathsf{Jacobi}} \qquad (16)\end{aligned}$$

for some universal constants $C_1 > 0$, where we recall that p_1 (resp. q_1) denotes the distribution of Y_1 (resp. X_1).

We now take a moment to discuss the implications about this theorem.

- *Iteration complexity.* When the target accuracy level ε is small enough, the number of iterations needed to yield $\mathsf{TV}\big(q_1, p_1\big) \le \varepsilon$ is no larger than

$$\text{(iteration complexity)} \qquad \frac{\mathrm{poly}(d)}{\sqrt{\varepsilon}}, \tag{17}$$

  ignoring any logarithmic factor in $1/\varepsilon$. Clearly, the dependency on $1/\varepsilon$ substantially improves upon the vanilla DDIM sampler, the latter of which has an iteration complexity proportional to $1/\varepsilon$ (Li et al., 2023).

- *Stability vis-a-vis score errors.* The discrepancy between the distribution of Y_1 and the target distribution of X_1 is proportional to the ℓ_2 score estimation error $\varepsilon_{\mathsf{score}}$ defined in Assumption 2.3, as well as the Jacobian error $\varepsilon_{\mathsf{Jacobi}}$ defined in Assumption 2.4. It is worth noting, however, that the same result might not hold if we remove Assumption 2.4. More specifically, when only score estimation accuracy is assumed, the deterministic sampler is not guaranteed to achieve small TV error; see Li et al. (2023) for an illustrative example.

Interpretation via second-order ODE. In order to help elucidate the rationale of the proposed sampler, we make note of an intimate connection between (15) and high-order ODE, the latter of which has facilitated the design of fast deterministic samplers (e.g., DPM-Solver (Lu et al., 2022a)).

In view of the relation (5), for any $0 < \gamma < 1$, let us first abuse the notation and introduce

$$X(\gamma) \overset{\mathrm{d}}{=} \sqrt{\gamma} X_0 + \sqrt{1-\gamma} Z, \quad Z \sim \mathcal{N}(0, I_d) \tag{18a}$$

$$s_\gamma^\star(X) \coloneqq \nabla_X \log p_{X(\gamma)}(X). \tag{18b}$$

We further consider the following continuous-time analog $\overline{\alpha}(t)$ of the discrete learning rate $\overline{\alpha}_t$ (cf. (4)):

$$\frac{\mathrm{d}\overline{\alpha}(t)}{\mathrm{d}t} = -\beta(t)\overline{\alpha}(t), \quad \overline{\alpha}(T) = \overline{\alpha}_T. \tag{18c}$$

Given that the probability flow ODE (9) yields identical marginal distributions as the forward process X_t (cf. (6)) for every t, invoking (18c), we can easily see that $X(\overline{\alpha}(t)) \overset{\mathrm{d}}{=} X_t$ can be generated as follows:

$$\frac{\mathrm{d}X\big(\overline{\alpha}(t)\big)}{\mathrm{d}\overline{\alpha}(t)} = \frac{1}{2\overline{\alpha}(t)}\Big(X\big(\overline{\alpha}(t)\big) + s^\star_{\overline{\alpha}(t)}\Big(X\big(\overline{\alpha}(t)\big)\Big)\Big) \tag{19}$$

where $X\big(\overline{\alpha}(T)\big) \sim q_T$. By taking $f\big(\gamma\big) = \frac{1}{\sqrt{\gamma}}X\big(\gamma\big)$, we can apply (19) to derive

$$\frac{\mathrm{d}f\big(\gamma\big)}{\mathrm{d}\gamma} = -\frac{1}{2\sqrt{\gamma^3}}X\big(\gamma\big) + \frac{1}{\sqrt{\gamma}}\frac{\mathrm{d}X\big(\gamma\big)}{\mathrm{d}\gamma} = \frac{1}{2\sqrt{\gamma^3}}s_\gamma^\star\Big(X\big(\gamma\big)\Big).$$

This taken together with $\overline{\alpha}_t = \overline{\alpha}_{t-1}\alpha_t$ (cf. (4)) immediately implies that

$$\begin{aligned}
&\frac{1}{\sqrt{\overline{\alpha}_{t-1}}}X(\overline{\alpha}_{t-1}) = \\
&\qquad \frac{1}{\sqrt{\overline{\alpha}_t}}X(\overline{\alpha}_t) + \frac{1}{2}\int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} s_\gamma^\star\big(X(\gamma)\big)\mathrm{d}\gamma, \\
\Longrightarrow \quad & X(\overline{\alpha}_{t-1}) = \\
&\qquad \frac{1}{\sqrt{\alpha_t}}X(\overline{\alpha}_t) + \frac{\sqrt{\overline{\alpha}_{t-1}}}{2}\int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}} s_\gamma^\star\big(X(\gamma)\big)\mathrm{d}\gamma.
\end{aligned} \tag{20}$$

With this relation in mind, we are ready to discuss the following approximation in discrete time:

- *Scheme 1:* If we approximate $s_\gamma^\star\big(X(\gamma)\big)$ for $\gamma \in [\overline{\alpha}_t, \overline{\alpha}_{t-1}]$ by $s_\gamma^\star\big(X(\gamma)\big) \approx s_{\overline{\alpha}_t}^\star\big(X(\overline{\alpha}_t)\big) \approx s_t(X_t)$, then we arrive at

$$\begin{aligned}
X(\overline{\alpha}_{t-1}) &\approx \frac{1}{\sqrt{\alpha_t}}X(\overline{\alpha}_t) + \left(\frac{\sqrt{\overline{\alpha}_{t-1}}}{\sqrt{\overline{\alpha}_t}} - 1\right)s_t(X_t) \\
&\approx \frac{1}{\sqrt{\alpha_t}}\left\{X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2}s_t(X_t)\right\},
\end{aligned}$$

  where we use the facts that $\overline{\alpha}_t/\overline{\alpha}_{t-1} = \alpha_t$ and $\alpha_t \approx 1$. This coincides with the deterministic sampler (8).

- *Scheme 2:* If we invoke a more refined approximation for $s_\gamma^\star\big(X(\gamma)\big)$ as

$$\begin{aligned}
&s_\gamma^\star\big(X(\gamma)\big) \\
&\approx s_{\overline{\alpha}_t}^\star\big(X(\overline{\alpha}_t)\big) + \underbrace{\frac{\mathrm{d}s_\gamma^\star\big(X(\gamma)\big)}{\mathrm{d}\gamma}}_{\approx \frac{s^\star_{\overline{\alpha}_t}(X(\overline{\alpha}_t)) - s^\star_{\overline{\alpha}_{t+1}}(X(\overline{\alpha}_{t+1}))}{\overline{\alpha}_t - \overline{\alpha}_{t+1}}}(\gamma - \overline{\alpha}_t) \\
&\approx s_t(X_t) + \frac{\gamma - \overline{\alpha}_t}{\overline{\alpha}_t - \overline{\alpha}_{t+1}}\left(s_t\big(X_t\big) - s_{t+1}\big(X_{t+1}\big)\right),
\end{aligned} \tag{21}$$

  then (20) can be approximated by

$$\begin{aligned}
&X(\overline{\alpha}_{t-1}) \\
&\approx \frac{1}{\sqrt{\alpha_t}}X(\overline{\alpha}_t) + \frac{\sqrt{\overline{\alpha}_{t-1}}s_t\big(X_t\big)}{2}\int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}}\frac{1}{\sqrt{\gamma^3}}\mathrm{d}\gamma + \\
&\frac{\sqrt{\overline{\alpha}_{t-1}}\Big(s_t\big(X_t\big) - s_{t+1}\big(X_{t+1}\big)\Big)}{2(\overline{\alpha}_t - \overline{\alpha}_{t+1})}\int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}}\frac{\gamma - \overline{\alpha}_t}{\sqrt{\overline{\alpha}^3}}\mathrm{d}\gamma \\
&\approx \frac{1}{\sqrt{\alpha_t}}\left\{X(\overline{\alpha}_t) + \frac{1-\alpha_t}{2}s_t(X_t)\right. \\
&\qquad \left. + \frac{(1-\alpha_t)^2}{4(1-\alpha_{t+1})}\Big(s_t\big(X_t\big) - \sqrt{\alpha_{t+1}}s_{t+1}\big(X_{t+1}\big)\Big)\right\},
\end{aligned} \tag{22}$$

which resembles the proposed sampler (15), and is computationally more appealing since it reuses the previous score function evaluation.

It is worth noting that similar approximation as in Scheme 2 has been invoked previously in Lu et al. (2022a, Eqn (3.6)) to construct high-order ODE solvers (e.g., the DPM-Solver-2, with 2 indicating second-order ODEs). Consequently, the acceleration achieved by our sampler is achieved through ideas akin to the second-order ODE; in turn, our convergence guarantees shed light on the effectiveness of high-order ODE solvers like the popular DPM-Solver.

3.2. Accelerated SDE-based sampler

Next, we turn to stochastic samplers, and propose a new stochastic sampling procedure that enjoys improved convergence guarantees compared to the DDPM-type sampler (10). To be precise, the proposed sampler begins by drawing $Y_T \sim \mathcal{N}(0, I_d)$ and adopts the following update rule:

$$Y_t^+ = \Phi_t(Y_t, Z_t), \quad Y_{t-1} = \Psi_t(Y_t^+, Z_t^+) \tag{23a}$$

for $t = T, \dots, 1$, where $Z_t, Z_t^+ \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d)$, and

$$\Phi_t(x, z) = x + \sqrt{\frac{1-\alpha_t}{2}} z, \tag{23b}$$

$$\Psi_t(y, z) = \frac{1}{\sqrt{\alpha_t}}\Big(y + (1-\alpha_t)s_t(y)\Big) + \sqrt{\frac{1-\alpha_t}{2}} z. \tag{23c}$$

The key difference between the proposed sampler and the original DDPM-type sampler lies in the additional operation $\Phi_t(\cdot,\cdot)$. In this step, a random noise Z_t is injected into the current sample Y_t to obtain an intermediate point Y_t^+, which together with another random noise Z_t^+ is subsequently fed into $\Psi_t(\cdot,\cdot)$ — a mapping identical to (10).

Theoretical guarantees. Let us present the convergence guarantees of the proposed stochastic sampler and their implications, followed by some interpretation of the design rationale of the algorithm.

Theorem 3.2. *Suppose that Assumptions 2.2 and 2.3 hold. Then the proposed stochastic sampler (23) with the learning rate schedule (14) achieves*

$$\begin{aligned}\mathsf{TV}\big(q_1, p_1\big) &\le \sqrt{\frac{1}{2}\mathsf{KL}\big(q_1 \parallel p_1\big)} \\ &\le C_1 \frac{d^3 \log^{4.5} T}{T} + C_1\sqrt{d}\varepsilon_{\mathsf{score}} \log^{1.5} T\end{aligned} \tag{24}$$

for some universal constant $C_1 > 0$.

Theorem 3.2 provides non-asymptotic characterizations for the data generation quality of the accelerated stochastic sampler. In comparison with the convergence theory for the DDPM-type sampler — which has a convergence rate proportional to $1/\sqrt{T}$ (Chen et al., 2022; 2023a; Li et al., 2023; Benton et al., 2023a) — Theorem 3.2 asserts that the proposed accelerated sampler achieves a faster convergence rate proportional to $1/T$. In contrast to Theorem 3.1 for the ODE-based sampler, the SDE-based sampler does not require continuity of the Jacobian matrix (i.e., Assumption 2.4). As before, the total-variation distance between X_1 and Y_1 is proportional to the ℓ_2 score estimation error when T is sufficiently large, which covers a broad range of target data distributions with no requirement on the smoothness or log-concavity of the data distribution.

Interpretation via higher-order approximation. Now we provide some insights into the motivation of the proposed sampler. We start with the characterizations of conditional density $p_{X_{t-1}|X_t}$. Denoting $\mu_t^\star(x_t) := \frac{1}{\sqrt{\alpha_t}}\left(x_t + (1-\alpha_t)\, s_t^\star(x_t)\right)$, we can approximate $p_{X_{t-1}|X_t}$ by

$$\begin{aligned} p_{X_{t-1}\,|\,X_t}(x_{t-1}\,|\,x_t) &\approx \exp\Big(-\frac{\alpha_t}{2(1-\alpha_t)}\cdot \\ &\Big\|\Big(I + \frac{1-\alpha_t}{2}J_{s_t^\star}(x_t)\Big)^{-1}\big(x_{t-1} - \mu_t^\star(x_t)\big)\Big\|_2^2\Big). \end{aligned} \tag{25}$$

which is tighter than the one used in analysis of the original SDE-based sampler (Li et al., 2023) by adopting a higher-order expansion. This in turn motivates us to consider the following sequence

$$\begin{aligned} Y_{t-1} = \sqrt{\frac{1-\alpha_t}{2}} Z_t^+ + \frac{1}{\sqrt{\alpha_t}}\Big(&\underbrace{Y_t + \sqrt{\frac{1-\alpha_t}{2}} Z_t}_{\Phi(Y_t, Z_t)} \\ &+ (1-\alpha_t)\underbrace{\Big(s_t^\star(Y_t) + \sqrt{\frac{1-\alpha_t}{2}} J_{s_t^\star}(Y_t) Z_t\Big)}_{\approx s_t^\star\left(\Phi(Y_t, Z_t)\right)}\Big) \end{aligned} \tag{26}$$

with $Z_t, Z_t^+ \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d)$. Note that $p_{Y_{t-1}|Y_t}(x_{t-1} \mid x_t)$ follows $\mathcal{N}\big(\mu_t^\star(x_t), \Sigma_t^\star(x_t)\big)$; here,

$$\Sigma_t^\star(x_t) = \tfrac{1-\alpha_t}{\alpha_t}\left(I + \tfrac{1-\alpha_t}{2}J_{s_t^\star}(x_t)\right)\left(I + \tfrac{1-\alpha_t}{2}J_{s_t^\star}(x_t)\right)^\top,$$

which aligns with (25). In addition, if we further employ $s_t^\star(Y_t) + \sqrt{\frac{1-\alpha_t}{2}} J_{s_t^\star}(Y_t) Z_t$ as a first-order approximation of $s_t^\star\big(Y_t + \sqrt{\frac{1-\alpha_t}{2}} Z_t\big)$, then we can arrive at the update rule of the proposed sampler in (23).

4. Experiments

In this section, we illustrate the performance of the proposed accelerated samplers, focusing on emphasizing the relative

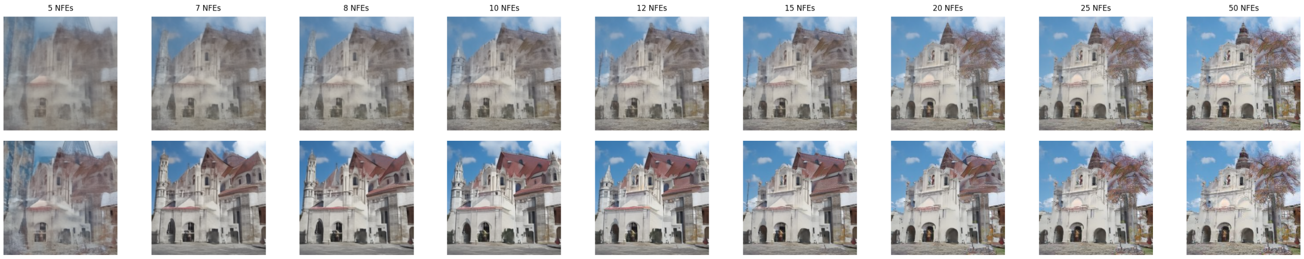
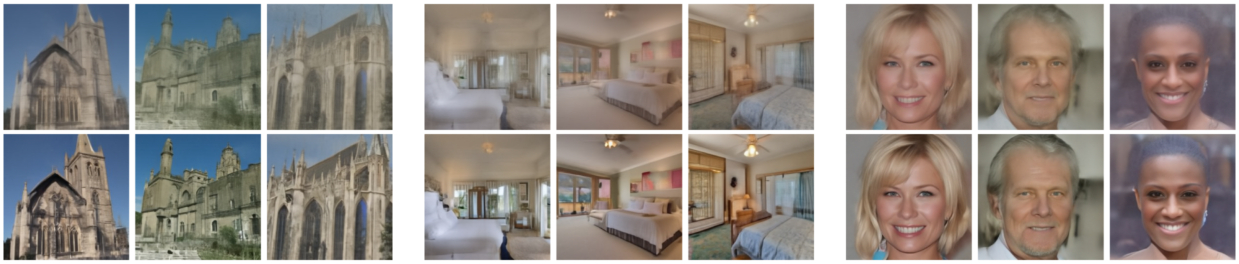


Figure 1. The progress of the generated samples over different numbers of NFEs (5 to 50), using pre-trained scores from the LSUN-Churches dataset. Top row: the vanilla DDIM-type sampler. Bottom row: the accelerated DDIM-type sampler (ours).



(a) LSUN-Churches

(b) LSUN-Bedroom

(c) CelebA-HQ

Figure 2. Examples of sampled images from the DDIM-type samplers with 5 NFEs, using pre-trained scores from the LSUN-Churches, LSUN-Bedroom, and CelebA-HQ datasets. For each dataset, the top image is the original DDIM-type sampler, and the bottom image is the accelerated DDIM-type sampler (ours).

comparisons with respect to the original DDIM/DDPM ones using the same pre-trained score functions. We specifically report results for deterministic samplers here, leaving the stochastic setting to Appendix A.

4.1. Practical implementation

In practice, the pre-trained score functions are often available in the form of noise-prediction networks $\epsilon_t(\cdot)$, which are connected via the following relationship in view of (7):

$$s_t^\star(X) := -\frac{1}{\sqrt{1-\overline{\alpha}_t}}\epsilon_t^\star(X), \tag{27}$$

and $\epsilon_t(\cdot)$ is the estimate of $\epsilon_t^\star(\cdot)$. To better align with the empirical practice, it is judicious that the integration in (20) be approximated in terms of $\epsilon_t^\star(X)$, leading to an equivalent rewrite as

$$X(\overline{\alpha}_{t-1}) = \frac{1}{\sqrt{\alpha_t}}X(\overline{\alpha}_t) - \frac{\sqrt{\overline{\alpha}_{t-1}}}{2}\int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3}\sqrt{1-\gamma}}\epsilon_\gamma^\star\big(X(\gamma)\big)\mathrm{d}\gamma.$$

Following similar discussions in Section 3.1, we discuss its first-order and second-order approximations in discrete time.

- *Scheme 1:* If we approximate $\epsilon_\gamma^\star\big(X(\gamma)\big)$ for $\gamma \in [\overline{\alpha}_t, \overline{\alpha}_{t-1}]$ by $\epsilon_\gamma^\star\big(X(\gamma)\big) \approx \epsilon_{\overline{\alpha}_t}^\star\big(X(\overline{\alpha}_t)\big) \approx \epsilon_t(X_t)$, then we arrive at

$$X(\overline{\alpha}_{t-1}) \approx \frac{1}{\sqrt{\alpha_t}}X(\overline{\alpha}_t) + \left(\sqrt{1-\overline{\alpha}_{t-1}} - \frac{\sqrt{1-\overline{\alpha}_t}}{\sqrt{\alpha_t}}\right)\epsilon_t(X_t), \tag{28}$$

which matches exactly with the DDIM sampler in Song et al. (2020).

- *Scheme 2:* If we invoke the refined approximation (21) in terms of $\epsilon_\gamma^\star\big(X(\gamma)\big)$, we have

$$\begin{aligned} X(\overline{\alpha}_{t-1}) &\approx \frac{1}{\sqrt{\alpha_t}}X(\overline{\alpha}_t) - \frac{\sqrt{\overline{\alpha}_{t-1}}\epsilon_t\big(X_t\big)}{2}\int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{1}{\sqrt{\gamma^3(1-\gamma)}}\mathrm{d}\gamma \\ &\quad - \frac{\sqrt{\overline{\alpha}_{t-1}}\Big(\epsilon_t\big(X_t\big)-\epsilon_{t+1}\big(X_{t+1}\big)\Big)}{2(\overline{\alpha}_t-\overline{\alpha}_{t+1})}\int_{\overline{\alpha}_t}^{\overline{\alpha}_{t-1}} \frac{(\gamma-\overline{\alpha}_t)}{\sqrt{\gamma^3(1-\gamma)}}\mathrm{d}\gamma, \end{aligned} \tag{29}$$

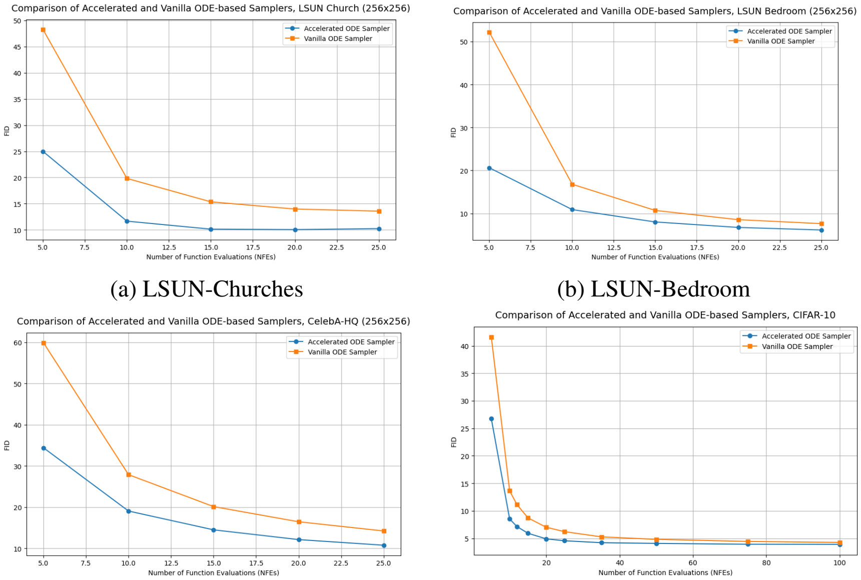
which after integration becomes:

$$\begin{aligned} X(\overline{\alpha}_{t-1}) &\approx \frac{1}{\sqrt{\alpha_t}}X(\overline{\alpha}_t) + \left(\sqrt{1-\overline{\alpha}_{t-1}} - \frac{\sqrt{1-\overline{\alpha}_t}}{\sqrt{\alpha_t}}\right)\epsilon_t(X_t) \\ &\quad + \left(\frac{\sqrt{\overline{\alpha}_{t-1}}}{\overline{\alpha}_t-\overline{\alpha}_{t+1}}\right)\Bigg(\overline{\alpha}_t\frac{\sqrt{1-\overline{\alpha}_{t-1}}}{\sqrt{\overline{\alpha}_{t-1}}} + \arcsin\sqrt{\overline{\alpha}_{t-1}} \\ &\quad -\overline{\alpha}_t\frac{\sqrt{1-\overline{\alpha}_t}}{\sqrt{\overline{\alpha}_t}} - \arcsin\sqrt{\overline{\alpha}_t}\Bigg)\big(\epsilon_{t+1}(X_{t+1}) - \epsilon_t(X_t)\big). \end{aligned} \tag{30}$$

This is our new sampler for implementation.

4.2. Experimental results

We use pre-trained score functions from Huggingface (von Platen et al., 2022) for the CelebA-HQ, LSUN-Bedroom, and LSUN-Churches datasets. Moreover, for the CIFAR-10



(a) LSUN-Churches

(b) LSUN-Bedroom

(c) CelebA-HQ

(d) CIFAR-10

Figure 3. The FID of the DDIM-type samplers for different datasets with respect to the NFEs.

dataset, we utilize pre-trained score functions from Ho et al. (2020) and the DPM-Solver codebase (Lu et al., 2022a). Note that we have not attempted to optimize the speed nor the performance using additional tricks, e.g., employing better score functions, but aim to corroborate our theoretical findings regarding the acceleration of the new samplers without training additional functions when the implementations are otherwise kept the same.

Figure 1 illustrates the progress of the generated samples over different numbers of function evaluations (NFEs) (between 5 and 50) from the same random seed, using pre-trained scores from the LSUN-Churches dataset. Here, the NFE is the same as the number of diffusion steps since each step takes one score evaluation. Our proposed accelerated DDIM-type sampler (cf. (28)) can generate high-quality images witin 10 NFEs, while the vanilla DDIM-type sampler (cf. (30)) requires more NFEs to achieve similar quality.

To further demonstrate the quality of the sampled images, Figure 2 provides examples of sampled images from the DDIM-type samplers with 5 NFEs, using pre-trained scores from CelebA-HQ, LSUN-Bedroom and LSUN-Churches datasets, respectively. It can be seen that the sampled images are crisper and less noisy from the accelerated DDIM-type sampler, compared with from the original one, indicating the effectiveness of our method.

As for the quantitative results, the FID scores during the sampling precoss for different datasets are provided in Figure 3. The quantitative advantage of the proposed deterministic sampler is highlighted by achieving FID scores that are halved compared to vanilla DDIM, using just 5 steps.

5. Discussion

In this paper, we have developed novel strategies to achieve provable acceleration in score-based generative modeling. The proposed deterministic sampler achieves a convergence rate $1/T^2$ that substantially improves upon prior theory for the probability flow ODE approach, whereas the proposed stochastic sampler enjoys a converge rate $1/T$ that also significantly outperforms the convergence theory for the DDPM-type sampler. We have demonstrated the stability of these samplers, establishing non-asymptotic theoretical guarantees that hold in the presence of ℓ_2-accurate score estimates. Our algorithm development for the deterministic case draws inspiration from higher-order ODE approximations in discrete time, which might shed light on understanding popular ODE-based samplers like the DPM-Solver. In comparison, the accelerated stochastic sampler is designed based on higher-order expansions of the conditional density.

Our findings further suggest multiple directions that are worthy of future exploration. For instance, our convergence theory remains sub-optimal in terms of the dependency on the problem dimension d, which calls for a more refined theory to sharpen dimension dependency. Additionally, given the conceptual similarity between our accelerated deterministic sampler and second-order ODE, it would be interesting to extend the algorithm and theory using ideas arising from third-order or even higher-order ODE. In particular, third-order ODE has been implemented in DPM-Solver-3, which is among the most effective DPM-Solvers in practice. Finally, it would be important to design higher-order solvers for SDE-based samplers, in order to unveil the degree of acceleration that can be achieved through high-order SDE.

Acknowledgements

G. Li is supported in part by the Chinese University of Hong Kong Direct Grant for Research. Y. Wei is supported in part by the NSF grants DMS-2147546/2015447, CAREER award DMS-2143215, CCF-2106778, CCF-2418156, and the Google Research Scholar Award. The work of T. Efimov and Y. Chi is supported in part by the grants ONR N00014-19-1-2404, NSF DMS-2134080, ECCS-2126634 and FHWA 693JJ321C000013. Y. Chen is supported in part by the Alfred P. Sloan Research Fellowship, the Google Research Scholar Award, the AFOSR grant FA9550-22-1-0198, the ONR grant N00014-22-1-2354, and the NSF grants CCF-2221009 and CCF-1907661.

Impact Statement

This paper aims to advance the theoretical and practical aspects of accelerated diffusion models. We recognize that while this technology has the potential for positive impacts in data processing and information dissemination, it also poses challenges in privacy and information security. Therefore, we emphasize the need for careful consideration of its effects on personal data protection and societal information flow when applying these models.

References

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

Benton, J., De Bortoli, V., Doucet, A., and Deligiannidis, G. (2023a). Linear convergence bounds for diffusion models via stochastic localization. *arXiv preprint arXiv:2308.03686*.

Benton, J., Deligiannidis, G., and Doucet, A. (2023b). Error bounds for flow matching methods. *arXiv preprint arXiv:2305.16860*.

Block, A., Mroueh, Y., and Rakhlin, A. (2020). Generative modeling with denoising auto-encoders and Langevin sampling. *arXiv preprint arXiv:2002.00107*.

Chen, H., Lee, H., and Lu, J. (2023a). Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. In *International Conference on Machine Learning*, pages 4735–4763.

Chen, H. and Ying, L. (2024). Convergence analysis of discrete diffusion model: Exact implementation through uniformization. *arXiv preprint arXiv:2402.08095*.

Chen, S., Chewi, S., Lee, H., Li, Y., Lu, J., and Salim, A. (2023b). The probability flow ODE is provably fast. *Neural Information Processing Systems*.

Chen, S., Chewi, S., Li, J., Li, Y., Salim, A., and Zhang, A. R. (2022). Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.11215*.

Chen, S., Daras, G., and Dimakis, A. (2023c). Restoration-degradation beyond linear diffusions: A non-asymptotic analysis for DDIM-type samplers. In *International Conference on Machine Learning*, pages 4462–4484.

Croitoru, F.-A., Hondru, V., Ionescu, R. T., and Shah, M. (2023). Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

De Bortoli, V. (2022). Convergence of denoising diffusion models under the manifold hypothesis. *arXiv preprint arXiv:2208.05314*.

De Bortoli, V., Thornton, J., Heng, J., and Doucet, A. (2021). Diffusion Schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709.

Dhariwal, P. and Nichol, A. (2021). Diffusion models beat GANs on image synthesis. *Advances in neural information processing systems*, 34:8780–8794.

Gao, X., Nguyen, H. M., and Zhu, L. (2023). Wasserstein convergence guarantees for a general class of score-based generative models. *arXiv preprint arXiv:2311.11003*.

Gao, X. and Zhu, L. (2024). Convergence analysis for general probability flow ODEs of diffusion models in Wasserstein distances. *arXiv preprint arXiv:2401.17958*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Guo, Z., Liu, J., Wang, Y., Chen, M., Wang, D., Xu, D., and Cheng, J. (2023). Diffusion models in bioinformatics and computational biology. *Nature Reviews Bioengineering*, pages 1–19.

Haussmann, U. G. and Pardoux, E. (1986). Time reversal of diffusions. *The Annals of Probability*, pages 1188–1205.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4).

Hyvärinen, A. (2007). Some extensions of score matching. *Computational statistics & data analysis*, 51(5):2499–2512.

Kazerouni, A., Aghdam, E. K., Heidari, M., Azad, R., Fayyaz, M., Hacihaliloglu, I., and Merhof, D. (2023). Diffusion models in medical imaging: A comprehensive survey. *Medical Image Analysis*, page 102846.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. *International Conference on Learning Representations*.

Lee, H., Lu, J., and Tan, Y. (2022). Convergence for score-based generative modeling with polynomial complexity. In *Advances in Neural Information Processing Systems*.

Lee, H., Lu, J., and Tan, Y. (2023). Convergence of score-based generative modeling for general data distributions. In *International Conference on Algorithmic Learning Theory*, pages 946–985.

Li, G., Huang, Y., Efimov, T., Wei, Y., Chi, Y., and Chen, Y. (2024a). Accelerating convergence of score-based diffusion models, provably. *arXiv preprint arXiv:2403.03852*.

Li, G., Huang, Z., and Wei, Y. (2024b). Towards a mathematical theory for consistency training in diffusion models. *arXiv preprint arXiv:2402.07802*.

Li, G., Wei, Y., Chen, Y., and Chi, Y. (2023). Towards faster non-asymptotic convergence for diffusion-based generative models. *arXiv preprint arXiv:2306.09251*.

Li, S., Chen, S., and Li, Q. (2024c). A good score does not lead to a good generative model. *arXiv preprint arXiv:2401.04856*.

Liang, Y., Ju, P., Liang, Y., and Shroff, N. (2024). Non-asymptotic convergence of discrete-time diffusion models: New approach and improved rate. *arXiv preprint arXiv:2402.13901*.

Liu, L., Ren, Y., Lin, Z., and Zhao, Z. (2022a). Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*.

Liu, X., Wu, L., Ye, M., and Liu, Q. (2022b). Let us build bridges: Understanding and extending diffusion generative models. *arXiv preprint arXiv:2208.14699*.

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*, 35:5775–5787.

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 preprint arXiv:2211.01095*.

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

Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., and Salimans, T. (2023). On distillation of guided diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306.

Nichol, A. Q. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171.

Pang, T., Xu, K., Li, C., Song, Y., Ermon, S., and Zhu, J. (2020). Efficient learning of generative models via finite-difference score matching. *Advances in Neural Information Processing Systems*, 33:19175–19188.

Pidstrigach, J. (2022). Score-based generative models detect manifolds. *arXiv preprint arXiv:2206.01018*.

Salimans, T. and Ho, J. (2021). Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265.

Song, J., Meng, C., and Ermon, S. (2020). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. (2023). Consistency models. In *International Conference on Machine Learning*.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

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

Tang, W. and Zhao, H. (2024a). Contractive diffusion probabilistic models. *arXiv preprint arXiv:2401.13115*.

Tang, W. and Zhao, H. (2024b). Score-based diffusion models via stochastic differential equations–a technical tutorial. *arXiv preprint arXiv:2402.07487*.

Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674.

von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., and Wolf, T. (2022). Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers.

Wang, Z., Zheng, H., He, P., Chen, W., and Zhou, M. (2022). Diffusion-gan: Training gans with diffusion. *arXiv preprint arXiv:2206.02262*.

Wibisono, A. and Yang, K. Y. (2022). Convergence in KL divergence of the inexact Langevin algorithm with application to score-based generative models. *arXiv preprint arXiv:2211.01512*.

Wu, Y., Chen, M., Li, Z., Wang, M., and Wei, Y. (2024). Theoretical insights for diffusion guidance: A case study for gaussian mixture models. *arXiv preprint arXiv:arXiv:2403.01639*.

Xiao, Z., Kreis, K., and Vahdat, A. (2021). Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*.

Xue, S., Yi, M., Luo, W., Zhang, S., Sun, J., Li, Z., and Ma, Z.-M. (2023). SA-Solver: Stochastic Adams solver for fast sampling of diffusion models. *arXiv preprint arXiv:2309.05019*.

Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., and Yang, M.-H. (2023). Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39.

Zhang, Q. and Chen, Y. (2022). Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*.

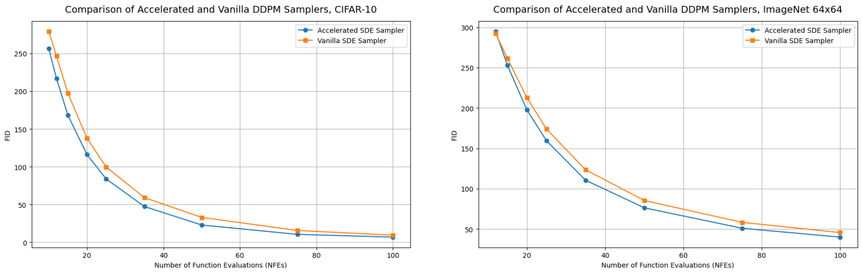
Zhang, Q., Tao, M., and Chen, Y. (2022). gddim: Generalized denoising diffusion implicit models. *arXiv preprint arXiv:2206.05564*.

Zhao, W., Bai, L., Rao, Y., Zhou, J., and Lu, J. (2023). UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. *arXiv preprint arXiv:2302.04867*.

Appendix

A. Additional experiments

To provide further quantitative comparisons, we calculate the FID scores of the generated images for DDPM-type samplers. For ImageNet, we use the pre-trained score functions from Improved DDPM (Nichol and Dhariwal, 2021). As shown in Figure 4, the slight FID gap still reflects the relative difference in image quality between the two samplers, with the accelerated DDPM sampler consistently outperforming the original sampler.



(a) Cifar10

(b) ImageNet64

Figure 4. The FID of the DDPM-type samplers for different datasets with respect to the NFEs, where the accelerated sampler consistently outperforms the original one.