Accelerating Score-based Generative Models with Preconditioned Diffusion Sampling

Hengyuan Ma[1], Li Zhang[1⋆], Xiatian Zhu[2], and Jianfeng Feng[1]

[1] Fudan University
[2] University of Surrey

https://github.com/fudan-zvg/PDS

Abstract. Score-based generative models (SGMs) have recently emerged as a promising class of generative models. However, a fundamental limitation is that their inference is very slow due to a need for many (*e.g.*, 2000) iterations of sequential computations. An intuitive acceleration method is to reduce the sampling iterations which however causes severe performance degradation. We investigate this problem by viewing the diffusion sampling process as a Metropolis adjusted Langevin algorithm, which helps reveal the underlying cause to be ill-conditioned curvature. Under this insight, we propose a model-agnostic ***preconditioned diffusion sampling*** (PDS) method that leverages matrix preconditioning to alleviate the aforementioned problem. Crucially, PDS is proven theoretically to converge to the original target distribution of a SGM, no need for retraining. Extensive experiments on three image datasets with a variety of resolutions and diversity validate that PDS consistently accelerates off-the-shelf SGMs whilst maintaining the synthesis quality. In particular, PDS can accelerate by up to 29× on more challenging high resolution (1024×1024) image generation.

Keywords: Image synthesis, score-based generative model, matrix preconditioning, ill-conditioned curvature.

1 Introduction

As an alternative framework to generative adversarial networks (GANs) [10], recent score-based generative models (SGMs) [31,32,33,30] have demonstrated excellent abilities in data synthesis (especially in high resolution images) with easier optimization [31], richer diversity [36], and more solid theoretic foundation [5]. Starting from a sample initialized with a Gaussian distribution, a SGM produces a target sample by simulating a diffusion process, typically a Langevin

⋆ Li Zhang (lizhangfd@fudan.edu.cn) is the corresponding author with School of Data Science, Fudan University. H. Ma and J. Feng are with Institute of Science and Technology for Brain-inspired Intelligence, Fudan University. X. Zhu is with Surrey Institute for People-Centred Artificial Intelligence, CVSSP, University of Surrey.

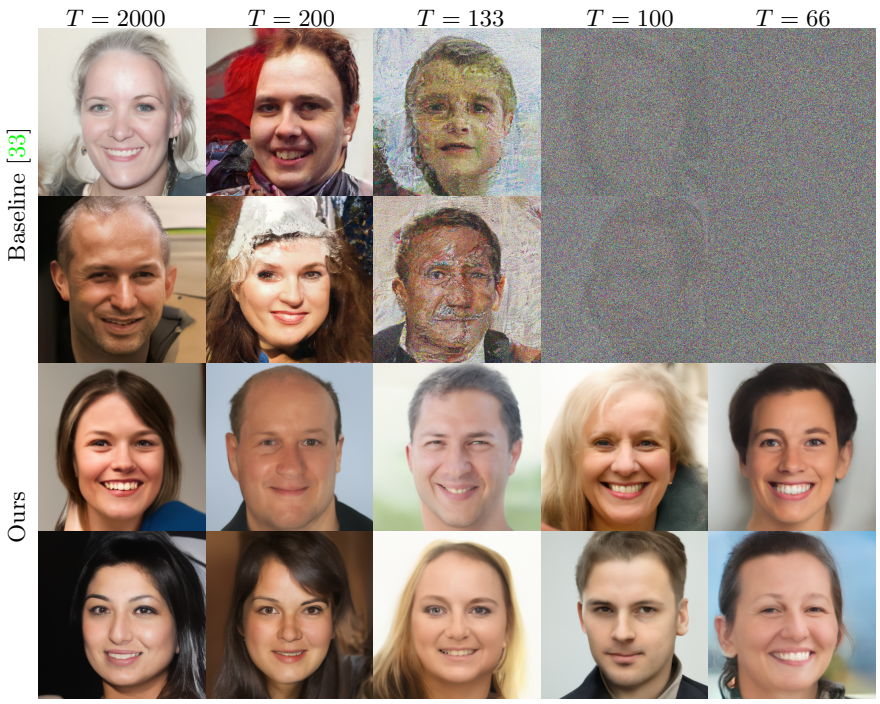

Fig. 1. Facial images at a resolution of 1024 × 1024 generated by NCSN++ [33] under a variety of sampling iterations (top) without and (bottom) with our PDS. It is evident that NCSN++ decades quickly with increasingly reduced sampling iterations, which can be well solved with PDS. In terms of running speed for generating a batch of 8 images, PDS reduces the time cost from 2030 seconds (the sampling iterations $T = 2000$) to 71 seconds ($T = 66$) on one NVIDIA RTX 3090 GPU, which delivers 29× acceleration. Dataset: FFHQ [18]. More samples in supplementary material.

dynamics. Compared to the state-of-the-art GANs [4,18,17], a significant drawback with existing SGMs is *drastically slower generation* due to the need of taking many iterations for a sequential diffusion process [33,23,36]. Formally, the discrete Langevin dynamic for sampling is typically formulated as

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \frac{\epsilon_t^2}{2} \bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x}_{t-1}) + \epsilon_t \mathbf{z}_t, 1 \leq t \leq T \tag{1}$$

where ϵ_t is the step size (a positive real scalar), $\mathbf{z}_t$ is an independent standard Gaussian noise, and T is the iteration number. Starting from a standard Gaussian sample $\mathbf{x}_0$, with a total of T steps this sequential sampling process gradually transforms $\mathbf{x}_0$ to the sample $\mathbf{x}_T$ that obeys the target distribution p^*. Often, T is at the scale of 1000s, and the entire sampling process is lengthy.

For accelerating the sampling process, a straightforward method is to reduce T by a factor and proportionally expand ϵ_t simultaneously, so that the number of calculating the gradient $\bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x})$, which consumes the major time, de-

creases whilst keeping the total update magnitude. However, this often makes pretrained SGMs fail in image synthesis. In general, we observe two types of failure: insufficient detailed structures (left of Fig. 3 and Fig. 4), and dazzling with heavy noises (left of Fig. 1 and Fig. 5). Conceptually, the sampling process as defined in Eq. (1) can be considered as a special case of Metropolis adjusted Langevin algorithm (MALA) [27,35,9]. When the coordinates of a target sample (*e.g.*, the pixel locations of a natural image) are strongly correlated, the isotropic Gaussian noises $\{\mathbf{z}_t\}$ would become *inefficient* for the variables $\mathbf{x}$, caused by the *ill-conditioned curvature* of the sampling process [9].

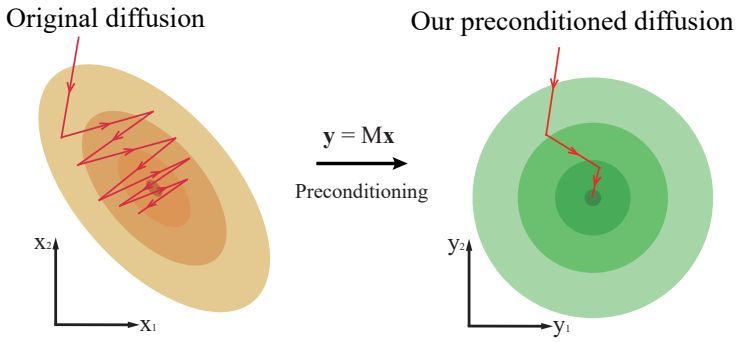

Fig. 2. Illustration of the preconditioning method for accelerating sampling process.

In light of this insight as above, we propose an efficient, model-agnostic ***preconditioned diffusion sampling*** (PDS) method for accelerating existing pretrained SGMs without the need for model retraining. The key idea is to make the rates of curvature become more similar along all the directions [27,21] using a *matrix preconditioning*, hence solving the ill-conditioned curvature problem, as demonstrated in Fig. 2. Formally, we enrich the above Langevin dynamics (Eq. (1)) by imposing a preconditioning operation into the diffusion process as

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \frac{\epsilon_t^2}{2} M M^{\mathsf{T}} \bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x}_{t-1}) + \epsilon_t M \mathbf{z}_t, \tag{2}$$

where M is the newly introduced preconditioning matrix designed particularly for regulating the behavior of accelerated diffusion processes. This proposed reformulation equips the diffusion process with a novel ability to *enhance* or *restrain* the generation of detailed structures via controlling the different frequency components[3] of the noises [2]. Crucially, according to the theorems with Fokker-Planck equation [8] PDS can preserve the original SGM's target distribution. Further, structured priors available with a target distribution can be also

[3] More theoretical explanation on why *directly* regulating the frequency domain of a diffusion process is possible is provided in Supplementary material .

accommodated, *e.g.*, the spatial structures of human faces. The computational cost of calculating M is marginal when using Fast Fourier Transform (FFT) [3]. In this work, we make the following **contributions**: **(1)** We investigate the low inference efficiency problem of off-the-shelf SGMs for high-resolution image synthesis, which is critical yet under-studied in the literature. **(2)** For sampling acceleration, we introduce a novel preconditioned diffusion sampling (PDS) process. PDS preconditions the existing diffusion process additionally imposed for adaptively regulating the added noises, whilst keeping the original target distributions in convergence. **(3)** With PDS, a variety of pretrained SGMs can be accelerated significantly for image synthesis of various spatial resolutions, without model retraining. In particular, PDS delivers 29× reduction in wall-clock time for high-resolution image synthesis.

2 Related work

Sohl-Dickstein et al. [28] first proposed to destroy the data distribution through a diffusion process slowly and learned the backward process to recover the data, inspired by non-equilibrium statistical physics. Later on, Song and Ermon [31] further explored SGMs by introducing the noise conditional score network (NCSN). Song and Ermon [32] proposed NCSNv2 that scaled NCSN for higher resolution image generation (*e.g.*, 256×256) by scaling noises and improving stability with moving average. Song et al. [33] summarized all the previous SGMs into a unified framework based on the stochastic differential equation (SDE) and proposed the NCSN++ model to generate high-resolution images via numerical SDE solvers for the first time. Bortoli et al. [5] provided the first quantitative convergence results for SGMs. Vahdat et al. [34] developed Latent Score-based Generative Model (LSGM) that trains SGMs in a latent space with the variational autoencoder framework. Another class of relevant generative models, mainly trained by reducing an evidence lower bound (ELBO) called denoising diffusion probabilistic models (DDPMs) [12,24,29,6,13,23,1], also demonstrate excellent performance on image synthesis. Commonly, all of the above works use isotropic Gaussian distributions for the diffusion sampling.

Recently there are some works proposed on accelerating SGMs. Dockhorn et al. [7] improved the SGMs with Hamiltonian Monte Carlo methods [22] and proposed critically-damped Langevin diffusion (CLD) based SGMs that achieves superior performance. Jolicoeur-Martineau et al. [16] utilized a numerical SDE solver with adaptive step sizes to accelerate SGMs. However, these methods are limited in the following aspects: **(1)** They tend to involve much extra computation. For example, CLD based SGMs expand the dimension of data by 2 times for learning the velocity of the diffusion. Jolicoeur-Martineau et al. [16] added a high-order numerical solver that increases the number of calling the SGM, resulting in much more time. In comparison, with our PDS the only extra calculation relates the preconditioning matrix that can be efficiently implemented by Fast Fourier Transform. **(2)** They are restricted to a single specific SGM while our PDS is model agnostic. **(3)** Unlike this work, none of them has demonstrated

a scalability to more challenging high-resolution image generation tasks (*e.g.*, FFHQ facial images).

3 Preliminary

Scored-based generative models (SGMs). Score matching is developed for non-normalized statistical learning [15]. Given i.i.d. samples of an unknown distribution p^*, score matching allows the model to directly approximate the *score function* $\bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x})$. SGMs aim to generate samples from p^* via score matching by simulating a Langevin dynamics initialized by Gaussian noise

$$d\mathbf{x} = \frac{g^2(t)}{2} \bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x}) dt + g(t) d\mathbf{w}, \tag{3}$$

where $g : \mathbb{R}^+ \to \mathbb{R}^+$ controls the step size and $d\mathbf{w}$ represents a Wiener process. With this process, we transform a sample drawn from an initial Gaussian distribution to approach the desired distribution p^*. A classical SGM, noise conditional score network (NCSN) [31], is trained by learning how to reverse a process of gradually corrupting the samples from p^*, and aims to match the score function. After training, NCSN starts from a Gaussian distribution and travels to the target distribution p^* by simulating an annealed Langevin dynamics.

Recent improvements. Song and Ermon [32] presented NCSNv2 that improves the original NCSN by designing better noise scales, iteration number, and step size. This new variant is also more stable by using the moving average technique. Song et al. [33] further proposed NCSN++ that utilizes an existing numerical solver of stochastic differential equations to enhance both the speed of convergence and the stability of the sampling method. Importantly, NCSN++ can synthesize high-resolution images at high quality.

Limitation analysis. Although SGMs have been able to generate images comparable to GANs [10], they are much slower due to the sequential computation during the sampling phase. For example, to produce 8 facial images at 1024×1024 resolution, a SGM spends more than 30 mins. To maximize the potential of SGMs, it is critical to solve this slow inference bottleneck.

4 Method

We aim to solve the slow inference problem with SGMs. For easier understanding, let us start from the most classical Langevin dynamics.

4.1 Steady-state distribution analysis

Consider the classical Langevin dynamics

$$d\mathbf{x} = \frac{\epsilon^2}{2} \bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x}) dt + \epsilon d\mathbf{w}, \tag{4}$$

where p^* is the target distribution, and $\epsilon > 0$ is the fixed step size. It is associated with a Fokker-Planck equation

$$\frac{\partial p}{\partial t} = -\frac{\epsilon^2}{2} \bigtriangledown_{\mathbf{x}} \cdot (\bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x}) p) + \frac{\epsilon^2}{2} \Delta_{\mathbf{x}} p, \tag{5}$$

where $p = p(\mathbf{x}, t)$ describes the distribution of $\mathbf{x}$ that evolves over time. The steady-state solution of Eq. (5) corresponds to the probabilistic density function of the steady-state distribution of Eq. (4), i.e., p^*

$$\bigtriangledown_{\mathbf{x}} \cdot (\bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x}) p) = \Delta_{\mathbf{x}} p. \tag{6}$$

The Fokker-Planck equation tells us how to preserve the steady-state distribution of the original process when we alter Eq. (4) for specific motivations. Concretely, we can impose an invertible linear operator M to the noise term $d\mathbf{w}$ and conduct the associated operation on the gradient term so that the steady-state distribution can be preserved. This design is formulated as:

$$d\mathbf{x} = \frac{\epsilon^2}{2}(MM^{\mathsf{T}} + S) \bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x}) dt + \epsilon M d\mathbf{w}, \tag{7}$$

where S is a skew-symmetric linear operator. In fact, we have

Theorem 1. *The steady-state distribution of Eq. (4) and Eq. (7) are the same, as long as the linear operator M is invertible and the linear operator S is skew-symmetric.*

We can extend the above results to a more general case as follows.

Theorem 2. *Consider the diffusion process*

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t) dt + G(t) d\mathbf{w}, \tag{8}$$

where $\mathbf{f} : \mathbb{R}^d \otimes \mathbb{R} \to \mathbb{R}^d$, $G : \mathbb{R} \to \mathbb{R}^{d \times d}$. M is an invertible $d \times d$ matrix and S is a skew-symmetric $d \times d$ matrix. Denote p^* as the steady-state distribution of Eq. (8), then the process

$$d\mathbf{x} = MM^{\mathsf{T}} \mathbf{f}(\mathbf{x}, t) dt + S \bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x}) dt + MG(t) d\mathbf{w}, \tag{9}$$

has the same steady-state distribution as Eq. (8), given $G(t)G(t)^{\mathsf{T}}$ and M^{T} are commutable $\forall t$.

Remark 1. The conditions of this theorem are all satisfied for the diffusion process used in NCSN, NCSNv2, and NCSN++.

Thm. 2 motivates us to design a preconditioning matrix as Eq. (7) while keeping the steady-state distribution simultaneously. This is also because, preconditioning has been proved to be able to significantly accelerate the stochastic gradient descent algorithm (SGD) and Metropolis adjusted Langevin algorithms (MALA) [27]. Besides, SGD provides another view for interpreting our method, that is, SGMs sequentially reduce the energy ($-\log p^*(\mathbf{x})$) of a sample $\mathbf{x}$ via stochastic gradient descent, with the randomness coming from the Gaussian noises added at every single step.

4.2 Preconditioned diffusion sampling

We study how to construct the preconditioning operator using M to accelerate the sampling phase of SGMs, with $S = 0$ for Eq. (7). It is observed that when reducing the iteration number for the sampling process of a SGM and expand the step size proportionally for a consistent accumulative update, the images generated tend to miss necessary detailed structures (see left of Fig. 3 and Fig. 4), or involve high-frequency noises (left of Fig. 1 and Fig. 5). These failure phenomena motivates us to leverage a preconditioning operator M serving as a filter to regulate the frequency distribution of the samples.

1. Given an input vector $\mathbf{x}$, we first use Fast Fourier Transform (FFT) [3] to map it into the frequency domain $\hat{\mathbf{x}} = F[\mathbf{x}]$. For images, we adopt the 2D FFT that implements 1D FFT column-wise and row-wise successively.
2. Then we adjust the frequency signal using a mask R in the same shape as $\mathbf{x}$: $R \odot \hat{\mathbf{x}}$, where $\odot$ means element-wise multiplication.
3. Lastly, we map the vector back to the original space by the inverse of Fast Fourier Transform: $F^{-1}[R \odot \hat{\mathbf{x}}]$.

For specific tasks (*e.g.*, human facial image generation), most samples might share a consistent structural characteristics. This prior knowledge however is unavailable with the noises added to each step in the diffusion process. To solve this problem, we further propose a space structure filter A for ***space preconditioning***, constructed by statistical average of random samples. This can be used to regulate the noise via element-wise multiplication as: $A \odot [\cdot]$. Combining the both operations above, we define a preconditioning operator M as

$$M[\cdot] = A \odot F^{-1}[R \odot F[\cdot]]. \tag{10}$$

To guarantee the invertibility of M, we set the elements of R strictly positive. For the tasks without clear space structure priors, we simply do not apply the space preconditioning by setting all the elements of A to 1. We operate M on the noise term $d\mathbf{w}$ and adjust the gradient term to keep the steady-state distribution as shown in Eq. (7), utilizing Thm. 1.

Interestingly, we found that the proposed method above is likely to even cause further model degradation. This is because, if we implement a variable transformation as $\mathbf{y} = M^{-1}\mathbf{x}$, Eq. (7) can be rewritten as

$$d\mathbf{y} = \frac{\epsilon^2}{2} \bigtriangledown_{\mathbf{y}} \log p^*(\mathbf{y}) dt + \epsilon d\mathbf{w},$$

which returns to the same format as the original process. The diffusion process is made worse since, M^{-1}, the inverse of M, could impose the exactly opposite effect of M. To overcome this challenge, we further substitute M with M^{-1} in Eq. (7) in order to take the positive effect of M as

$$d\mathbf{x} = \frac{\epsilon^2}{2} M^{-1} M^{-\mathsf{T}} \bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x}) dt + \epsilon M^{-1} d\mathbf{w}. \tag{11}$$

Since in this case, we can rewrite Eq. (11) in the same format as the original process, after applying the variable transformation $\mathbf{y} = M\mathbf{x}$.

A general formulation. or theory completeness, we further briefly discuss the possibility to construct preconditioning matrix using the matrix S (Eq. (7)) as an accelerator of the diffusion process. This is motivated by the theories from [25,26,20] that the term $S \bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x})dt$ drives a solenoidal flow that makes the system converge faster to the steady state. According to [14], under the regularity conditions, $|\mathbf{x}(t)|$ usually does not reach the infinity in a finite time, and the convergence of an *autonomous* (the right side of the equation does not contain time explicitly) diffusion process

$$d\mathbf{x} = \frac{\epsilon^2}{2} \bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x})dt + \epsilon d\mathbf{w}$$

can be accelerated by introducing a vector field $C(\mathbf{x}) \in \mathbb{R}^d \to \mathbb{R}^d$

$$d\mathbf{x} = \frac{\epsilon^2}{2} \bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x})dt + C(\mathbf{x})dt + \epsilon d\mathbf{w},$$

where $C(\mathbf{x})$ should satisfy

$$\bigtriangledown_{\mathbf{x}} \cdot (\frac{C(\mathbf{x})}{p^*(\mathbf{x})}) = 0.$$

It is easy to show that $C(\mathbf{x}) = S \bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x})$ satisfies the above condition. However, the diffusion process of existing SGMs is typically *not autonomous*, due to the step size ϵ varies across time designed to guarantee numerical stability. Despite this, we consider it is still worth investigating the effect of S for the sampling process for completeness (see evaluation in Sec. 5). As such, our investigation of preconditioning matrix is expanded from the invertible symmetric matrix in form of MM^T, to more general cases where preconditioning matrices can be written as $MM^\mathsf{T} + S$.

4.3 Instantiation of preconditioned diffusion sampling

We summarize our **preconditioned diffusion sampling** (PDS) method for accelerating the diffusion sampling process in Alg. 1. For generality, we write the original diffusion process as

$$\mathbf{x}_t = \mathbf{h}(\mathbf{x}_{t-1}, t) + \phi(t)\mathbf{z}_t, \tag{12}$$

where $\mathbf{h}(\mathbf{x}_{t-1}, t)$ represents the drift term and $\phi(t)$ the function controlling the scale of the noise $\mathbf{z}_t$. We take the real part whilst dropping the imaginary part generated every step as it can not be utilized by the SGMs. Now we construct the space and frequency preconditioning filter. Given a target dataset image with distribution p^*, its space preconditioning filter A is calculated as

$$A(c, w, h) = \log\left(\mathbb{E}_{\mathbf{x}\sim p^*(\mathbf{x})}\left[\mathbf{x}(c, w, h)\right]\right) + 1\Big), \tag{13}$$

Algorithm 1 Preconditioned diffusion sampling

Input: The frequency R and space A preconditioning operators, the target sampling iterations T;

Diffusion process:
Drawing an initial sample $\mathbf{x}_0 \sim \mathcal{N}(0, I_{C\times H\times W})$
for $t = 1$ **to** T **do**
  Drawing a noise $\mathbf{w}_t \sim \mathcal{N}(0, I_{C\times H\times W})$
  Applying PDS: $\boldsymbol{\eta}_t \leftarrow F^{-1}[F[\mathbf{w}_t \bullet A] \bullet R]$ ▷ $\bullet$ means element-wise division
  Calculating the drift term $\mathbf{d}_t \leftarrow \mathbf{h}(\mathbf{x}_{t-1}, t, \epsilon_t)$
  Applying PDS: $\mathbf{d}_t \leftarrow F^{-1}[F[F^{-1}[F[\mathbf{d}_t] \bullet R] \bullet A^2] \bullet R]$
  Calculating the solenoidal term $S_t \leftarrow S \bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x}_{t-1})$
  Diffusion $\mathbf{x}_t \leftarrow Re[\mathbf{d}_t + S_t + \phi(t)\eta_t]$ ▷ $Re[\cdot]$ means taking the real part
end for

Output: $\mathbf{x}_T$

where $1 \le c \le C, 1 \le w \le W, 1 \le h \le H$ are the channel, width and height dimensions of image. There are two approaches for calculating the filter R. The first approach is to utilize the statistics of the dataset. Specifically, we first define the frequency statistics given a specific image dataset that we are aimed to synthesize as

$$R(c, w, h) = \log\left(\mathbb{E}_{\mathbf{x}\sim p^*(\mathbf{x})}\left[F[\mathbf{x}] \odot \overline{F[\mathbf{x}]}\right](c, w, h) + 1\right) \tag{14}$$

where F is Discrete Fourier Transform, $\odot$ is the element-wise multiplication. In practice, we normalize both space and frequency filter for stability; see supplementary material for more detail. Empirically, 200 images randomly sampled from the dataset is enough for estimating this statistics, therefore this involves marginal extra computation. We observe that this approach works well for accelerating NCSN++ [33], but has less effects on accelerating NCSN [31] and NCSNv2 [32]. The possible reason is that these two models are not sophisticated enough as NCSN++ to utilize the delicate information from the frequency statistics. To address this issue, we propose the second approach which constructs the filter R simply using two parameters λ and r. λ specifies the ratio for shrinking or amplifying the coordinates located out of the circle $\{(h - 0.5H)^2 + (w - 0.5W)^2 \le 2r^2\}$, selected according to the failure behaviour of the vanilla SGM, and The radial range of the filter is controlled by r. See supplementary material for more details. This method works well on accelerating NCSN [31] and NCSNv2 [32].

Remark 2. For the computational complexity of PDS, the major overhead is from FFT and its inverse that only have the complexity of $O(CHW(\log H + \log W))$ [3], which is neglectable compared to the whole diffusion complexity.

5 Experiments

In our experiments, the objective is to show how off-the-shelf SGMs can be accelerated significantly with the assistance of the proposed PDS whilst keeping the image synthesis quality, without model retraining. See supplementary material for the detailed parameter settings and the implementation details.

Datasets. For image synthesis, we use MNIST, CIFAR-10 [19], LSUN (the tower, bedroom and church classes) [37], and FFHQ [18] datasets. Note, for all these datasets, the image height and width are identical, i.e., $H = W$.

Baselines. For evaluating the model agnostic property of our PDS, we test three recent SGMs including NCSN [31], NCSNv2 [32] and NCSN++ [33].

Experiments on MNIST. We use NCSN [31] as the SGM for the simplest digital image generation (28×28). The results are shown in Fig. 3. We observe that when reducing the sampling iterations from 1000 to *20* for acceleration, the original sampling method tends to generate images that lack the digital structure (see the left part of Fig. 3). This suggests us to enlarge a band of frequency part of the diffusion process. Therefore, we set $(r, \lambda) = (0.2H, 1.6)$. It is observed that our PDS can produce digital images with the fine digital structure well preserved under the acceleration rate.

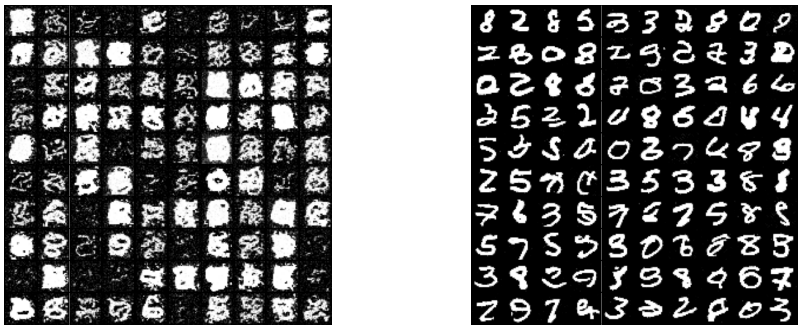

Fig. 3. Sampling using NCSN [31] on MNIST (28×28). **Left**: Results by the original sampling method with 20 sampling iterations. **Right**: Results by our PDS with 20 sampling iterations. More samples in supplementart material.

Experiments on CIFAR-10. Compared to DDPMs, SGMs have much worse performance when the number of sample iterations is relatively small. Our PDS can greatly alleviate this issue as shown in Table. 1, where we evaluate NCSN++ for generating CIFAR-10 (32×32) by FID [11] score. We compare PDS with DDIM [29] and the Analytic-DDIM [1], two representative DDPMs. It is observed that NCSN++ with PDS achieves the best FID scores under different acceleration cases. We apply filter R described by Eq. (14).

Table 1. FID scores of vanilla NCSN++ [33], NCSN++ with PDS, DDIM [29], and Analytic-DDIM [1] under different iterations on CIFAR-10.

T	NCSN++	DDIM	Analytic-DDIM	NCSN++ W/ PDS
100	29.39	6.08	3.55	**3.26**
200	4.35	4.02	3.39	**2.61**

Experiments on LSUN [37]. We first evaluate NCSNv2 [32] to generate church images at a resolution of 96×96 and tower at a resolution of 128×128. For both classes, when accelerated by reducing the iterations from original 3258 to *108* for tower and from original 3152 to *156* for church, we observe that the original sampling method tends to generate images *without sufficient detailed appearance*, similar as the situation on MNIST. Therefore, we also encourage the frequency part of the diffusion process that responsible for the details. The results are displayed in Fig. 4. It is evident that PDS can still generate rich fine details, even when the diffusion process is accelerated up to $20 \sim 30$ times.

Further, we evaluate NCSN++ [33] to generate bedroom and church images at a resolution of 256×256. In this case, we instead observe that the original sampling method tends to generate images *with overwhelming noises* once accelerated (left of Fig. 5). We hence set filter R using Eq. (14) to regulate the frequency part of the diffusion process. As demonstrated in Fig. 5, our PDS is able to prevent the output images from being ruined by heavy noises. All these results suggest the ability of our PDS in regulating the different frequency components in the diffusion process of prior SGMs.

Experiments on FFHQ [18]. We use NCSN++ [33] to generate high-resolution facial images at a resolution of 1024×1024. Similar as on LSUN, we also find out that when accelerated, the original sampling method is vulnerable with heavy noises and fails to produce recognizable human faces. For example, when reducing the iteration from original 2000 to 100, the output images are full of noises and unrecognizable. Similarly, we address this issue with our PDS with filter R. We also apply the space preconditioning to utilize the structural characteristics shared across the whole dataset. It is shown in Fig. 1, PDS can maintain the image synthesis quality using only as less as *66* iterations. In summary, all the above experiments indicate that our method is highly scalable and generalizable across different visual content, SGMs, and acceleration rates.

Evaluation on running speed. Apart from the quality evaluation on image synthesis as above, we further compare the running speed between the vanilla and our PDS using NCSN++ [33]. In this test, we use one NVIDIA RTX 3090 GPU. We track the average wall-clock time of generating a batch of 8 images. As shown in Table 2, our PDS can significantly reduce the running time, particularly for high-resolution image generation on the FFHQ dataset.

Fig. 4. Sampling using NCSNv2 [32] on LSUN (church 96×96 and tower 128×128). **Left**: The original sampling method with 156 iterations for church and 108 iterations for tower. **Right**: PDS sampling method with 156 iterations for church and 108 iterations for tower. More samples in supplementary material.

Fig. 5. Sampling using NCSN++ [33] on LSUN (church and bedroom) (256×256). **Left**: The original sampling method with 166 sampling iterations. **Right**: PDS sampling method with 166 sampling iterations. More examples in supplementary material.

Parameter analysis. We investigate the effect of PDS's two parameters r and λ in mentioned Sec. 4.3. We use NCSN++ [33] with the sampling iterations $T = 166$ on LSUN (bedroom). It is observed in Fig. 6 that there exists a large good-performing range for each parameter. If λ is too high or r is too low, PDS will degrade to the vanilla sampling method, yielding corrupted images; Instead, if λ is too low or r is too high, which means over-suppressing high-frequency signals in this case, pale images with fewer shape details will be generated. For NCSN++ [33], since we directly use the statistics information to construct R, there is no need to worry about selecting r and λ.

Further analysis. In this section, we study the effect of the solenoidal term $S \bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x})$ [4] to the diffusion process. As proved in Thm. 2, as long as S is

[4] For NCSN++ [33], we use $\bigtriangledown_{\mathbf{x}} \log p_t(\mathbf{x})$, where p_t is the distribution function of $\mathbf{x}$ at t, since $\bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x})$ is inaccessible in NCSN++.

Table 2. Evaluating the wall-clock time of generating a batch of 8 images. *SGM:* NCSN++ [33]. *Time unit:* Seconds.

Dataset	LSUN	FFHQ
Vanilla	1173	2030
PDS	**90**	**71**
Speedup times	**13**	**29**

Fig. 6. Parameter analysis. Sampling produced by NCSN++ [33] w/ PDS on LSUN (bedroom) (256 × 256) with 166 sampling iterations. We set (r, λ) to a variety of combination.

skew-symmetric, it will not change the steady-state distribution of the original process. To verify this claim, we generalize the original process as

$$d\mathbf{x} = \frac{\epsilon^2}{2}(M^{-1}M^{-\mathbf{T}} + \omega S) \bigtriangledown_{\mathbf{x}} \log p^*(\mathbf{x})dt + \epsilon M^{-1}d\mathbf{w},$$

where ω is the parameter that controls the scale of S. In Fig. 7, we set $S[\cdot] = Re[F[\cdot] - F^{\mathsf{T}}[\cdot]]$ which is obviously skew-symmetric. We change the scale of ω from 1 to 1000 for evaluating its impact on the output samples. It is observed that ω does not affect the quality of output images. This verifies that S does not change the steady-state distribution of the original diffusion process. Additionally, we perform similar tests with different iterations and other different skew-symmetric operator S. We still observe no obvious acceleration effect from the solenoidal term (see supplementary material).

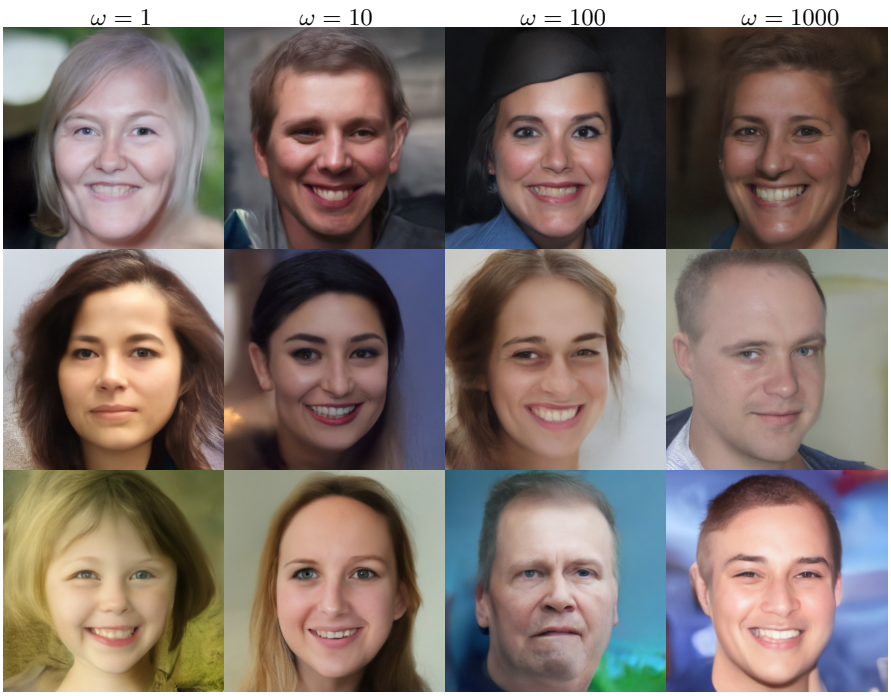

Fig. 7. Samples produced by NCSN++ [33] w/ PDS on FFHQ (1024x1024) with different solenoidal terms. Sampling iteration: 66. More samples in supplementary material.

6 Conclusion

In this work, we have proposed a novel preconditioned diffusion sampling (PDS) method for accelerating off-the-shelf score-based generative models (SGMs), without model retraining. Considering the diffusion process as a Metropolis adjusted Langevin algorithm, we reveal that existing sampling suffers from ill-conditioned curvature. To solve this, we reformulate the diffusion process with matrix preconditioning whilst preserving its steady-state distribution (the target distribution), leading to our PDS solution. Experimentally, we show that PDS significantly accelerates existing state-of-the-art SGMs while maintaining the generation quality.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (Grant No. 6210020439), Lingang Laboratory (Grant No. LG-QS-202202-07), Natural Science Foundation of Shanghai (Grant No. 22ZR1407500), Shanghai Municipal Science and Technology Major Project (Grant No. 2018SHZDZX01 and 2021SHZDZX0103), Science and Technology Innovation 2030 - Brain Science and Brain-Inspired Intelligence Project (Grant No. 2021ZD0200204).

References

1. Bao, F., Li, C., Zhu, J., Zhang, B.: Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. ICLR (2022) 4, 10, 11
2. Bovik, A.C.: The Essential Guide to Image Processing (2009) 3
3. Brigham, E.O.: The fast Fourier transform and its applications (1988) 4, 7, 9
4. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: ICLR (2019) 2
5. De Bortoli, V., Thornton, J., Heng, J., Doucet, A.: Diffusion schrödinger bridge with applications to score-based generative modeling. In: NeurIPS (2021) 1, 4
6. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. In: NeurIPS (2021) 4
7. Dockhorn, T., Vahdat, A., Kreis, K.: Score-based generative modeling with critically-damped langevin diffusion. In: ICLR (2022) 4
8. Gardiner, C.W., et al.: Handbook of stochastic methods (1985) 3
9. Girolami, M., Calderhead, B.: Riemann manifold langevin and hamiltonian monte carlo methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology) (2011) 3
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014) 1, 5
11. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017) 10
12. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020) 4
13. Ho, J., Saharia, C., Chan, W., Fleet, D.J., Norouzi, M., Salimans, T.: Cascaded diffusion models for high fidelity image generation. arXiv preprint (2021) 4
14. Hwang, C.R., Hwang-Ma, S.Y., Sheu, S.J.: Accelerating diffusions. The Annals of Applied Probability (2005) 8
15. Hyvärinen, A., Dayan, P.: Estimation of non-normalized statistical models by score matching. JMLR (2005) 5
16. Jolicoeur-Martineau, A., Li, K., Piché-Taillefer, R., Kachman, T., Mitliagkas, I.: Gotta go fast when generating data with score-based models. arXiv preprint arXiv (2021) 4
17. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: ICLR (2018) 2
18. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR (2019) 2, 10, 11
19. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009) 10
20. Lelievre, T., Nier, F., Pavliotis, G.A.: Optimal non-reversible linear drift for the convergence to equilibrium of a diffusion. Journal of Statistical Physics (2013) 8
21. Li, C., Chen, C., Carlson, D., Carin, L.: Preconditioned stochastic gradient langevin dynamics for deep neural networks. In: AAAI (2016) 3
22. Neal, R.M., et al.: Mcmc using hamiltonian dynamics. Handbook of markov chain monte carlo (2011) 4
23. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint (2021) 2, 4

24. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: ICML (2021) 4
25. Ottobre, M.: Markov chain monte carlo and irreversibility. Reports on Mathematical Physics (2016) 8
26. Rey-Bellet, L., Spiliopoulos, K.: Irreversible langevin samplers and variance reduction: a large deviations approach. Nonlinearity (2015) 8
27. Roberts, G.O., Stramer, O.: Langevin diffusions and metropolis-hastings algorithms. Methodology and computing in applied probability (2002) 3, 6
28. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: ICML (2015) 4
29. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: ICLR (2020) 4, 10, 11
30. Song, Y., Durkan, C., Murray, I., Ermon, S.: Maximum likelihood training of score-based diffusion models. In: NeurIPS (2021) 1
31. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. In: NeurIPS (2019) 1, 4, 5, 9, 10
32. Song, Y., Ermon, S.: Improved techniques for training score-based generative models. In: NeurIPS (2020) 1, 4, 5, 9, 10, 11, 12
33. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: ICLR (2021) 1, 2, 4, 5, 9, 10, 11, 12, 13, 14
34. Vahdat, A., Kreis, K., Kautz, J.: Score-based generative modeling in latent space. In: NeurIPS (2021) 4
35. Welling, M., Teh, Y.W.: Bayesian learning via stochastic gradient langevin dynamics. In: ICML (2011) 3
36. Xiao, Z., Kreis, K., Vahdat, A.: Tackling the generative learning trilemma with denoising diffusion gans. In: ICLR (2022) 1, 2
37. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint (2015) 10, 11