Fast Sampling of Diffusion Models via Operator Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Diffusion models have found widespread adoption in various areas. However, sampling from them is still slow because it involves emulating a reverse stochastic process with hundreds-to-thousands of neural network evaluations. Inspired by the recent success of neural operators in accelerating differential equations solving, we approach this problem by solving the underlying neural differential equation from an operator learning perspective. We examine probability flow ODE trajectories in diffusion model and observe a compact energy spectrum that can be learned efficiently in Fourier space. With this insight, we propose diffusion Fourier neural operator (DFNO) with temporal convolution in Fourier space to parameterize the operator that maps initial condition to the solution trajectory. DFNO can apply to any diffusion models and generate high-quality samples in one step. Our method achieves the state-of-the-art clean FID of 5.9 (legacy FID 4.72) on CIFAR-10 using one network evaluation.

1 Introduction

Diffusion models, also known as score-based generative models, have emerged as a powerful generative modeling framework in various areas. They have achieved state-of-the-art (SOTA) performance in many applications including image generation [1, 2, 3, 4], molecule generation [5], audio synthesis [6, 7] and model robustness [8, 9]. However, sampling from diffusion models is still slower than other generative models such as generative adversarial networks (GAN) [10] by several orders of magnitude. Accelerating the sampling process of diffusion models remains challenging but important in applying diffusion models in many downstream tasks. Many studies have worked on the fast sampling of diffusion models which can be summarized into two categories.

Training-free sampling methods focus on solving the reverse stochastic differential equation (SDE) or the corresponding probability flow ordinary differential equation (ODE) from a numerical perspective, which can be applied to any trained score model without extra training. SDE-based samplers often have better sample quality than ODE-based samplers but they are much slower and require hundreds if not thousands of function evaluations [1, 11]. ODE-based methods are fully deterministic and allow for larger steps in discretization. Existing studies work on reducing the approximation error with less steps [1, 11, 12, 13, 14, 15, 16] but still need more than 10 function evaluations to generate high-quality samples.

Training-based sampling methods require extra training including knowledge distillation [17, 18] and learning the noise schedule [19, 20]. Training-based methods work in the few-step regime with less than 10 steps. The current SOTA progressive distillation [18] reduces the number of steps down to 4-8 without losing much sample quality. However, it requires progressive training from fine resolution to coarse resolution. It also breaks in the limit of one function evaluation. DDGAN [21] achieves

Under review at the NeurIPS 2022 Workshop on Score-Based Methods. Do not distribute.

similar results as progressive distillation by leveraging conditional GAN to model the denoising distributions or equivalently the reverse stochastic process, allowing for large denoising steps. LSGM [2] accelerates sampling by encoding the data distribution into a smooth latent distribution that is close to a Gaussian prior and obtains better image quality with 20 to 100 steps.

Our contributions. Inspired by the recent success of neural operators [22, 23, 24] in solving differential equations, we propose to solve the probability flow ODE of diffusion models from an operator learning perspective. We examine the characteristics of the ODE trajectories sampled from trained diffusion models [11, 25] and observe a compact energy spectrum. With this observation, we propose a diffusion Fourier neural operator (DFNO) with temporal convolution in the Fourier space to obtain probability flow trajectories efficiently.

- DFNO only takes one function evaluation to sample and has better generalization ability than distillation methods. With the trajectory information guiding the sampling, DFNO achieves the state-of-the-art FID of 5.9 for CIFAR-10 in the one-function-evaluation setting.
- Temporal convolution blocks in the Fourier space can learn a trajectory as a function of time in the Fourier space efficiently. DFNO inherits the discretization invariant property from the Fourier neural operator [23] over the temporal dimension. One can train DFNO with high-resolution in time for stronger supervision and sample at low-resolution for fast inference.
- Compared to the current SOTA progressive distillation [18], DFNO is easier to train and not limited to specific time step scheme. This allows us to learn from a large class of ODE-based samplers including training-based methods.

2 Background

We consider the general class of score-based generative models in a unified continuous-time framework proposed by [11], which includes different variants of diffusion models [26, 25]. We will use score-based models interchangeably with the diffusion models. Suppose the data distribution is p_{data}. The forward pass is a diffusion process $\{\mathbf{x}(t)\}$ starting from 0 to T can be expressed as

$$\mathrm{d}\mathbf{x} = f(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}, \tag{1}$$

where $\mathbf{w}$ is the standard Wiener process, and $f(\mathbf{x}, t) : \mathbb{R}^d \to \mathbb{R}^d$ and $g(t) : \mathbb{R} \to \mathbb{R}$ are the drift and diffusion coefficients respectively. Diffusion models choose f, g such that $\mathbf{x}(0) \sim p_{\mathrm{data}}$ and $\mathbf{x}(T) \sim \mathcal{N}(0, \mathbf{I})$. Song et al. [11] show that the following probability flow ODE produces the same marginal distributions $p_t(\mathbf{x})$:

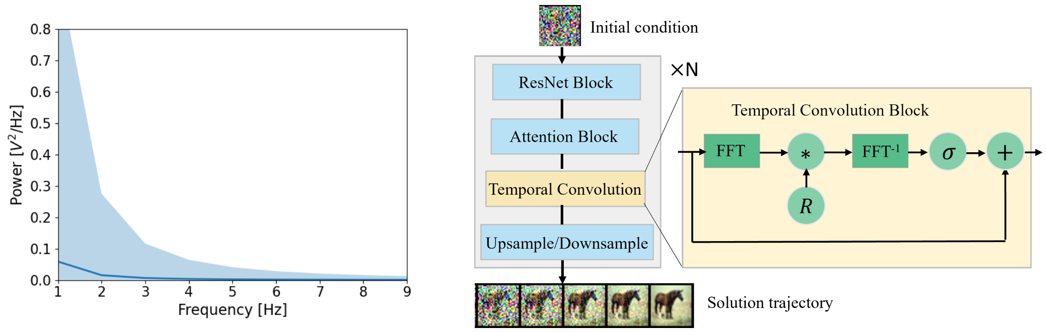
$$\mathrm{d}\mathbf{x} = f(\mathbf{x}, t)\mathrm{d}t - \frac{1}{2}g(t)^2\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\mathrm{d}t. \tag{2}$$

The sampling process eventually becomes solving the probability flow ODE 2 from T to 0 given the initial condition $\mathbf{x}(T)$. Furthermore, $f(\mathbf{x}, t)$ often has the affine form $f(\mathbf{x}, t) = f(t)\mathbf{x}$, where $f : \mathbb{R} \to \mathbb{R}$. We can simplified the ODE 2 into a semi-linear ODE. Integrating both sides over time gives the explicit form of solution for any $t < s$:

$$\mathbf{x}(t) = \phi(t, s)\mathbf{x}(s) - \int_s^t \phi(t, \tau)\frac{g(\tau)^2}{2}\nabla_{\mathbf{x}} \log p_\tau(\mathbf{x})\mathrm{d}\tau, \tag{3}$$

where $\phi(t, s) = \exp\left(\int_s^t f(\tau)\mathrm{d}\tau\right)$. The ODE is often solved using numerical techniques such as Euler discretization [27] or multistep methods [14]. The score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ is usually parameterized by $\hat{\epsilon}_\theta(\mathbf{x}_t) \approx -\sigma_t \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$, where σ_t is the noise schedule [11, 25].

Neural operators [23, 22] are designed to solve the differential equations fast by learning a parametric map between two Banach spaces. They are constructed as a stack of kernel integration layers where the kernel function is parameterized by learnable weights. More details are in appendix A.2.



(a) Power spectrum of the ODE trajectories sampled from "DDPM++ cont. (VP)" model trained by [11] on CIFAR10. The mean is computed over all pixel locations and channels of randomly generated trajectories. Most power concentrates in the ≤ 5 Hz regime. The shaded region represents maximum and minimum power.

(b) Architecture of DFNO. Blue blocks are commonly used modules in diffusion models. The temporal convolution block first approximates the Fourier transform over temporal dimension via fast Fourier transform. It then multiply it by the filter $R \in \mathbb{C}^{k \times d_{\text{out}} \times d_{\text{in}}}$ and do inverse Fourier transform. The shortcut connection restores the high-frequency information without extra cost. Temporal convolution blocks are introduced after every attention block.

Figure 1: Compact spectrum and DFNO architecture design.

3 Learning the trajectory as a function of time

3.1 Problem statement

Given any initial condition $\mathbf{x}(T) \sim \mathcal{N}(0, \mathbf{I})$, our goal is to learn a neural operator that predicts the probability flow trajectory $\{\mathbf{x}(t)\}_s^0$ with time flowing from s to 0 defined in equation 3, where the end point $\mathbf{x}(0)$ is the data. Suppose the time domain $D = [0, s]$, $s > 0$. Let $\mathcal{A}$ be the finite-dimensional space of initial condition, $\mathcal{U} = \mathcal{U}(D; \mathbb{R}^d)$ denote the space of the target continuous time functions with output value in $\mathbb{R}^d$. From the exact solution $\mathbf{x}(t)$ in equation 3, we know the unique solution operator $G^* : \mathcal{A} \to \mathcal{U}$ exists and is a weighted integral operator of the score function. We build a neural operator G_θ parameterized by θ to approximate the solution operator G^* by minimizing:

$$\min_{\theta} \mathbb{E}_{\mathbf{x}_T \sim \mathcal{N}(0,\mathbf{I})} \mathcal{L}\left(G_\theta\left(\mathbf{x}_T\right) - G^*\left(\mathbf{x}_T\right)\right), \tag{4}$$

where $\mathcal{L} : \mathcal{U} \to \mathbb{R}_+$ is some loss functional such as L^p-norm for some $p \geq 1$.

3.2 Compact power spectrum

Learning the solution operator G^* is a challenging task in general. However, we observe that the trajectory of probability flow ODEs has a compact energy spectrum over the time dimension and thus can be learned efficiently in the Fourier space. Figure 1a visualizes the energy spectrum of ODE trajectories sampled from the diffusion model "DDPM++ cont. (VP)" trained by [11] on CIFAR10. Appendix A.1 explains the details of the power spectrum.

3.3 Temporal convolution block in Fourier space

Based on the special characteristic of the ODE trajectory and the integral expression of the exact ODE solution in equation 3, we build our temporal convolution block with Fourier integral operator $\mathcal{K}$ to efficiently model the trajectory. Given an integrable function $u : D \to \mathbb{R}^{d_{\text{in}}}$, the Fourier transform operator $\mathcal{F}$ is defined as

$$\left(\mathcal{F}u\right)_j(k) = \int_D u_j(t) \exp\left(-2\pi i k t\right) \mathrm{d}t, \tag{5}$$

for $j = 1, \ldots, d_{\text{in}}$, where i is the imaginary unit, k is the frequency. The Fourier integral operator $\mathcal{K}_\phi$ parameterized by ϕ is defined as

$$\left(\mathcal{K}_\phi u\right)(t) = \mathcal{F}^{-1}\left(R_\phi \cdot \left(\mathcal{F}u\right)\right)(t) = \int_D (\mathcal{F}^{-1} R_\phi)(t) u(t) \mathrm{d}t, \tag{6}$$

where $R_\phi \in \ell^2(\mathbb{Z}; \mathbb{C}^{d_{\text{out}} \times d_{\text{in}}})$ is the Fourier transform of a kernel function parameterized by ϕ that we learn from data, and the second equality holds by the convolution theorem. Given an input function u, the temporal convolution layer $\mathcal{P}$ is defined as

$$(\mathcal{P}u)(t) = u(t) + \sigma\left(\left(\mathcal{K}_\phi u\right)(t)\right), \tag{7}$$

where σ is a point-wise activation function. Figure 1b demonstrates the implementation details of the temporal convolution block.

3.4 Architecture of DFNO

As demonstrated in Figure 1b, the overall architecture of DFNO is similar to the UNet structure that is commonly used in diffusion models. We introduce temporal convolution after every attention block. Suppose the time resolution is M. The input noise is repeated M times in the first block and each copy will be mixed with the corresponding time embedding in the ResNet block. More details are provided in the appendix A.3.

4 Experiments

We use the Frechet inception distance (FID) [28] to evaluate the quality of generated images. FID is computed between 50,000 generated images and CIFAR10 train set with the clean-fid library [29]. We use the checkpoint of "DDPM++ cont. (VP)" model by [11] trained on CIFAR10 [30]. 1 million trajectories are sampled following the corresponding probability flow ODE of the variance preseving (VP) SDE to train DFNO. The We use L^1-norm as the loss functional. Table 1 compares our results against the recent training-free and training-based methods. Our method is clearly the best in the one-function-evaluation setting, even better than most training-free methods with 10 steps.

Method	Model evaluations	FID score
DFNO (ours)	1	**5.92**
Knowledge Distillation [17]	1	9.36
Knowledge Distillation (our architecture)	1	8.06
Progressive Distillation [18]	1	9.12
	2	4.51
	4	3.00
GGDM + PRED + TIME [20]	5	13.77
	10	8.23
DDIM [27]	10	13.36
	20	6.84
DPM-solver-2 [15]	12	5.28
DPM-solver-3 [15]	12	6.03
3-DEIS [14]	5	16.09
	10	4.17

Table 1: Comparison of fast sampling methods on CIFAR-10 for diffusion models in the literature.

Ablation study The results of our ablation study are reported in the Table 2. We observe that 4-step is much better than 1-step. Quadratic time step scheme and $1/\sqrt{\sigma_t}$ loss weighting also improve the sample quality. The setting of each ablation is explained in appendix A.4.

5 Conclusion

In this paper, we propose diffusion Fourier neural operator (DFNO), a training-based fast sampling method for diffusion models. DFNO leverages the compact power spectrum characteristic of the probability flow ODE trajectory and models it efficiently with Fourier integral operator. Experiments show that our method archieves the best performance in one-function-evalution setting.

References

[1] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.

[2] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

[3] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and BjÃ¶rn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[5] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

[6] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *ICLR*, 2021.

[7] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *International Conference on Machine Learning*, pages 8599–8608. PMLR, 2021.

[8] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning (ICML)*, 2022.

[9] Nicholas Carlini, Florian Tramer, J Zico Kolter, et al. (certified!!) adversarial robustness for free! *arXiv preprint arXiv:2206.10550*, 2022.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[11] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[12] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[13] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2021.

[14] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.

[15] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022.

[16] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

[17] Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.

[18] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2021.

[19] Max WY Lam, Jun Wang, Rongjie Huang, Dan Su, and Dong Yu. Bilateral denoising diffusion models. *arXiv preprint arXiv:2108.11514*, 2021.

[20] Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. In *International Conference on Learning Representations*, 2021.

[21] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. In *International Conference on Learning Representations*, 2021.

[22] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.

[23] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[24] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces. *arXiv preprint arXiv:2108.08481*, 2021.

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[26] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[27] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[28] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[29] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *CVPR*, 2022.

[30] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

A Appendix

A.1 Energy spectrum

The discrete-time Fourier transform of the signal $x(t)$ with period T is given by

$$X_j = \sum_{i=1}^{N} x(t_i) \exp\left(-\frac{2\pi}{T} j i t_i\right), \tag{8}$$

where $t_i = \frac{iT}{N}$. $\frac{j}{T}$ is the frequency. j is called the frequency mode. Let $\Delta = \frac{1}{N}$ be the time step. The spectrum is defined as the product of Fourier transform of x with its conjugate:

$$S_j = \frac{2\Delta^2}{T} X_j X_j^*, \tag{9}$$

where X_j^* is the complex conjugate. In practice, the statistics are computed over all pixel locations and channels of randomly generated trajectories. $T = 1$ and the sampling frequency is 1000 Hz to avoid aliasing. We observe that most power concentrates in the regime where the frequency mode is less than 5.

A.2 Background: neural operators

Let $\mathcal{A}$ and $\mathcal{U}$ be two Banach spaces and $G : \mathcal{A} \to \mathcal{U}$ be a non-linear map. Suppose we have a finite collection of data $\{a_i, u_i\}_{i=1}^N$ where $a_i \sim \mu$ are i.i.d. samples from the distribution μ supported on $\mathcal{A}$ and $u_i = G(a_i)$. Neural operators aim to learn G_ϕ parameterized by ϕ to approximate G from the observed data by minimizing the empirical risk given by

$$\min_\phi \mathbb{E}_{a\sim\mu} \|G(a) - G_\phi(a)\|_{\mathcal{U}} \approx \min_\phi \frac{1}{N} \sum_{i=1}^{N} \|u_i - G_\phi(a_i)\|_{\mathcal{U}}. \tag{10}$$

The architecture of neural operators is constructed as a stack of kernel integration layers where the kernel function is parameterized by learnable weights.

A.3 Architecture detail

The overall architecture is similar to the UNet structure used in diffusion models. On top of that, we introduce temporal convolution after each attention layer and replace all the Conv2d layers with Conv3d. So the intermediate feature map will have an additional time dimension compared to standard UNet. Suppose the output time resolution is M. The input noise has a shape of (B, C, H, W) where B is the batch size, C is the number of channels, H, W are the height and width. In the first convolution block, the input noise will be repeated M times as the initial trajectory. We add the time embeddings of M steps in each ResNet block.

A.4 Ablation detail

For all results in the same column, we keep all the other settings the same except for the control variable. The left column reports the ablation on the temporal resolution with default setting of uniform time steps and $1/\sqrt{\sigma_t}$ loss weighting. The middle column compares uniform and quadratic time step scheme with the default setting of resolution 4 and $1/\sqrt{\sigma_t}$ loss weighting. The right column compares different loss weightings with the default setting of resolution 4 and uniform time steps.

Resolution	FID score	Time step scheme	FID score	Loss weighting	FID score
1	7.74	uniform	6.21	uniform weights	7.8
4	6.21	quadratic	5.92	$1/\sqrt{\sigma_t}$	7.2
6	6.17				

Table 2: Left: ablation on the temporal resolution; Middle: ablation on the time step scheme; Right: ablation on the loss weighting. The numbers are clean FID score [29].

A.5 Model complexity and inference Time

"DDPM++cont. (VP)" architecture has 106,632,579 parameters. The DFNO with resolution 4 has 114,562,435 parameters. The computation cost of one model evaluation of DFNO is 2 times as that of the standard score model when the batchsize is 2 on 16G-V100.