FoLD: Efficient Fourier-series-based Score Estimation for Langevin Diffusion

[1]Siddarth Asokan, [2]Aadithya Srikanth, [3]Nishanth Shetty, [4]Chandra Sekhar Seelamantula
[1]Microsoft Research Lab, #9, "Vigyan", Lavelle Road, Bengaluru, India
[2,3,4]Department of Electrical Engineering, Indian Institute of Science, Bengaluru, India
[1]sasokan@microsoft.com, [2]srikanth.aadithya@gmail.com, {[3]nishanths,[4]css}@iisc.ac.in

Abstract

Score-based models comprise a neural network trained to approximate the Stein score, which is the gradient of the logarithm of the target distribution. Novel samples are obtained by Langevin diffusion over the approximated score field. In this extended abstract, we explore an efficient, low-resource, plug-and-play framework to computing the score of the distribution based on a Fourier-series approximation. We derive closed-form approaches to estimating the score based on the Fourier coefficients of the data distribution using novel Fourier score networks (FSN) with predetermined weights. The proposed FSN can be incorporated into various existing Langevin sampling schemes. We demonstrate sampling with noise-conditioned score-based samplers, wherein the proposed approaches result in a $20\times$ *faster network weight computation, over the baselines on synthetic Gaussian data, and a two-fold reduction in iterations when applied to the latent-space representations of image datasets.*

1. Introduction

Recent advancements in generative models, particularly diffusion and score-based models [7, 23–25, 27], have enabled the transformation of high-dimensional noise into realistic images. These models function by gradually introducing noise into data samples ($\boldsymbol{x} \in \mathbb{R}^n$) from a standard Gaussian distribution ($\boldsymbol{z} \sim p_{\boldsymbol{z}} = \mathcal{N}(\mathbf{0}, \mathbb{I})$), and then reversing this process to generate images. The forward process is described as a stochastic differential equation (SDE) combining data-dependent drift and noise-based diffusion. The reverse process, where noise is transformed back into images, is approximated using neural networks, yielding a generative model. This approaches discretize the reverse SDE [10, 11, 20, 30, 31], implemented via Langevin Monte Carlo, and has achieved state-of-the-art performance in image generation and reconstruction [4, 8, 12–14].

Score-based Generative Models: Central to score-based models, as identified by Song *et al.* [24, 26, 27], is the Stein score function ($\nabla_{\boldsymbol{x}} \ln (p_d(\boldsymbol{x}))$), representing the gradient of the logarithm of the target data distribution. These models typically train a neural network ($s_\theta(\boldsymbol{x})$) to approximate this score by minimizing a score-matching loss. This loss is equivalent to minimizing the sum of the norm of the score network output and the trace of its Jacobian. Simplifications include Sliced score matching (SSM), which uses random Gaussian projections, and Denoising score matching, focusing on scores of noise-perturbed data densities.

Fourier Bases and Neural Networks: Fourier Bases and Neural Networks: Multi-layer networks with sinusoidal activations (SIREN) have shown performance improvements when learning representations of images, sound, and wavefields [22]. Fourier Neural Operators (FNO), as proposed by Li *et al.,* [18] parameterize integral kernels in Fourier space and have been applied to approximate various partial differential equations (PDEs), like the Burgers', Navier-Stokes, or Darcy flow equations, with applications in modeling turbulent flows and zero-shot super-resolution [3, 17, 28, 32]. Asokan and Seelamantula [1] utilized a Fourier-series approximation in GANs to solve a Poisson PDE.

1.1. Our Contribution

In this extended abstract, we introduce a novel method for estimating the Stein score through a Fourier-series approximation, specifically in the context of Langevin diffusion. Building upon the Fourier approximation method discussed in [1], we develop the Fourier Score Network (FSN), which is designed to replace traditional residual and convolutional score networks [24, 25, 27], offering a more compute and memory efficient alternative. Our approach, entitled **Fo**urier-series-based **L**angevin **D**iffusion (**FoLD**), leverages the FSN to estimate both the true score of the target distribution and its noise-conditioned variants. This facilitates a *plug-and-play* integration with existing sampling algorithms [24, 25].

Experimental validation on synthetic Gaussian data

[1]Work done while at the Department of Electrical Engineering, Indian Institute of Science, Bengaluru, India.

Siddarth Asokan and Nishanth Shetty are supported by the Qualcomm Innovation Fellowship 2023. Nishanth Shetty is supported by the Prime Minister's Research Fellowship.

demonstrates that the FoLD approach achieves a two-fold increase in sampling over baseline methods in modeling the latent-space representations of images while eliminating the need to train the score network. The computation of the FSN parameters is about 20× faster than training NCSNs.

Our work is distinct from the closest related research by Zheng *et al.* [32] and Lim *et al.* [19], wherein the authors consider neural operators to transform Gaussian distributions into continuous-time solution trajectories of the reverse diffusion process, relying on temporal convolution layers parameterized in the Fourier space.

2. Score-based Langevin Diffusion

We first recall the score-based Langevin diffusion formulation. Langevin Monte Carlo (LMC) has been the *de facto* strategy for sampling from a given distribution, where we have access to the true *score function*, $\nabla_{\boldsymbol{x}} \ln\left(p_d(\boldsymbol{x})\right)$. LMC is an instance of Markov chain Monte Carlo (MCMC), and is a discretization of the Langevin diffusion SDE given by $\mathrm{d}\boldsymbol{x}(t) = -\nabla_{\boldsymbol{x}} f(\boldsymbol{x}(t))\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}\boldsymbol{B}(t)$, where $\boldsymbol{B}(t)$ denotes Brownian motion. The associated discretized update is:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma \nabla_{\boldsymbol{x}} f(\boldsymbol{x})\big|_{\boldsymbol{x}=\boldsymbol{x}_t} + \sqrt{2\gamma}\,\boldsymbol{z}_t, \tag{1}$$

where $\boldsymbol{z}_t \sim p_{\boldsymbol{z}} = \mathcal{N}(\mathbf{0}, \mathbb{I}_n)$ is an instance of standard Gaussian noise drawn at time instant t, and is independent of the sample $\boldsymbol{x}_t$. The evolution is initialized with parametric noise, *i.e.*, $\boldsymbol{x}_0 \sim p_{\boldsymbol{z}}$. Different choices of the mapping function f give rise to various choice of diffusion models. In this paper, we consider score-based approaches, where $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = -\nabla_{\boldsymbol{x}} \ln(p_d(\boldsymbol{x}))$ [24]. In practice, γ is annealed as the iterations progress, typically on a linear or a geometric scale, giving rise to the annealed Langevin sampler:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \gamma_t \nabla_{\boldsymbol{x}} \ln\left(p_d(\boldsymbol{x})\right)\big|_{\boldsymbol{x}=\boldsymbol{x}_t} + \sqrt{2\gamma_t}\,\boldsymbol{z}_t, \tag{2}$$

Existing score-based approaches train a neural network $s_\theta(\boldsymbol{x})$ to approximate the score, by means of a score-matching loss, originally derived in the context of independent component analysis [9]:

$$\mathcal{L}^{\mathrm{SM}}(\theta) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_d}\left[\mathrm{Tr}\left(\nabla_{\boldsymbol{x}} s_\theta(\boldsymbol{x})\right) + \frac{1}{2}\|s_\theta(\boldsymbol{x})\|_2^2\right], \tag{3}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace operator, and $\nabla_{\boldsymbol{x}} s_\theta(\boldsymbol{x})$ is the Jacobian of the score network. The output of the trained network is used to generate samples through annealed Langevin dynamics. To circumvent computing the Jacobian, Song et al. [26] proposed a sliced score-matching loss ($\mathcal{L}^{\mathrm{SSM}}$), wherein the trace of the Jacobian is replaced with a stochastic estimate, computed over random Gaussian projections, *i.e.,* $\mathrm{Tr}\left(\nabla_{\boldsymbol{x}} s_\theta(\boldsymbol{x})\right) \approx \mathbb{E}_{\boldsymbol{v}\sim p_{\boldsymbol{z}}}\left[\boldsymbol{v}^{\mathrm{T}} \nabla_{\boldsymbol{x}} s_\theta(\boldsymbol{x}) \boldsymbol{v}\right]$. However, as noted by Song et al. [24], a major limitation is that the predicted score is *weak* in regions far away from the target distribution, which is typically the case at the start of the Markov chain. To circumvent this issue, the noise-conditional score network (NCSN) was proposed in [24], wherein the denoising score-matching (DSM) loss, defined as $\mathcal{L}^{\mathrm{DSM}}(\theta;\sigma) = \frac{1}{2}\,\mathbb{E}_{\boldsymbol{x}\sim p_d}\,\mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbb{I})}\left[\left\|s_\theta(\tilde{\boldsymbol{x}},\sigma) + \frac{\tilde{\boldsymbol{x}}-\boldsymbol{x}}{\sigma^2}\right\|_2^2\right]$ is computed over a set of predetermined noise levels/scales $\{\sigma_\ell\}_{\ell=1}^{L}$, giving rise to the NCSN loss: $\mathcal{L}^{\mathrm{NCSN}}(\theta) = \frac{1}{L}\sum_{\ell=1}^{L} \lambda_{\sigma_\ell} \mathcal{L}^{\mathrm{DSM}}(\theta;\sigma_\ell)$, where λ_{σ_ℓ} is a function of σ_ℓ. Subsequent works, such as those by Jolicoeur-Martineau *et al.,* [10], and Karras *et al.,* [11] considered improved discretizations of the underlying differential equation to accelerate the sampling process. Recently, denoising diffusion GANs (DDGANs) [29] have been introduced, wherein a GAN is trained to model the diffusion process, with the generator and discriminator networks conditioned on the time index, while in [2, 6, 15], GAN discriminators have been used for classifier-guided Langevin sampling.

3. Fourier-series-based Score Estimation

We first recall the Fourier-series representation proposed for the GAN discriminator in WGAN-FS by Asokan and Seelamantula [1]. Consider the data distribution $p_d(\boldsymbol{x}),\ \boldsymbol{x} \in \mathbb{R}^n$, with the Fourier approximation given by:

$$p_d(\boldsymbol{x}) \approx \frac{1}{2} + \sum_{\boldsymbol{m}\in[M]^n} \alpha^r_{\boldsymbol{m}} \cos(\omega_o\langle\boldsymbol{m},\boldsymbol{x}\rangle) + \alpha^i_{\boldsymbol{m}} \sin(\omega_o\langle\boldsymbol{m},\boldsymbol{x}\rangle), \tag{4}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product, $\boldsymbol{m} \in \mathbb{Z}^n$ is an n-dimensional vector with integer entries, $[M]^n$ denotes the Cartesian product space $\{1,\ldots,M\}^n$, and $\boldsymbol{\omega}_{\boldsymbol{m}} = \omega_o\boldsymbol{m} = \omega_o[m_1, m_2, \ldots, m_n]^{\mathrm{T}}$ denotes the harmonics, where in turn, $\omega_o = \frac{2\pi}{T}$ is the fundamental frequency, T being the fundamental period. The Fourier coefficients $\alpha_{\boldsymbol{m}}$ are expressed via approximations of the characteristic function as

$$\alpha^r_{\boldsymbol{m}} = \mathrm{Re}\{\alpha_{\boldsymbol{m}}\} \approx \frac{1}{NT} \sum_{\substack{k=1\\ \boldsymbol{x}_k\sim p_d}}^{N} \cos(\omega_o\langle\boldsymbol{m},\boldsymbol{x}_k\rangle),\ \text{and} \tag{5}$$

$$\alpha^i_{\boldsymbol{m}} = \mathrm{Im}\{\alpha_{\boldsymbol{m}}\} \approx \frac{1}{NT} \sum_{\substack{k=1\\ \boldsymbol{x}_k\sim p_d}}^{N} \sin(\omega_o\langle\boldsymbol{m},\boldsymbol{x}_k\rangle). \tag{6}$$

In this work, we extend the above framework to score estimation. We consider two approaches — one corresponding to the standard score estimate; and the other corresponding to the denoising NCSN approach. Starting from the Fourier approximation of $p_d(\boldsymbol{x})$ in Equation (4), and leveraging the property that $\nabla_{\boldsymbol{x}} \ln\left(p_d(\boldsymbol{x})\right) = \frac{1}{p_d(\boldsymbol{x})}\,\nabla_{\boldsymbol{x}} p_d(\boldsymbol{x})$, we obtain

$$\begin{aligned}\nabla_{\boldsymbol{x}} \ln\left(p_d(\boldsymbol{x})\right) \approx &\left(\frac{\omega_o}{p_d(\boldsymbol{x})}\right) \sum_{\boldsymbol{m}\in[M]^n} \alpha^i_{\boldsymbol{m}} \cos(\omega_o\langle\boldsymbol{m},\boldsymbol{x}\rangle)\boldsymbol{m} \\ &- \left(\frac{\omega_o}{p_d(\boldsymbol{x})}\right) \sum_{\boldsymbol{m}\in[M]^n} \alpha^r_{\boldsymbol{m}} \sin(\omega_o\langle\boldsymbol{m},\boldsymbol{x}\rangle)\boldsymbol{m},\end{aligned} \tag{7}$$

where $p_d(\boldsymbol{x})$, $\alpha^r_{\boldsymbol{m}}$, and $\alpha^i_{\boldsymbol{m}}$ are computed as per Equations (4)–(6). The score of the data distribution can be computed in closed-form using a neural network model, but with predetermined weights – we refer to the network as the Fourier score network $s_{\text{FSN}}(\boldsymbol{x})$. Score-based Langevin sampling can be carried out *without the need for training* a score network $s_\theta(\boldsymbol{x})$. The $s_{\text{FSN}}(\boldsymbol{x})$ parameters $\alpha_{\boldsymbol{m}}$ are computed one-shot over training samples, obviating the need for backpropagation-based optimization.

However, as with the baseline NCSN, the magnitude of the score is small far away from the target p_d, which significantly slows down the sampling process (*mixing time* of the Markov chain). To circumvent this issue, we also consider noise-conditional counterparts of FoLD (NC-FoLD) in modeling the smoothed score, $\nabla_{\boldsymbol{x}} \ln\big(\big(p_d * \mathcal{N}(\mathbf{0}, \sigma_\ell^2 \mathbb{I})\big)(\boldsymbol{x})\big)$.

An alternative to computing the score is to estimate the Fourier representation of $p_d(\boldsymbol{x})$ or $(p_d * \mathcal{N}(\mathbf{0}, \sigma_\ell^2\mathbb{I}))(\boldsymbol{x})$, and subsequently compute the score via automatic differentiation (`autodiff`). To facilitate a fair comparison between the two approaches, we implemented the FSN as a custom-layer neural network, with both p_d and $\nabla_{\boldsymbol{x}} \ln(p_d)$ as the outputs.

The FoLD Algorithms: Algorithm 1 presents *sampling* and *coefficient estimation* in FoLD and NC-FoLD. In FoLD, the Fourier coefficients $\{\alpha_{\boldsymbol{m}}\}$ are pre-computed using Equations (5) and (6) over batches of data drawn from p_d. The sampling strategy remains unchanged from the baselines [24, 25]. In NC-FoLDv2, we incorporate noise conditioning by scaling the output of s_{FSN} by the noise variance. In the NC-FoLD, the score is computed on noise-convolved versions of p_d. We recompute the Fourier coefficients at the start of each noise level while sampling.

Algorithm 1: Sampling and coefficient estimation for FoLD, NC-FoLD, and NC-FoLDv2 variants.

Inputs: Data $\boldsymbol{x} \sim p_d$, noise levels $\{\sigma_\ell\}_{\ell=1}^{L}$, $p_{\sigma_\ell} = \mathcal{N}(\mathbf{0}, \sigma_\ell^2\mathbb{I})$, # steps T_ℓ, # iterations $T_{\max}$.
Fourier parameters: Score network s_{FSN}, period T, truncation order M, batch size B, # batches N_B.
<u>Fourier Langevin Diffusion (FoLD) sampler:</u>
if *FoLD or NC-FoLDv2* **then**
  Compute: Fourier coefficients $\alpha_{\boldsymbol{m}}$ of p_d
Initialize: $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$
for $\ell = 1$ *to* L **do**
  if *NC-FoLD* **then**
    Compute: Fourier coefficients of $p_d * p_{\sigma_\ell}$
  for $t = 1$ *to* T_ℓ **do**
    Draw: $\boldsymbol{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$
    Compute: Annealing parameter γ_t
    Sample: $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma_t\, s_{\text{FSN}}(\boldsymbol{x}_t) + \sqrt{2\gamma_t}\boldsymbol{z}_t$
Output: Samples output by FoLD: $\boldsymbol{x}_t$ at $t = T_{\max}$
<u>Fourier Score Network (FSN) coefficients:</u>
while *j = 1,2,. . . ,*N_B **do**
  if *FoLD or NC-FoLDv2* **then**
    Sample: $\{\boldsymbol{x}_k;\ k = 1, 2, \ldots, B,\ \boldsymbol{x}_k \sim p_d\}$
  else if *NC-FoLD* **then**
    Sample: $\{\boldsymbol{x}_k;\ k{=}1, \ldots, B,\ \boldsymbol{x}_k \sim p_d * p_{\sigma_\ell}\}$
  Compute: Estimate $\alpha^i_{\boldsymbol{m},j}$; $\alpha^r_{\boldsymbol{m},j}$ (Eqs. (5)-(6))
Compute: $\alpha^r_{\boldsymbol{m}} {=} \frac{1}{N_B}\sum_j \alpha^r_{\boldsymbol{m},j}$; $\alpha^i_{\boldsymbol{m}} {=} \frac{1}{N_B}\sum_j \alpha^i_{\boldsymbol{m},j}$
Update: Fourier score network s_{FSN} with $\alpha^i_{\boldsymbol{m}}$, $\alpha^r_{\boldsymbol{m}}$

4. Experimental Validation

To validate the performance of the proposed Fourier score network, we compare FoLD variants with NCSN [24], and NCSNv2 [25], and SSM [26]. Consider the two-component Gaussian target distribution from [24]: $p_d(\boldsymbol{x}) = \frac{1}{5}\mathcal{N}(-\mathbf{5}, \mathbb{I}) + \frac{4}{5}\mathcal{N}(\mathbf{5}, \mathbb{I})$, where $\mathbf{5}$ is a 2-D vector with both entries equal to 5. The baseline variants consider a three-layer fully-connected network with 128 nodes and `softplus` activation in the hidden layers. We train the score network with SSM and NCSN losses over 10^4 iterations with the Adam optimizer [16]. The learning rate is set to 0.001, and the batch size is set to 128. For the Fourier score network, based on the analysis presented in [1], and the ablation experiments reported in Section 4.1 (cf. Figure 1), we consider the period $T = 20$, with 50 harmonics, *i.e.,* $M = 50$. The FSN coefficients are computed over 2×10^6 samples, comprising 1000 batches of 2000 samples each. The coefficient estimation in FSN takes 1.32 ± 0.05 seconds, while training the score network s_θ in the baselines takes 20.266 ± 0.2 seconds, *i.e.,* FSNs demonstrate nearly a $20\times$ speedup! In terms of parameter count, the score network in NCSN models consist of nearly 17×10^3 trainable parameters, whereas, for FSNs, with $M = 50$, we have 15×10^3 parameters, of which 5×10^3 are preselected harmonics $[M]^n$, and 10^4 coefficients $\{\alpha_{\boldsymbol{m}}\}$, computed via Equations (5) and (6).

On Gaussian learning tasks, we observe that FoLD is able to model the score field accurately, with a significantly lower network complexity and *training time*. For sampling on the 2-D GMM, the baselines consider $\{\sigma_\ell\}_{\ell=1}^{L}$ to be a geometric progression, with $L = 10$, $\sigma_1 = 20$ and $\sigma_{10} = 1$. The sampler is run for $T_L = 100$ steps for each noise level. In FoLD variants, we consider identical noise levels, but run the sampler for $T_L = 10$ steps.

An in [1], FSNs suffer from the *curse of dimensionality* when scaled to high dimensions. We therefore employ the FoLD sampler on the latent-space of images learnt by an autoencoder. We present results on MINST, SVHN and CelebA datasets, with 15-, 32-, and 63-dimensional latent-space representations, respectively. We observe a two-fold improvement in convergence over the baseline NCSN, with the FoLD sampler being run for $T_L = 50$ steps, which is attributed to the superior approximation of the score by the FSN over the baselines.

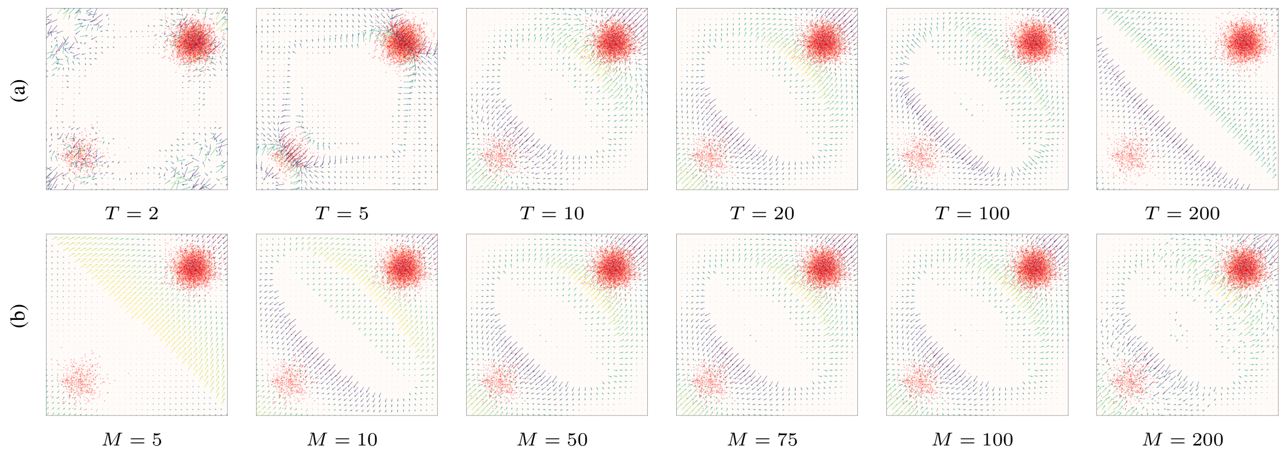

Figure 1. (Color online) Comparison of the score field generated by the Fourier score network in FoLD for various choices of (a) the fundamental period T, for truncation order $M = 75$; and (b) the truncation order M, for $T = 20$, when modeled on the two-component GMM $p_d = \frac{1}{5}\mathcal{N}(-\mathbf{5}, \mathbb{I}) + \frac{4}{5}\mathcal{N}(\mathbf{5}, \mathbb{I})$. We observe that, when T underestimates the spread of the data, as is the case for $T = 2$, there are aliasing artifacts. For large values of T or small values of M ($T > 100$ or $M < 10$ in the example considered), the score field does not accurately point towards p_d. Here, setting $T = 20$ and $M = 75$ is a viable compromise between compute load and approximation quality.

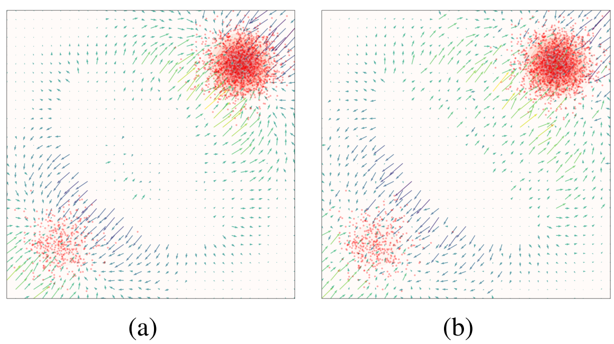

Figure 2. (Color online) Comparison of the score fields generated by (a) the Fourier score network in FoLD; and (b) auto-differentiation applied to the Fourier approximation of p_d in FoLD.

Figure 3. (Color online) Images generated by FoLD, when the FSN coefficients are modelled on the latent-space of images.

4.1. Ablation Experiments

In our ablation studies, we compared the closed-form Fourier Score Network (FSN) with the auto-differentiation method applied to the Fourier-series approximation of p_d. As per the results in Figure 3, both methods show similar performance. However, FSNs are more efficient, requiring approximately 0.0145 ± 0.002 seconds per forward pass during sampling, compared to 0.075 ± 0.003 seconds for each iteration using the auto-differentiation method. This indicates that FSNs are approximately four times faster.

We also investigated how the truncation order M and the assumed period T in the Fourier-series impact score approximation. As shown in Figure 1, accurate estimations of T improve performance, aligning with findings in [1]. However, underestimating T leads to aliasing and convergence to aliased targets. Beyond $M > 50$, we noticed a significant increase in computation time without substantial improvements in approximation quality. Consequently, we limited the Fourier series to $M = 50$ harmonics.

5. Discussion and Conclusions

This extended abstract considers the challenge of score-based Langevin diffusion, introducing an efficient approach to computing the score via a Fourier-series approximation. The Fourier score network eliminates the need for traditional training, relying instead on closed-form calculations of Fourier coefficients, significantly reducing pre-processing times in score-based sampling. As a proof-of-concept, we validated our approach with experiments on 2-D synthetic Gaussian data and conducted ablation studies on noise-conditional score computation, fundamental period T, and truncation length M. Our method demonstrated a two-fold speedup in sampling and a $20\times$ speedup by avoiding gradient-descent-based training.

Key future research directions include investigating higher-order discretization [10, 20, 30, 31], momentum-based acceleration [5] in sampling algorithms, and adapting the Fourier score networks to model latent-space distributions in high-resolution autoencoder-based models [21].

References

[1] S. Asokan and C. S. Seelamantula. Euler-Lagrange analysis of generative adversarial networks. *Journal of Machine Learning Research (JMLR)*, pages 1–100, 2023. 1, 2, 3, 4

[2] S. Asokan, N. Shetty, A. Srikanth, and C. S. Seelamantula. GANs settle scores! *arXiv preprints, arXiv:2306.01654*, 2023. 2

[3] B. Bonev, T. Kurth, C. Hundt, J. Pathak, M. Baust, K. Kashinath, and A. Anandkumar. Spherical Fourier neural operators: Learning stable dynamics on the sphere. *arXiv preprint arXiv:2306.03838*, 2023. 1

[4] P. Dhariwal and A. Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, pages 8780–8794, 2021. 1

[5] T. Dockhorn, A. Vahdat, and K. Kreis. Score-based generative modeling with critically-damped Langevin diffusion. In *International Conference on Learning Representations*, 2022. 4

[6] J.-Y. Franceschi, M. Gartrell, L. D. Santos, T. Issenhuth, E. de Bézenac, M. Chen, and A. Rakotomamonjy. Unifying gans and score-based diffusion as generative particle models. *arXiv preprint, arXiv:2305.16150*, abs/2305.16150, 2023. 2

[7] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *arXiv preprint, arXiv:2006.11239*, 2020. 1

[8] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47): 1–33, 2022. 1

[9] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. 2

[10] A. Jolicoeur-Martineau, K. Li, R. Piché-Taillefer, T. Kachman, and I. Mitliagkas. Gotta go fast with score-based generative models. In *The NeurIPS DLDE-I Workshop*, 2021. 1, 2, 4

[11] T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022. 1, 2

[12] B. Kawar, G. Vaksman, and M. Elad. SNIPS: Solving noisy inverse problems stochastically. *Advances in Neural Information Processing Systems*, 34:21757–21769, 2021. 1

[13] B. Kawar, M. Elad, S. Ermon, and J. Song. Denoising diffusion restoration models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022.

[14] B. Kawar, J. Song, S. Ermon, and M. Elad. Jpeg artifact correction using denoising diffusion restoration models. In *Neural Information Processing Systems (NeurIPS) Workshop on Score-Based Methods*, 2022. 1

[15] D. Kim, Y. Kim, S. J. Kwon, W. Kang, and I. Moon. Refining generative process with discriminator guidance in score-based diffusion models. In *Intl. Conf. on Machine Learning*, 2023. 2

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015. 3

[17] N. Kovachki, S. Lanthaler, and S. Mishra. On universal approximation and error bounds for Fourier neural operators. *Journal of Machine Learning Research*, 22(290):1–76, 2021. 1

[18] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. 1

[19] J. H. Lim et al. Score-based diffusion models in function space. *arXiv preprint arXiv:2302.07400*, 2023. 2

[20] H. Ma, L. Zhang, X. Zhu, and J. f Feng. Accelerating score-based generative models with preconditioned diffusion sampling. In *The 17th European Conf. on Computer Vision*, 2022. 1, 4

[21] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 4

[22] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, pages 7462–7473, 2020. 1

[23] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *Intl. Conf. on Learning Representations*, 2021. 1

[24] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019. 1, 2, 3

[25] Y. Song and S. Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems 33*, 2020. 1, 3

[26] Y. Song, S. Garg, J. Shi, and S. Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, pages 574–584, 2020. 1, 2, 3

[27] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1

[28] C. White, R. Tu, J. Kossaifi, G. Pekhimenko, K. Azizzadenesheli, and A. Anandkumar. Speeding up Fourier neural operators via mixed precision. *arXiv:2307.15034*, 2023. 1

[29] Z. Xiao, K. Kreis, and A. Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations (ICLR)*, 2022. 2

[30] Q. Zhang and Y. Chen. Fast sampling of diffusion models with exponential integrator. In *The 11th International Conference on Learning Representations*, 2023. 1, 4

[31] Q. Zhang, J. Song, and Y. Chen. Improved order analysis and design of exponential integrator for diffusion models sampling. *arXiv:2308.02157*, 2023. 1, 4

[32] H. Zheng, W. Nie, A. Vahdat, K. Azizzadenesheli, and A. Anandkumar. Fast sampling of diffusion models via operator learning. In *International Conference on Machine Learning*, pages 42390–42402. PMLR, 2023. 1, 2