

SCORENF: SCORE-BASED NORMALIZING FLOWS FOR SAMPLING UNNORMALIZED DISTRIBUTIONS

Vikas Kanaujia, Vipul Arora

Department of Electrical Engineering, IIT Kanpur, India

ABSTRACT

Unnormalized probability distributions are central to modeling complex physical systems across various scientific domains. Traditional sampling methods, such as Markov Chain Monte Carlo (MCMC), often suffer from slow convergence, critical slowing down, poor mode mixing, and high autocorrelation. In contrast, likelihood-based and adversarial machine learning models, though effective, are heavily data-driven, requiring large datasets and often encountering mode covering and mode collapse. In this work, we propose ScoreNF, a score-based learning framework built on the Normalizing Flow (NF) architecture, integrated with an Independent Metropolis-Hastings (IMH) module, enabling efficient and unbiased sampling from unnormalized target distributions. We show that ScoreNF maintains high performance even with small training ensembles, thereby reducing reliance on computationally expensive MCMC-generated training data. We also present a method for assessing mode-covering and mode-collapse behaviours. We validate our method on synthetic 2D distributions (MOG-4 and MOG-8) and the high-dimensional ϕ^4 lattice field theory distribution, demonstrating its effectiveness for sampling tasks.

Index Terms— MCMC, Score modelling, Normalising Flows, Independent Metropolis Hastings.

1. INTRODUCTION

Unnormalized probability distributions—such as the Boltzmann distribution of form $e^{-H(x)}$ are extensively encountered in a wide range of scientific fields, including statistical physics, biological sciences and Bayesian inference [1, 2, 3, 4]. These distributions often arise in scenarios where the partition function or normalization constant is analytically intractable, particularly in high-dimensional settings. Efficient sampling from such distributions is crucial for estimating physical observables, performing uncertainty quantification, and enabling posterior inference. However, traditional sampling methods like Markov Chain Monte Carlo (MCMC) [5] face significant limitations, including slow convergence, critical slowing down, poor mode mixing, and high autocorrelation between samples.

In recent years, machine learning-based approaches have shown considerable promise in addressing these limitations. Several generative models like Normalising flows [6, 7, 8], GAN [9, 10], VAE [11, 12] and Diffusion models [13] have been proposed for approximating target distributions. By leveraging neural networks, these methods can learn complex transformations that map simple base distributions to the target distribution, thus enabling efficient sampling.

In the context of un-normalized distributions, Boltzmann Generators [14] leveraging Normalizing Flows are commonly used to generate independent, unbiased samples from the Boltzmann distribution by learning invertible mappings from a simple latent space

to the complex target distribution. Despite their strengths, Boltzmann Generators present several limitations. Their training typically depends on MCMC-generated samples to estimate KL divergence, reintroducing significant computational overhead. Moreover, the use of the forward KL divergence (FKL) as an objective induces mode-covering behavior—i.e., assigning probability mass to low-density regions and resulting in high variance in downstream observable estimates. In contrast, optimizing with reverse KL divergence (RKL) tends to induce mode collapse, as it emphasizes only a few of the high-density regions while neglecting low-probability modes, thereby failing to capture the full support of the target distribution and leading to biased estimates of observables.

Similarly other machine learning models based on likelihood maximization or adversarial training offer alternative solutions, they typically require large datasets and often suffer from issues such as mode collapse and mode covering. This motivates the development of more efficient and reliable sampling frameworks for unnormalized distributions, particularly in data-scarce or computationally demanding regimes.

Score-based generative models [15], that aim to learn the score function $\nabla_x \log p_d(x)$ —the gradient of the data log-density—have been widely applied in various domains such as image generation [16], audio generation [17, 18], and more. Once trained, these models enable sampling via Langevin dynamics, where the learned score function—estimated through score matching—guides the iterative generation of samples from the target distribution. In this paper, we incorporate the learning philosophy of score estimation into the Normalizing Flow (NF) framework. Unlike traditional score-based models that rely on iterative Langevin dynamics for sampling, our approach enables sample generation through a single-pass transformation of noise, leveraging the invertibility of the flow architecture. Our main contributions are as follows:

- We propose ScoreNF, a framework that integrates score-based learning into normalizing flows (NFs) to model the target distribution more effectively. By leveraging score information, ScoreNF enhances the expressiveness of the model, resulting in improved approximation fidelity and mitigating common issues such as mode collapse and mode coverage.
- We demonstrate that our proposed method, ScoreNF, maintains near-identical performance even with a significantly reduced ensemble size during training. This substantially lowers the reliance on computationally expensive MCMC simulations for data generation.
- We assess the efficacy of the proposed method across several benchmark distributions, including MOG-4, MOG-8, and the 64-dimensional scalar ϕ^4 theory distribution.
- Additionally, we present an evaluation method to assess the extent of mode coverage and mode collapse in the modelled distribution.

2. METHOD

2.1. Problem Formulation

Given a dataset of samples $\mathcal{D} = \{\mathbf{x}_i \in \mathcal{X}\}_{i=1}^N$ drawn from a target distribution $p(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^d$, the objective is to learn a parameterized generative model $q_\theta(\mathbf{x})$ such that $q_\theta(\mathbf{x}) \approx p(\mathbf{x})$. The goal is to achieve a high-fidelity approximation of the target distribution while utilizing as few samples as possible in the learning.

2.2. Background

Here, we briefly review the key principles of score-based generative models and normalizing flows relevant to our proposed approach.

Score-based generative models [15, 19] aim to approximate an unknown data distribution $p(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^d$, by learning its score function, defined as the gradient of the log-density, $\nabla_{\mathbf{x}} \log p(\mathbf{x})$.

Rather than modeling density $p(\mathbf{x})$ directly, these methods learn a parametric score function $s_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x})$ estimated by minimizing the Fisher divergence between the true and model scores:

$$\mathcal{L}(\theta) = \frac{1}{2} \mathbb{E}_{p(\mathbf{x})} [\|s_\theta(\mathbf{x}) - \nabla_{\mathbf{x}} \log p(\mathbf{x})\|^2] \quad (1)$$

In the absence of true score $s(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x})$, $\mathcal{L}(\theta)$ can be simplified under certain regularity conditions [20] as

$$\mathcal{L}(\theta) = \mathbb{E}_{p(\mathbf{x})} [\text{Tr}(\nabla_{\mathbf{x}} s_\theta(\mathbf{x})) + \frac{1}{2} \|s_\theta(\mathbf{x})\|^2] \quad (2)$$

where Tr is the trace. It could also be approximated using denoising score matching (DSM)[15]. Once the score function is learned, new samples can be generated by simulating a reverse-time stochastic differential equation (SDE) or Langevin dynamics [15, 19].

Normalising flows [6, 7] construct complex probability distributions by applying a sequence of invertible transformations to a simple base distribution. Given a base random variable $\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$, a normalizing flow defines a bijective mapping $\mathbf{x} = f_\theta(\mathbf{z})$, where f_θ is an invertible function with tractable Jacobian determinant.

The probability density of $\mathbf{x}$ under the model $q_\theta(\mathbf{x})$ is obtained using the change of variables formula:

$$q_\theta(\mathbf{x}) = p_{\mathbf{z}}(f_\theta^{-1}(\mathbf{x})) \left| \det \left(\frac{\partial f_\theta^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right) \right| \quad (3)$$

By designing f_θ such that both the inverse and the Jacobian determinant are tractable, normalizing flows allow for exact likelihood estimation and efficient sampling.

Independent Metropolis Hastings (IMH) Algorithm: The Metropolis-Hastings (MH) algorithm is a Markov Chain Monte Carlo (MCMC) method for sampling from a target distribution $p(x)$, known either exactly or up to a normalizing constant. It constructs a Markov chain that satisfies the detailed balance condition [21]. A new sample $\mathbf{x}'$ is proposed from a distribution $q(\mathbf{x}'|\mathbf{x})$, and accepted with probability:

$$p_{\text{accept}}(\mathbf{x}'|\mathbf{x}) = \min \left(1, \frac{q(\mathbf{x}|\mathbf{x}')p(\mathbf{x}')}{q(\mathbf{x}'|\mathbf{x})p(\mathbf{x})} \right) \quad (4)$$

In the Independent Metropolis-Hastings (IMH) variant [22], proposals are drawn independently of the current state, i.e., $q(\mathbf{x}'|\mathbf{x}) = q(\mathbf{x}')$, simplifying the acceptance probability to:

$$p_{\text{accept}}(\mathbf{x}'|\mathbf{x}) = \min \left(1, \frac{q(\mathbf{x})p(\mathbf{x}')}{q(\mathbf{x}')p(\mathbf{x})} \right) \quad (5)$$

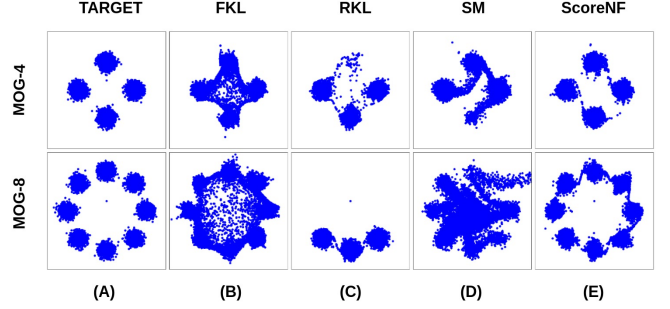


Fig. 1. Sample plots for the MOG-4 and MOG-8 distributions generated using FKL, RKL, SM, and ScoreNF. Mode collapse is evident for the RKL approach, while mode covering behavior is observed in both FKL and SM approaches across all synthetic distributions.

This is useful when sampling from $p(\mathbf{x})$ is intractable but its unnormalized density is accessible.

2.3. Proposed Method

In the NF framework, training is typically performed by minimizing either the forward KL divergence (FKL), i.e., $KL(p||q_\theta)$, or the reverse KL divergence (RKL), i.e., $KL(q_\theta||p)$, where $p(\mathbf{x})$ denotes the target distribution and $q_\theta(\mathbf{x})$, the modelled distribution parameterized by the flow parameters, θ . Optimization using FKL requires samples from the target distribution, as it relies on maximizing likelihood. This approach tends to induce a mode-covering behavior in the learned distribution, encouraging coverage of low probability regions with significant probability mass, as illustrated in Fig. 1(B). In contrast, RKL-based training requires access only to an explicit form of the target density function; it does not require samples from the target distribution. Training samples are drawn from the base distribution and transformed through the flow model. However, RKL minimization typically exhibits a mode-seeking tendency, often leading to mode collapse where the learned distribution fails to capture all modes of the target, as illustrated in Fig. 1(C). To address the limitations associated with both mode-collapse (arising from RKL minimization) and mode-covering (from FKL minimization), we leverage a score-matching objective to train NF. RKL optimization tends to concentrate on a few dominant modes. In contrast, score-based learning utilizes target samples as anchor points for matching local gradient estimates, which promotes more balanced coverage across all modes. Score learning, when combined with RKL, facilitates the learning of all modes in the target distribution, thereby reducing mode collapse.

Incorporating score-based learning within the NF framework requires computation of both the target score function, $s(\mathbf{x})$, and the model score, $s_\theta(\mathbf{x})$. Given that the target density $p(\mathbf{x})$ is either known exactly or even up to normalizing constant, the exact score function can be evaluated analytically, given by:

$$s(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}) \quad (6)$$

$$= \nabla_{\mathbf{x}} \log \tilde{p}(\mathbf{x}), \quad p(\mathbf{x}) \propto \tilde{p}(\mathbf{x}) \quad (7)$$

$$= -\nabla_{\mathbf{x}} H(\mathbf{x}), \quad \tilde{p}(\mathbf{x}) = e^{-H(\mathbf{x})} \quad (8)$$

While prior approaches [20, 15] directly parameterize the score function using neural networks, NFs explicitly model the density (Eq. 3). This enables direct computation of the model score via automatic

	MOG-4				MOG-8			
	NLL↓	RNLL(↓)	ESS(↑)	AR↑	NLL↓	RNLL(↓)	ESS(↑)	AR↑
FKL	2.53 ± 0.10	3.54 ± 0.89	50.76 ± 23.77	73.28 ± 0.73	3.30 ± 0.04	4.21 ± 0.45	45.37 ± 21.30	62.31 ± 1.00
RKL	3.00 ± 0.64	2.21 ± 0.02	18.73 ± 5.40	72.42 ± 5.89	372.84 ± 232.36	2.94 ± 0.13	78.83 ± 24.17	92.77 ± 0.44
SM	2.94 ± 0.05	3.33 ± 0.10	33.72 ± 1.96	46.24 ± 1.56	4.72 ± 0.32	14.04 ± 2.91	34.41 ± 13.83	18.35 ± 5.15
ScoreNF	2.21 ± 0.01	2.27 ± 0.02	83.06 ± 24.82	88.75 ± 0.42	3.01 ± 0.02	3.07 ± 0.06	48.08 ± 21.86	77.51 ± 0.84

Table 1. Results for the MOG-4 and MOG-8 distributions using 1,000 training samples. Reported values represent the average and standard error computed over three random seeds for each method.

	NLL(↓)	RNLL(↓)	ESS(↑)	AR(%) (↑)
FKL	13.26 ± 0.10	-10.35 ± 0.98	27.76 ± 7.24	19.06 ± 0.97
RKL	22.63 ± 3.85	-18.93 ± 0.97	41.76 ± 0.99	68.29 ± 3.55
SM	16.31 ± 0.10	-20.74 ± 0.12	33.77 ± 12.96	68.41 ± 1.76
ScoreNF	12.05 ± 0.02	-16.31 ± 0.21	45.52 ± 0.41	55.75 ± 0.55

Table 2. Results for ϕ^4 model distribution ($d = 64$) using 10,000 training samples. Reported values represent the average and standard error computed over three random seeds for each method.

differentiation, $s_\theta(\mathbf{x}) = \nabla_{\mathbf{x}} \log q_\theta(\mathbf{x})$. The score matching objective simplifies to:

$$\mathcal{L}_{SM}(\theta) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \|\nabla_{\mathbf{x}} \log q_\theta(\mathbf{x}) - \nabla_{\mathbf{x}} \log p(\mathbf{x})\|^2 \quad (9)$$

$$= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \|\nabla_{\mathbf{x}} \log q_\theta(\mathbf{x}) + \nabla_{\mathbf{x}} H(\mathbf{x})\|^2 \quad (10)$$

Score-based training alone is insufficient to fully capture the target distribution, as it primarily aligns local gradient information rather than global density structure. To enhance distribution learning, we incorporate both the RKL and a score-matching objective into the training loss. The overall training objective is defined as follows:

$$\mathcal{L}_{net}(\theta) = \mathcal{L}_{RKL}(\theta) + \lambda_1 \mathcal{L}_{SM}(\theta) \quad (11)$$

Where λ_1 is a hyperparameter, $\mathcal{L}_{SM}$ is defined in Eq. 10, and $\mathcal{L}_{RKL}$ is defined as follows:

$$\mathcal{L}_{RKL} = \mathbb{E}_{\mathbf{z}} \left[\log p_{\mathbf{z}}(\mathbf{z}) - \log \left| \det \frac{\partial f_\theta(\mathbf{z})}{\partial \mathbf{z}} \right| - \log p(f_\theta(\mathbf{z})) \right] \quad (12)$$

Score matching aligns the score functions—i.e., the gradients of log-densities—of the model and target, its behaviour encourages broader mode coverage [23]. In contrast, the RKL objective inherently exhibits a mode-seeking bias, which can lead to mode collapse. To mitigate this issue, we introduce the score-matching loss as a regularizer to the RKL loss. This combination helps preserve multiple modes during optimization. As training progresses and the model becomes more aligned with the target distribution, the weight of the score-matching loss is gradually annealed down, allowing the RKL term to refine the density estimation without inducing mode-collapse.

One may wonder if score matching here could be replaced by FKL, which is also sample driven. However, these desired effects are not observed with FKL. This discrepancy arises from the fundamental differences between score learning and FKL optimization. FKL, primarily based on likelihood estimation, depends exclusively on samples and not on the target density $p(\mathbf{x})$ itself, whereas score learning uses both the samples as well as $p(\mathbf{x})$. As a result, spurious samples have little impact on score learning but can significantly distort the distribution learned with FKL. Even with limited samples,

score learning effectively guides the modeled distribution toward the target. We observe the same empirically as well.

After training, the NF model can generate samples and evaluate their exact densities. To correct for sampling bias introduced by model approximation, we apply the IMH algorithm, using the learnt distribution as the proposal. Samples are then accepted or rejected based on the acceptance probability defined in Eq. 5, yielding asymptotically unbiased samples from the target distribution.

3. EXPERIMENTS

This section outlines the experiments conducted on various distributions, including brief descriptions of each distribution and the evaluation metrics used.

3.1. Distributions

MOG-4 and MOG-8: To assess model performance, we use synthetic 2D Mixture of Gaussians distributions—MOG-4 and MOG-8 [24]. These benchmarks provide clear insights into mode coverage and mode collapse, as the ground truth distribution and its individual modes are known in advance. MOG-4 consists of a mixture of 4 Gaussian components, whereas MOG-8 contains 8 components. Using the PyTorch distribution library, we generate 10,000 samples for training and 5,000 samples each for validation and testing, corresponding to $N = 4, 8$.

Scalar ϕ^4 theory: It is a computational physics model to study scalar field theories [25]. Given a scalar field $\mathbf{x} \in \mathbb{R}^d$ defined on a 2D square lattice with d sites, the energy function, $H(\mathbf{x})$ is given by:

$$H(\mathbf{x}) = \sum_{l=1}^d \left(\lambda_1 x_l^4 + \lambda_2 x_l^2 + 2 \sum_{l' \in n(l)} (x_l^2 - x_l x_{l'}) \right) \quad (13)$$

where $n(l)$ denotes the set of lattice sites adjacent to the l^{th} site. The coupling constants are set to $\lambda_1 = 4$ and $\lambda_2 = -4$. HMC is employed to generate 10,000 samples for training, and 5,000 samples each for validation and testing, all corresponding to the system size $d = 64$.

3.2. Metrics

We use the following four metrics for evaluation:

Assessing Mode Collapse and Mode Coverage: Generally, **Negative Log-Likelihood (NLL)** is used to measure how well the model fits the data.

$$\text{NLL} = -\mathbb{E}_{p(\mathbf{x})} [\log(q_\theta(\mathbf{x}))] \quad (14)$$

A high NLL suggests mode collapse. But generally, NLL values are not significantly affected in case of mode coverage. We present

Sample Size	Model	MOG-4				MOG-8			
		NLL↓	RNLL(↓)	ESS(↑)	AR↑	NLL↓	RNLL(↓)	ESS(↑)	AR↑
10000	FKL	2.20 ± 0.01	2.38 ± 0.09	89.74 ± 6.20	84.81 ± 0.36	2.92 ± 0.00	3.06 ± 0.04	74.89 ± 17.64	82.29 ± 0.36
	ScoreNF	2.18 ± 0.01	2.26 ± 0.02	96.22 ± 3.21	90.38 ± 0.52	2.94 ± 0.01	3.00 ± 0.05	68.26 ± 21.44	80.10 ± 0.71
1000	FKL	2.53 ± 0.10	3.54 ± 0.89	50.76 ± 23.77	73.28 ± 0.73	3.30 ± 0.04	4.21 ± 0.45	45.37 ± 21.30	62.31 ± 1.00
	ScoreNF	2.21 ± 0.01	2.27 ± 0.02	83.06 ± 24.82	88.75 ± 0.42	3.01 ± 0.02	3.07 ± 0.06	48.08 ± 21.86	77.51 ± 0.84
250	FKL	3.20 ± 0.13	7.08 ± 0.38	54.43 ± 24.10	34.84 ± 0.96	3.91 ± 0.06	6.86 ± 0.91	41.43 ± 17.73	32.32 ± 0.63
	ScoreNF	2.24 ± 0.02	2.27 ± 0.04	70.06 ± 23.15	84.50 ± 0.69	3.11 ± 0.04	3.14 ± 0.13	43.62 ± 25.68	77.67 ± 1.66

Table 3. Performance comparison of the ScoreNF model with FKL across varying training sample sizes on the MOG-4 and MOG-8 distributions.

Sample Size	Model	NLL(↓)	RNLL(↓)	ESS(↑)	AR↑
10000	FKL	13.26 ± 0.10	-10.35 ± 0.98	27.76 ± 7.24	19.06 ± 0.97
	ScoreNF	12.05 ± 0.02	-16.31 ± 0.21	45.52 ± 0.41	55.75 ± 0.55
1000	FKL	14.49 ± 0.18	-7.65 ± 1.14	30.24 ± 3.85	13.52 ± 2.12
	ScoreNF	12.06 ± 0.00	-16.65 ± 0.06	33.99 ± 2.65	56.41 ± 0.80
250	FKL	17.84 ± 0.08	-7.11 ± 1.32	23.74 ± 1.99	9.11 ± 1.06
	ScoreNF	12.09 ± 0.02	-15.88 ± 0.06	34.22 ± 2.66	53.51 ± 1.33

Table 4. Performance comparison of the ScoreNF model with FKL across varying training sample sizes on the Scalar ϕ^4 theory distribution.

reverse NLL (RNLL) as a strong indicator of mode coverage.

$$\text{RNLL} = -\mathbb{E}_{q_\theta(\mathbf{x})}[\log p(\mathbf{x})] \quad (15)$$

Together with NLL, this metric diagnoses the distribution behavior. High NLL and low RNLL indicate mode collapse, low NLL and high RNLL indicate mode coverage, while low values for both indicate that the model closely matches the target distribution.

Effective Sample Size (ESS) : ESS measures the quantity of independent information contained within a sample and is defined as

$$\text{ESS} = \frac{(\frac{1}{N} \sum_i p(\mathbf{x}_i)/q_\theta(\mathbf{x}_i))^2}{\frac{1}{N} \sum_i (p(\mathbf{x}_i)/q_\theta(\mathbf{x}_i))^2} \quad (16)$$

Higher values indicate more effective sampling of the target distribution.

Acceptance Rate (AR) : It is the percentage of accepted samples out of all evaluated samples in the IMH algorithm [24].

We compare the proposed method against three baselines: NF trained with forward KL divergence (FKL), NF trained with reverse KL divergence, and NF trained via score matching (SM).

4. RESULTS AND DISCUSSION

4.1. MOG-4 and MOG-8

For the synthetic distributions, we compute all the evaluation metrics, the results of which are summarized in Table 1. A comparative visualization of the generated samples across all methods is provided in Fig. 1. We observe that FKL-based training exhibits mode-covering behaviour. In contrast, RKL-based training suffers from mode collapse, failing to capture one mode in the MOG-4 setting and up to five modes in MOG-8, which is further evidenced by increased NLL values. Despite these omissions, RKL may achieve better ESS, AR and RNLL scores due to its tendency to concentrate probability mass on fewer, high-density modes, thereby reducing penalization from uncovered regions. This is a limitation of the metrics, and not a merit of the method. Hence, NLL and RNLL should be seen together.

In contrast, our proposed method, ScoreNF, achieves superior performance, with sample plots closely aligning with the target distribution. This visual accuracy is corroborated by low NLL and RNLL values, reflecting better density estimation.

4.2. Scalar ϕ^4 distribution

The results for the scalar ϕ^4 distribution are presented in Table 2. ScoreNF outperforms FKL-, RKL-, and SM-based methods, achieving low NLL values without mode collapse. Although RKL and SM exhibit low RNLL values indicative of limited mode coverage, their high NLL values reveal substantial mode collapse. The RNLL of ScoreNF is slightly higher than that of RKL and SM, indicating relatively high mode coverage, but it is still lower than that of FKL, reflecting relatively reduced mode coverage. ScoreNF achieves significantly better overall performance by avoiding mode collapse and mode covering present in other methods.

Model performance with FKL, SM, and ScoreNF depends on the number of training samples from the target distribution. To assess this, we trained FKL and ScoreNF using ensembles of 10,000, 1,000, and 250 samples on both Synthetic and ϕ^4 distributions. Results in Tables 3 and 4 show that FKL performance degrades notably as ensemble size decreases, consistent with its likelihood-based estimation [26]. In contrast, ScoreNF remains robust and largely unaffected by reductions in ensemble size.

5. CONCLUSIONS

In this work, we present ScoreNF, a score-based learning method in the NF framework designed for sampling from unnormalized distributions. ScoreNF effectively addresses key challenges faced by traditional NF models, notably mitigating both mode collapse and mode covering. To assess mode collapse, negative log-likelihood (NLL) remains a reliable metric. To complement this, we use a metric, RNLL, which, when used alongside NLL, effectively quantifies the model’s mode coverage behavior. Combined, these metrics offer a robust framework for evaluating both mode collapse and mode coverage. ScoreNF, though reliant on target samples for score estimation, achieves strong performance even with limited data. Empirical results on the ϕ^4 distribution and synthetic distributions (MOG-4 and MOG-8) demonstrate its ability to capture complex multimodal distributions. Future work may extend ScoreNF to larger systems with complex dynamics and multiple symmetries.

6. REFERENCES

- [1] Tara Akhound-Sadegh, Jarrid Rector-Brooks, Avishek Joey Bose, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcín Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, Nikolay Malkin, and Alexander Tong, “Iterated denoising energy matching for sampling from boltzmann densities,” in *Proceedings of the 41st International Conference on Machine Learning*. 2024, pp. 760–786, PMLR.
- [2] Michael S Albergo, Gurtej Kanwar, and Phiala E Shanahan, “Flow-based generative models for markov chain monte carlo in lattice field theory,” *Physical Review D*, vol. 100, no. 3, pp. 034515, 2019.
- [3] Shuo-Hui Li and Lei Wang, “Neural network renormalization group,” *Physical review letters*, vol. 121, no. 26, pp. 260601, 2018.
- [4] Joey Bose, Tara Akhound-Sadegh, Guillaume Huguët, Kilian FATRAS, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael M Bronstein, and Alexander Tong, “Se (3)-stochastic flow matching for protein backbone generation,” in *The Twelfth International Conference on Learning Representations*, 2023.
- [5] Julian Besag, “An introduction to markov chain monte carlo methods,” in *Mathematical foundations of speech and language processing*, pp. 247–270. Springer, 2004.
- [6] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker, “Normalizing flows: An introduction and review of current methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 3964–3979, 2021.
- [7] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan, “Normalizing flows for probabilistic modeling and inference,” *Journal of Machine Learning Research*, vol. 22, no. 57, pp. 1–64, 2021.
- [8] Danilo Rezende and Shakir Mohamed, “Variational inference with normalizing flows,” in *International conference on machine learning*. PMLR, 2015, pp. 1530–1538.
- [9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [10] Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye, “A review on generative adversarial networks: Algorithms, theory, and applications,” *IEEE transactions on knowledge and data engineering*, vol. 35, no. 4, pp. 3313–3332, 2021.
- [11] Diederik P Kingma, Max Welling, et al., “An introduction to variational autoencoders,” *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.
- [12] Diederik P. Kingma and Max Welling, “Auto-encoding variational bayes,” *CoRR*, vol. abs/1312.6114, 2013.
- [13] Jonathan Ho, Ajay Jain, and Pieter Abbeel, “Denoising diffusion probabilistic models,” *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
- [14] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu, “Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning,” *Science*, vol. 365, no. 6457, pp. eaaw1147, 2019.
- [15] Yang Song and Stefano Ermon, “Generative modeling by estimating gradients of the data distribution,” *Advances in neural information processing systems*, vol. 32, 2019.
- [16] Nishanth Shetty and Chandra Sekhar Seelamantula, “Monte carlo score matching for image generation,” in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.
- [17] Santiago Pascual, Gautam Bhattacharya, Chunghsin Yeh, Jordi Pons, and Joan Serrà, “Full-band general audio synthesis with score-based diffusion,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [18] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan, “Wavegrad: Estimating gradients for waveform generation,” in *International Conference on Learning Representations*, 2021.
- [19] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole, “Score-based generative modeling through stochastic differential equations,” in *International Conference on Learning Representations*, 2021.
- [20] Aapo Hyvärinen, “Estimation of non-normalized statistical models by score matching,” *Journal of Machine Learning Research*, vol. 6, no. 24, pp. 695–709, 2005.
- [21] Siddhartha Chib and Edward Greenberg, “Understanding the metropolis-hastings algorithm,” *The american statistician*, vol. 49, no. 4, pp. 327–335, 1995.
- [22] James Brofos, Marylou Gabrié, Marcus A Brubaker, and Roy R Lederman, “Adaptation of the independent metropolis-hastings sampler with normalizing flow proposals,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 5949–5986.
- [23] Siwei Lyu, “Interpretation and generalization of score matching,” in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 359–366.
- [24] Vikas Kanaujia, Mathias S Scheurer, and Vipul Arora, “Advnf: Reducing mode collapse in conditional normalising flows using adversarial learning,” *SciPost Physics*, vol. 16, no. 5, pp. 132, 2024.
- [25] Ankur Singha, Dipankar Chakrabarti, and Vipul Arora, “Conditional normalizing flow for markov chain monte carlo sampling in the critical region of lattice field theory,” *Physical Review D*, vol. 107, no. 1, pp. 014512, 2023.
- [26] Samuel L Braunstein, “How large a sample is needed for the maximum likelihood estimator to be approximately gaussian?,” *Journal of Physics A: Mathematical and General*, vol. 25, no. 13, pp. 3813, 1992.