Efficient and Unbiased Sampling of Boltzmann Distributions via Consistency Models

Fengzhe Zhang[*]
University of Cambridge
fz287@cam.ac.uk

Jiajun He[*]
University of Cambridge
jh2383@cam.ac.uk

Laurence I. Midgley
Ångström AI
University of Cambridge
laurence@angstrom-ai.com

Javier Antorán
Ångström AI
University of Cambridge
javier@angstrom-ai.com

José Miguel Hernández-Lobato
Ångström AI
University of Cambridge
jmh233@cam.ac.uk

Abstract

Diffusion models have shown promising potential for advancing Boltzmann Generators. However, two critical challenges persist: (1) inherent errors in samples due to model imperfections, and (2) the requirement of hundreds of functional evaluations (NFEs) to achieve high-quality samples. While existing solutions like importance sampling and distillation address these issues separately, they are often incompatible, as most distillation models lack the necessary density information for importance sampling. This paper introduces a novel sampling method that effectively combines Consistency Models (CMs) with importance sampling. We evaluate our approach on both synthetic energy functions and equivariant n-body particle systems. Our method produces unbiased samples using only 6-25 NFEs while achieving a comparable Effective Sample Size (ESS) to Denoising Diffusion Probabilistic Models (DDPMs) that require approximately 100 NFEs.

1 Introduction

Sampling from Boltzmann distributions is a crucial task in statistical physics. Efficient sampling would enable the prediction of properties for new materials and drugs via computational simulations, reducing the need for costly experiments. However, the high dimensionality and multimodal nature of these distributions pose significant challenges. Traditional methods like Monte Carlo Markov Chain (MCMC) [10, 2] and Molecular Dynamics (MD) are often too time-consuming in complex settings.

Model-based Boltzmann generators [15] offer an alternative approach to amortize the sampling process. Diffusion Models (DMs) [19, 3] have shown promise in this regard, generating high-quality samples by gradually denoising from random noise. However, DMs face two major drawbacks: slow sample generation and biased estimations due to discrepancies between the model and true distribution.

While combining DMs (specifically, DDPM [3]) with Importance Sampling (IS) can address bias, it still requires hundreds of steps for a high Effective Sample Size (ESS). On the other hand, distillation techniques [16, 21, 8] can significantly reduce sampling time. Yet, applying importance sampling to distilled models remains challenging due to the absence of an explicit model density.

[*]Equal Contribution

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

In this work, leveraging recently developed Consistency Models (CMs) [21, 8, 9], we introduce an algorithm that significantly accelerates sampling while still supporting IS to correct the bias. Specifically, our contributions include:

- A novel method integrating IS with Bidirectional Consistency Models (BCMs) to accelerate sampling. This approach alternates between deterministic steps along an ODE trajectory and stochastic steps via an SDE for both proposal and target distributions. We use BCMs to accelerate the ODE part and the SDE to provide valid density for IS.
- Introduction of E(3)-Equivariant CMs for molecular applications and extension of Consistency Trajectory Models (CTMs) to Bidirectional CTMs (BCTMs), which shows more accurate bidirectional traversal than BCMs on molecular applications.
- Empirical verification on synthetic datasets and equivariant n-body systems, demonstrating unbiased sample production with only 6-25 NFEs, significantly outperforming the DDPM baseline.

2 Background

Before presenting our proposed method, we outline key preliminaries: Importance Sampling (IS), score-based Diffusion Models (DMs), and Bidirectional Consistency Models (BCMs).

Importance Sampling. To estimate integrals of the form $\mathbb{E}_{\boldsymbol{x}\sim p}[\phi(\boldsymbol{x})]$, where p is a target distribution and ϕ is an evaluable function, Self-Normalized Importance Sampling (SNIS) is commonly used when direct sampling from p is infeasible and only its unnormalized version $\bar{p}$ can be evaluated. Given a proposal distribution q from which sampling is possible, the integral can be approximated as:

$$\mathbb{E}_{\boldsymbol{x}\sim p}[\phi(\boldsymbol{x})] \approx \frac{\sum_{n=1}^{N} w_n \phi(\boldsymbol{x}^{(n)})}{\sum_{n=1}^{N} w_n} = \sum_{n=1}^{N} \bar{w}_n \phi(\boldsymbol{x}^{(n)}), \tag{1}$$

where $\boldsymbol{x}^{(n)} \sim q$, $w_n = \bar{p}(\boldsymbol{x}^{(n)})/q(\boldsymbol{x}^{(n)})$ are importance weights, and $\bar{w}_n = w_n / \sum_{m=1}^{N} w_m$ are normalized weights. The effectiveness of the IS estimator is measured by the Effective Sample Size (ESS): $\widehat{\text{ESS}} = 1/\sum_{n=1}^{N} \bar{w}_n^2$, with $1 \leq \widehat{\text{ESS}} \leq N$. SNIS provides asymptotically unbiased and consistent estimates, with bias and variance diminishing as the number of samples N increases.

Score-based Diffusion Models. Diffusion Models (DMs) generate samples by gradually removing noise from Gaussian samples. This process can be formulated as solving a reverse Stochastic Differential Equation (SDE), such as in the Denoising Diffusion Probabilistic Model (DDPM) [3], or a Probability Flow (PF) Ordinary Differential Equation (ODE) [23]: $d\boldsymbol{x} = \left[\boldsymbol{f}(\boldsymbol{x}, t) - \frac{1}{2} g(t)^2 \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] dt$, where $t \in [0, T]$, $\boldsymbol{f}$ and g are drift and diffusion coefficients of $\boldsymbol{x}_t$, and p_t is the marginal density at time t. Score-based DMs learn to approximate $\nabla \log p_t(\boldsymbol{x})$ using score matching [6, 22]. For sampling, this learned score function is used to solve the PF ODE from T to ϵ. From now on, we adopt the choices made by [7], setting $T = 80$, $\epsilon = 0.002$, $\boldsymbol{f}(\boldsymbol{x}, t) = \mathbf{0}$ and $g(t) = \sqrt{2t}$.

Consistency Models. A significant limitation of DMs is their slow sampling speed, often requiring hundreds of numbers of functional evaluations (NFEs). Consistency Models (CMs) [21] address this issue by directly learning the integral of the PF ODE, which enables mapping any point $\boldsymbol{x}_t$ at time t to the starting time ϵ along the same solution trajectory, allowing one-step or few-step sampling. Consistency Trajectory Models (CTMs) [8] extend CMs by learning to traverse from $\boldsymbol{x}_t$ at any time t to time u along the denoising direction (i.e., $u \leq t$) on the same solution trajectory. This extension provides greater flexibility in balancing sample quality and NFEs. Bidirectional Consistency Models (BCMs) [9] further generalize the approach by enabling both forward and backward traversal along the trajectory, offering more versatility in both sampling and the inversion process.

3 Method

[14, 13] provide a general framework for constructing asymptotically unbiased estimators in generative models involving both discrete and continuous random variables. In this paper, our baseline approach combines recently developed Diffusion Models (DMs) with Importance Sampling (IS) to

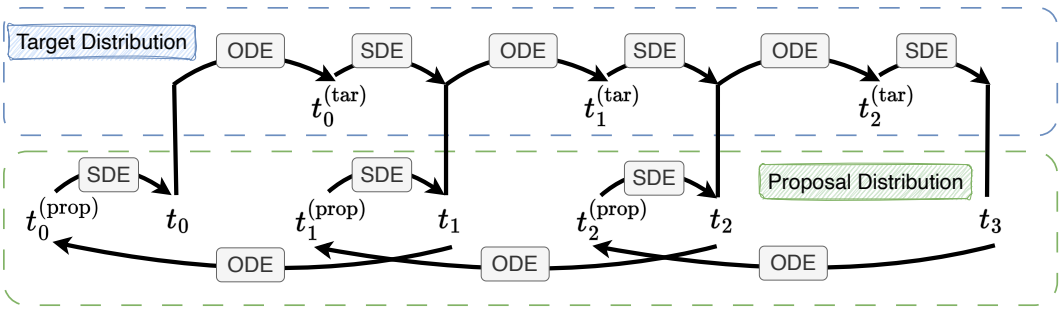

Figure 1: Overview of our proposed sampling method, which combines IS with BCMs.

achieve an asymptotically unbiased estimator. To implement this, we introduce sequential target and proposal distributions. Specifically, for a DDPM with N time steps, we define the proposal and target distributions in the joint space for $\boldsymbol{x}_{t_0}$ across all time steps. We include a detailed description of this baseline in Appendix A. While producing unbiased samples, this method is computationally inefficient, requiring $N \approx 100$ even for simple targets. Reducing N increases sample errors and proposal-target distribution deviations, negatively impacting IS effectiveness.

To address these challenges, we introduce Consistency Models (CMs) to reduce the required number of steps. Incorporating CMs to define the proposal distribution for importance sampling necessitates computing the density of the Probability Flow Ordinary Differential Equation (PF ODE). Ideally, using the relationship between Neural ODEs [1] and PF ODEs, this density could be computed via the instantaneous change of variables formula as explained by [23]. However, this computation is often expensive, and using the Skilling-Hutchinson trace estimator [18, 5] introduces bias in the density estimation, undermining the effectiveness of importance sampling. Consequently, CMs cannot be directly applied to IS. To yield proposal samples with well-defined and easily computable densities for IS, we add some Gaussian noise to the CM output. We can view CM as a deterministic traversal along the denoising PF ODE and view the additional Gaussian noise as a short traversal along the diffusion SDE. As for the target, we can simply add Gaussian noise according to the diffusion SDE to each time step.

However, this approach still faces efficiency issues due to the mismatch between the proposal and the target. This arises because the proposal, defined by CMs and SDEs, is a sequence of conditional Gaussians with nonlinear transformations over the means, while the target distribution is simply a sequence of linear conditional Gaussians. To mitigate this mismatch, a natural solution is to redefine the target using a similar nonlinear transformation, applied inversely. Thus, we modify our joint target distribution: we first move deterministically along the PF ODE in reverse (i.e., toward the diffusion direction), and then add noise according to the diffusion SDE to ensure a valid density. As illustrated in Appendix B, this design achieves better alignment between the target and the proposal.

Since our method requires efficient traversal along the PF ODE in both forward and backward directions, BCMs naturally emerge as the ideal model for achieving this bidirectional traversal. Figure 1 provides an overview of this process and we detail the proposal and target in the following:

Proposal Distribution. The proposal distribution is defined as $p(\boldsymbol{x}_{t_N}) \prod_{n=1}^{N} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_{n-1}} | \boldsymbol{x}_{t_n})$ where

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_{n-1}} | \boldsymbol{x}_{t_n}) = \mathcal{N}\left(\boldsymbol{x}_{t_{n-1}}; f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t_n}, t_n \to t_{n-1}^{(\text{prop})}\right), \left(t_{n-1}^2 - \left(t_{n-1}^{(\text{prop})}\right)^2\right) \boldsymbol{I}\right), \tag{2}$$

for $n = 1, \ldots, N$, where $t_{n-1}^{(\text{prop})} < t_{n-1} < t_n$. Here, f_θ represents the BCM mapping $\boldsymbol{x}_{t_n}$ from time t_n to $t_{n-1}^{(\text{prop})}$ along the ODE trajectory. Then, noise is added according to the diffusion SDE to move forward to t_{n-1}. For the last time step, unlike conventional definitions, we fix $t_0^{(\text{prop})}$ at ϵ, and tune t_0 as a hyperparameter. Samples at t_0 are then returned for importance sampling.

Target Distribution. Similarly, the target distribution also uses the ODE and SDE framework. For a sample at time t_n, we first map it forward to $t_n^{(\text{tar})}$ along the PF ODE trajectory. Then, we add noise according to the diffusion SDE to reach t_{n+1}. Denoting the true distribution as π, the target

Table 1: Integral estimates using 100,000 samples, averaged over 5 runs (mean ± std dev). True value from Monte Carlo sampling of target distribution. Bracketed numbers indicate NFE used.

Test $\phi(\boldsymbol{x})$	GMM-40 ($d = 2$)			GMM-40 ($d = 10$)			DW-4 ($d = 8$)		
	True Value	DDPM+IS (100)	BCM+IS (12)	True Value	DDPM+IS (300)	BCM+IS (24)	True Value	DDPM+IS (150)	BCTM+IS (24)
$\log \|\boldsymbol{x}\|_2$	$3.183_{\pm 0.00}$	$3.174_{\pm 0.01}$	$3.186_{\pm 0.01}$	$4.247_{\pm 0.00}$	$4.247_{\pm 0.02}$	$4.258_{\pm 0.02}$	$1.638_{\pm 0.00}$	$1.641_{\pm 0.01}$	$1.639_{\pm 0.00}$
$\log \|\boldsymbol{x}\|_1$	$3.448_{\pm 0.00}$	$3.438_{\pm 0.01}$	$3.451_{\pm 0.01}$	$5.264_{\pm 0.00}$	$5.265_{\pm 0.03}$	$5.275_{\pm 0.02}$	$2.510_{\pm 0.00}$	$2.514_{\pm 0.01}$	$2.511_{\pm 0.01}$
$\cos(\|\boldsymbol{x}\|_2)$	$0.076_{\pm 0.00}$	$0.080_{\pm 0.01}$	$0.078_{\pm 0.02}$	$0.005_{\pm 0.00}$	$-0.045_{\pm 0.03}$	$0.035_{\pm 0.05}$	$0.382_{\pm 0.00}$	$0.397_{\pm 0.03}$	$0.387_{\pm 0.02}$

distribution is defined as $\pi(\boldsymbol{x}_{t_0}) \prod_{n=1}^{N} q_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_n} | \boldsymbol{x}_{t_{n-1}})$, where:

$$q_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_n} | \boldsymbol{x}_{t_{n-1}}) = \mathcal{N}\left(\boldsymbol{x}_{t_n}; f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_{t_{n-1}}, t_{n-1} \to t_{n-1}^{(\text{tar})}\right), \left(t_n^2 - \left(t_{n-1}^{(\text{tar})}\right)^2\right) \boldsymbol{I}\right), \tag{3}$$

for $n = 1, \ldots, N$, where $t_{n-1} \leq t_{n-1}^{(\text{tar})} < t_n$.

Time Step Optimization. Given the forms of the proposal and target distributions, we still need to determine the time steps $t_n, t_n^{(\text{tar})}, t_n^{(\text{prop})}$ for each n (except for $t_0^{(\text{prop})}$ and t_N, which are fixed to ϵ and T, respectively). To tune these parameters, we employ a stochastic gradient-based approach that minimizes the forward KL divergence between the joint target and proposal distributions, estimating the KL divergence using samples from the target distribution. Details on the hyperparameter tuning are included in Appendix C. As mentioned by [12, 11], another possible objective function would be the α-divergence with $\alpha = 2$, which is equivalent to the variance of importance weights. However, in our experiments, we faced optimization issues when using this metric; thus, further investigation is left as future work.

Model Extensions. To apply our approach to equivariant datasets (e.g., molecules), we incorporate EGNNs [17, 4] to achieve E(3)-equivariant DMs and (B)CMs. Specifically, following [24, 4], we maintain equivariance by defining target and proposal distributions on the zero-center-of-gravity linear subspace for particle datasets. However, we observed that standard CM and BCM perform poorly when enforcing equivariance. In contrast, we found CTM [8] with E(3)-equivariance to be more accurate, likely due to its unique parameterization, which can distill the teacher PF ODE more accurately. Therefore, we extend CTMs to Bidirectional CTMs (BCTMs). While our empirical results show that BCM and BCTM do not differ significantly in GMM toy experiments, BCTMs achieve better performance for equivariant potentials. Appendix D provides further details.

4 Experiments

We evaluate our algorithm on synthetic and equivariant n-body system datasets, including a 40-component Gaussian Mixture Model (GMM) in 2D and 10D, and a 4-particle double-well potential (DW-4) in 8D. For GMM targets, we train score-based DMs and BCM/BCTM using MLPs, following [7]'s preconditioning and parameterization methods. For DW-4, we employ equivariant DMs and BCTMs. Due to training challenges with equivariant BCMs, we report only BCTM results for this dataset. Specifically, we perform two evaluations:

ESS Comparison. We report the ESS estimated by samples from the proposal to assess the efficiency of importance sampling. Figure 2 shows the ESS results for our proposed algorithm using both BCM and BCTM, as well as the baseline algorithm. Notably, compared to the baseline, our method reduces the NFEs required to achieve the same level of ESS by about 85%.

Integral Estimation. Only looking at *estimated* ESS can be misleading. When the target is broader than the proposal, the ESS estimated from finite proposal samples can appear high, even though importance sampling (IS) can present significant errors.

Therefore, we also evaluate the estimation of the integral of some specific test functions ϕ. The results are summarized in Table 1. More detailed results are included in Appendix E. As we can see, our algorithm effectively corrects the model error and achieves a similar performance as the baseline algorithm with much fewer NFEs.

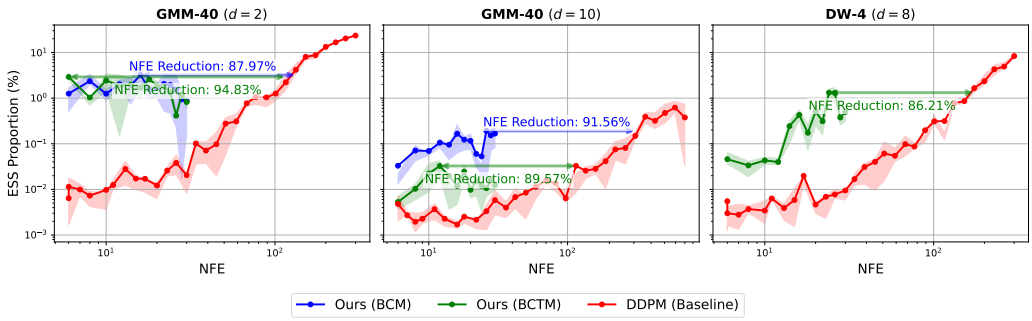

Figure 2: ESS of our proposed algorithm with BCM and BCTM. We also include DDPM as baseline. We estimate ESS with 100,000 samples and report the mean, the first and the third quantile.

5 Conclusions and Limitations

In this work, we propose a sampling algorithm that integrates Importance Sampling (IS) with Consistency Models (CMs), enabling unbiased sampling with only a handful of NFEs. Our method largely outperforms the baseline under limited computational budgets, demonstrating the potential for efficient applications. However, unlike DDPM, which can use more Number of Function Evaluations (NFE) to achieve a higher Effective Sample Size (ESS), our method tends to plateau when the NFE exceeds 10-20. Moreover, we tuned hyperparameters using the forward KL but found it less effective in higher-dimensional spaces. Future work can focus on designing better trade-offs between NFE and performance, as well as identifying more effective hyperparameter tuning metrics.

References

[1] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[2] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[4] Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pages 8867–8887. PMLR, 2022.

[5] Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

[6] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[7] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

[8] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.

[9] Liangchen Li and Jiajun He. Bidirectional consistency models. *arXiv preprint arXiv:2403.18035*, 2024.

[10] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[11] Laurence Illing Midgley, Vincent Stimper, Gregor NC Simm, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Flow annealed importance sampling bootstrap. *arXiv preprint arXiv:2208.01893*, 2022.

[12] Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM Transactions on Graphics (ToG)*, 38(5):1–19, 2019.

[13] Kim A Nicoli, Christopher J Anders, Lena Funcke, Tobias Hartung, Karl Jansen, Pan Kessel, Shinichi Nakajima, and Paolo Stornati. Estimation of thermodynamic observables in lattice field theories with deep generative models. *Physical review letters*, 126(3):032001, 2021.

[14] Kim A Nicoli, Shinichi Nakajima, Nils Strodthoff, Wojciech Samek, Klaus-Robert Müller, and Pan Kessel. Asymptotically unbiased estimation of physical observables with neural samplers. *Physical Review E*, 101(2):023304, 2020.

[15] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

[16] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

[17] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[18] John Skilling. The eigenvalues of mega-dimensional matrices. *Maximum Entropy and Bayesian Methods: Cambridge, England, 1988*, pages 455–466, 1989.

[19] 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.

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

[21] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[22] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[23] 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.

[24] 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.

A Baseline: Combining Importance Sampling with Diffusion Models

We now consider task of estimating the integral $\mathbb{E}_{\boldsymbol{x}_0\sim\pi}[\phi(\boldsymbol{x}_0)]$ where π is the target distribution and ϕ is the test function of interest. We denote the unnormalized target distribution to be $\bar{\pi}$ which we will be able to evaluate. We aim to train a diffusion models to act as the proposal distribution which we can draw samples from. Suppose we have a time interval $[\epsilon, T]$ and we discretize the time interval into N sections with $N+1$ time steps such that $\epsilon = t_0 < t_1 < \cdots < t_N = T$. Assume the conditional proposal distribution of the diffusion model is $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_{n-1}}|\boldsymbol{x}_{t_n})$, and the conditional noise distribution is $q(\boldsymbol{x}_{t_n}|\boldsymbol{x}_{t_{n-1}})$ for $n = 1, \ldots, N$. Using SNIS, the integral can be estimated as

$$\mathbb{E}_{\boldsymbol{x}_0\sim\pi}[\phi(\boldsymbol{x}_0)] \approx \mathbb{E}_{\boldsymbol{x}_{t_0}\sim\pi}[\phi(\boldsymbol{x}_{t_0})] \tag{4}$$

$$= \int \phi(\boldsymbol{x}_{t_0})\pi(\boldsymbol{x}_{t_0})d\boldsymbol{x}_{t_0} \tag{5}$$

$$= \int \phi(\boldsymbol{x}_{t_0})\pi(\boldsymbol{x}_{t_0}) \left(\prod_{n=1}^{N} q(\boldsymbol{x}_{t_n}|\boldsymbol{x}_{t_{n-1}})\right) d\boldsymbol{x}_{t_{0:N}} \tag{6}$$

$$= \int \phi(\boldsymbol{x}_{t_0}) \left(\frac{\pi(\boldsymbol{x}_{t_0})\prod_{n=1}^{N} q(\boldsymbol{x}_{t_n}|\boldsymbol{x}_{t_{n-1}})}{p_{\boldsymbol{\theta}}(\boldsymbol{x}_N)\prod_{n=1}^{N} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_{n-1}}|\boldsymbol{x}_{t_n})}\right) p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_{0:N}})d\boldsymbol{x}_{t_{0:N}} \tag{7}$$

$$\approx \frac{1}{K}\sum_{k=1}^{K} \underbrace{\left(\frac{\pi(\boldsymbol{x}_{t_0}^{(k)})\prod_{n=1}^{N} q(\boldsymbol{x}_{t_n}^{(k)}|\boldsymbol{x}_{t_{n-1}}^{(k)})}{p_{\boldsymbol{\theta}}(\boldsymbol{x}_N^{(k)})\prod_{n=1}^{N} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_{n-1}}^{(k)}|\boldsymbol{x}_{t_n}^{(k)})}\right)}_{w_k} \phi(\boldsymbol{x}_{t_0}) \tag{8}$$

where $\boldsymbol{x}_{t_{0:N}}^{(k)}$ for $k = 1, \ldots, K$ are samples from the joint proposal distribution $p_{\boldsymbol{\theta}}$ and w_k for $k = 1, \ldots, K$ are the importance weights. Note that ϵ is chosen such that the error for the first approximation is negligible.

We now need to specify the exact forms of $q(\boldsymbol{x}_{t_n}|\boldsymbol{x}_{t_{n-1}})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_{n-1}}|\boldsymbol{x}_{t_n})$ as well as how to specify the time steps $\{t_n\}$. For the noising distribution, we will define it according to the forward SDE used by [7], i.e. $q(\boldsymbol{x}_{t_n}|\boldsymbol{x}_{t_{n-1}}) = \mathcal{N}\left(\boldsymbol{x}_{t_n}; \boldsymbol{x}_{t_{n-1}}, \left(t_n^2 - t_{n-1}^2\right)\boldsymbol{I}\right)$. For the denoising distribution, we can define it in a manner similar to DDIM [20] (derived based on the settings used by [7]):

$$q_{\boldsymbol{\sigma}}(\boldsymbol{x}_{t_{n-1}}|\boldsymbol{x}_{t_{n-1}}, \boldsymbol{x}_0) = \mathcal{N}\left(\boldsymbol{x}_{t_{n-1}}; \boldsymbol{x}_{t_n}\sqrt{\frac{t_{n-1}^2 - \sigma_{n-1}^2}{t_n^2}} + \boldsymbol{x}_0\left(1 - \sqrt{\frac{t_{n-1}^2 - \sigma_{n-1}^2}{t_n^2}}\right), \sigma_{n-1}^2\boldsymbol{I}\right) \tag{9}$$

We define σ_{n-1} as $\sigma_{n-1} = \eta\sqrt{\frac{(t_n^2 - t_{n-1}^2)t_{n-1}^2}{t_n^2}}$ where $\eta \in [0, 1]$. When $\eta = 1$, this corresponds to the denoising distribution of DDPM [3], and when $\eta = 0$, the sampling becomes deterministic, corresponding to Euler's first method for solving the PF ODE. Our empirical results indicate that $\eta = 1$ consistently yields the highest ESS and lowest IS variance. Therefore, we use the denoising distribution same to that in DDPM [3].

In our experiments, we adopt the same parameterization, preconditioning, and training methods as introduced by [7]. Consequently, it is natural to consider the time schedule specified by them, with $t_i = \left(\epsilon^{\frac{1}{\rho}} + \frac{i-1}{N-1}\left(T^{\frac{1}{\rho}} - \epsilon^{\frac{1}{\rho}}\right)\right)^{\rho}$ and $\rho = 7$. However, during subsequent experiments, we observed that increasing the value of ρ results in a higher ESS. When $\rho \to \infty$, the time schedule approaches: $t_i = \epsilon\left(\frac{T}{\epsilon}\right)^{\frac{i-1}{N-1}}$. This configuration is equivalent to arranging the time steps evenly in logarithmic space. We found that this approach yields the highest ESS across various target distributions when using the baseline method. Therefore, we employ this time schedule in all subsequent experiments related to the baseline method.

Strengths: The method is grounded in sound theoretical foundations and is straightforward to implement, requiring no modifications to a trained diffusion model. With a sufficient number of steps, it can achieve a high ESS, indicating excellent alignment between the proposal and target distributions. This results in low-variance importance sampling estimates, even in high-dimensional spaces.

Limitations: The primary weakness of the baseline approach lies in the number of steps required to achieve low-variance estimates. Our experiments reveal that even in one-dimensional space, approximately 100 steps may be necessary to obtain reasonable integral estimates. For higher-dimensional spaces, the required number of time steps increases exponentially, quickly becoming computationally infeasible.

B Efficacy of Alternating ODE-SDE Target Distribution Design

To demonstrate the importance of target distribution design in aligning proposal and target distributions for importance sampling, we conducted a comparative study using a simple Gaussian Mixture Model (GMM) with two components as the target distribution. We trained a CTM to learn the target GMM distribution. Setting the number of time steps to three, we optimized the algorithm parameters as detailed in Appendix C for two scenarios: (1) target distribution modeled by the diffusion SDE alone, and (2) our proposed design with target distribution formed by alternating ODE and SDE. We sampled $\boldsymbol{x}_{t_{0:2}}$ from both proposal and target distributions and visualized them pairwise in Figure 3.

Figure 3: Visualization of proposal and target distributions for different target distribution designs.

Figure 3 demonstrates that combining both ODE and SDE leads to better alignment between proposal and target distributions, validating our hypothesis. It's important to note that this example uses only 3 time steps; increasing the number of time steps will lead to even better alignment, as we will show in the experiment section.

C Tuning the Time Steps

Recall that the goal in setting the time steps is to minimize the variance of the IS estimator. Therefore, the key to optimizing our algorithm lies in setting the time steps by minimizing certain objective functions designed to reduce IS variance. To achieve this, we need to introduce a suitable parameterization for the time steps, and we will define three sets of parameters that determine these time steps.

For a time interval $[\epsilon, T]$, we reparameterize the three sets of time points as follows:

$$t_n = \begin{cases} T, & n = N \\ \mu_n(t_{n+1} - \epsilon) + \epsilon, & n = 0, \dots, N-1 \end{cases} \tag{10}$$

$$t_n^{(\text{tar})} = (t_{n+1} - t_n)\eta_n + t_n, \quad n = 0, \dots, N-1 \tag{11}$$

$$t_n^{(\text{prop})} = \begin{cases} \gamma_n(t_n - \epsilon) + \epsilon, & n = 1, \dots, N-1 \\ \epsilon, & n = 0 \end{cases} \tag{12}$$

Note that in Eq. (10), the time steps are defined sequentially, with each time step at n falling between the minimum time point ϵ and the previous time point t_{n+1}. A similar approach is used in Eq. (11) to define the target time steps, where each target time point is positioned between the time points t_n and t_{n+1}. For the proposal time points in Eq. (12), each proposal time point is positioned between t_n and the minimal time point ϵ, except for the last proposal time point $n = 0$, which is fixed to be ϵ.

However, in later experiments, we found that the optimal proposal time points (and t_0) are best defined to ensure that the proposal variance equals the target variance whenever possible:

$$t_n^{(\text{prop})} = \begin{cases} \sqrt{\max\{t_n^2 + \left(t_n^{(\text{tar})}\right)^2 - t_{n+1}^2, \epsilon^2\}}, & n = 1, \dots, N-1 \\ \epsilon, & n = 0 \end{cases} \tag{13}$$

$$t_0 = \min\{\sqrt{\max\{t_1^2 - \left(t_0^{(\text{tar})}\right)^2 + \epsilon^2, \epsilon^2\}}, t_1^{(\text{tar})}\} \tag{14}$$

To ensure that the variance matches, we aim to define the proposal time points such that $t_n^2 - \left(t_{n-1}^{(\text{tar})}\right)^2 = t_{n-1}^2 - \left(t_{n-1}^{(\text{prop})}\right)^2$. Additionally, we enforce the constraint that the proposal time points must be greater than the minimal time point ϵ. The case for $n = 0$ differs slightly because of the way that proposal distribution defined for $n = 0$ (see Figure 1). Here, we apply the constraints that the last time point t_0 lies between the minimal time ϵ and $t_n^{(\text{tar})}$. Notice that $\{\mu_n\}$ and $\{\eta_n\}$ are parameters within the range $[0, 1]$. During optimization, we can apply a sigmoid function to enforce this constraint and optimize the parameters to find the optimal values.

D Extending Consistency Trajectory Models

We first review the training method for Consistency Trajectory Models (CTMs) and then describe how it can be extended to be bidirectional. To train CTMs, we focus on scenarios where a pre-trained diffusion model is available. The training loss for CTMs consists of two components: the soft consistency loss and an auxiliary loss to enhance training performance and facilitate the learning of the student model.

Soft Consistency Loss: Suppose we have an ODE solver, a pre-trained score model $\boldsymbol{\phi}$, a student model $\boldsymbol{G}_\theta$, and a teacher model $\boldsymbol{G}_{\theta^-}$ where $\theta^- = \text{stopgrad}(\mu\theta^- + (1-\mu)\theta)$. Given a fixed time interval $[\epsilon, T]$, we first sample a time point $t \in [\epsilon, T]$ and then sample a time point $s \in [\epsilon, t)$. We then sample another time point $u \in [s, t)$. Next, we sample $\boldsymbol{x}_0 \sim p_{\text{data}}$ from the data distribution and add noise to $\boldsymbol{x}_0$ to obtain $\boldsymbol{x}_t$ following the forward diffusion SDE.

For the teacher model, we use the ODE solver to move from t to u using the pre-trained score model. We then obtain the sample at time s using the teacher model, which is subsequently mapped to time $t_0 = \epsilon$ to obtain $\boldsymbol{x}_{\text{target}}$. The process is summarized as follows:

$$\tilde{\boldsymbol{x}}_u = \text{Solver}(\boldsymbol{x}_t, t, u; \boldsymbol{\phi}) \tag{15}$$

$$\tilde{\boldsymbol{x}}_s = \boldsymbol{G}_{\boldsymbol{\theta}^-}(\tilde{\boldsymbol{x}}_u, u, s) \tag{16}$$

$$\boldsymbol{x}_{\text{target}} = \boldsymbol{G}_{\boldsymbol{\theta}^-}(\tilde{\boldsymbol{x}}_s, s, \epsilon) \tag{17}$$

where $\epsilon \le s \le u < t \le T$.

For the student model, we directly map $\boldsymbol{x}_t$ from time t to time s and then map it again to time ϵ. The process is summarized as follows:

$$\tilde{\boldsymbol{x}}_s = \boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, s) \tag{18}$$

$$\boldsymbol{x}_{\text{est}} = \boldsymbol{G}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_s, s, \epsilon) \tag{19}$$

The loss function is defined as the distance between $\boldsymbol{x}_{\text{target}}$ and $\boldsymbol{x}_{\text{est}}$:

$$\mathcal{L}_{\text{CTM}}(\boldsymbol{\theta};\boldsymbol{\phi}) := \mathbb{E}_{t,s,u,\boldsymbol{x}_0}\mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[d(\boldsymbol{x}_{\text{target}},\boldsymbol{x}_{\text{est}})\right] \tag{20}$$

Auxiliary Losses: In addition to the soft consistency loss, [8] introduce two auxiliary losses to facilitate student learning: the Denoising Score Matching (DSM) loss and an adversarial loss. The DSM loss is used to enforce that $g_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, t)$ should act as a denoiser for any t. The DSM loss is given by:

$$\mathcal{L}_{\text{DSM}}(\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{x}_0,t}\mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\|\boldsymbol{x}_0 - \boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t,t)\|_2^2\right] \tag{21}$$

In our experiments, we found that the adversarial loss did not significantly improve performance (it was mainly used to enhance image quality in the original paper), so we omit the adversarial loss and focus on the CTM and DSM losses.

Extending Consistency Trajectory Models to Bidirectional: In later experiments, we get inspired by Bidirectional Consistency Models (BCMs), which found that it is not necessary to restrict s to be smaller than t. Instead, by relaxing this constraint and only requiring $s \neq t$, the CTM distilled from the pre-trained score model becomes a Bidirectional Consistency Trajectory Model (BCTM). This extension allows us to travel along the PF ODE trajectory not only backward in time but also forward in time using only 1 NFE. We will demonstrate the effectiveness of this extended model in the experiment section.

E More Experiment Results

Table 2 presents additional results for integral estimation tasks. DDPM + MC directly performs integral estimation using MC with samples from a trained DDPM model. The results show clear bias across all three true distributions compared to true values, validating our motivation for unbiased estimation in practical applications. Both the baseline and our proposed algorithm correct this inherent bias through importance sampling. However, our method achieves similar ESS and integral estimation results with significantly fewer Number of NFE than the baseline.

Table 2: Integral estimates using 100,000 samples, averaged over 5 runs (mean $\pm$ std dev). True value from Monte Carlo sampling of target distribution. Bracketed numbers indicate the NFE used.

Task $\mathbb{E}_\pi[\phi(\boldsymbol{x})]$	$\log\|\boldsymbol{x}\|_2$	$\log\|\boldsymbol{x}\|_1$	$\cos(\|\boldsymbol{x}\|_2)$	ESS (%)
GMM-40 ($d=2$)				
True Samples + MC	3.183 ± 0.002	3.448 ± 0.002	0.076 ± 0.002	N/A
DDPM (100) + MC	3.121 ± 0.002	3.381 ± 0.002	0.087 ± 0.003	N/A
DDPM (100) + IS	3.174 ± 0.008	3.438 ± 0.008	0.080 ± 0.013	2.6 ± 1.1
BCTM (12) + IS	3.186 ± 0.009	3.451 ± 0.010	0.078 ± 0.015	2.8 ± 1.3
GMM-40 ($d=10$)				
True Samples + MC	4.247 ± 0.001	5.264 ± 0.001	0.005 ± 0.003	N/A
DDPM (300) + MC	4.149 ± 0.001	5.147 ± 0.001	0.000 ± 0.002	N/A
DDPM (300) + IS	4.247 ± 0.021	5.265 ± 0.027	-0.045 ± 0.025	0.2 ± 0.1
BCTM (24) + IS	4.258 ± 0.017	5.275 ± 0.020	0.035 ± 0.048	0.2 ± 0.1
DW-4 ($d=8$)				
True Samples + MC	1.638 ± 0.000	2.510 ± 0.000	0.382 ± 0.000	N/A
DDPM (150) + MC	1.614 ± 0.000	2.481 ± 0.000	0.283 ± 0.001	N/A
DDPM (150) + IS	1.639 ± 0.004	2.513 ± 0.004	0.389 ± 0.017	1.2 ± 0.2
BCTM (24) + IS	1.640 ± 0.005	2.511 ± 0.006	0.391 ± 0.020	1.2 ± 0.5