One-Step Diffusion Samplers via Self-Distillation and Deterministic Flow

Pascal Jutras-Dubé
Purdue University

Jiaru Zhang
Purdue University

Ziran Wang
Purdue University

Ruqi Zhang
Purdue University

Abstract

Sampling from unnormalized target distributions is a fundamental yet challenging task in machine learning and statistics. Existing sampling algorithms typically require many iterative steps to produce high-quality samples, leading to high computational costs. We introduce *one-step diffusion samplers*, which learn a step-conditioned ODE such that a single large step reproduces the trajectory of many small ones via a state-space consistency loss. We further show that standard ELBO estimates in diffusion samplers degrade in the few-step regime because common discrete integrators yield mismatched forward/backward transition kernels. Motivated by this analysis, we derive a deterministic-flow (DF) importance weight for ELBO estimation without a backward kernel. To calibrate DF, we introduce a volume-consistency regularization that aligns the accumulated volume change along the flow across step resolutions. Our proposed sampler therefore achieves both fast sampling and stable evidence estimate in only one or a few steps. Across challenging synthetic and Bayesian benchmarks, it achieves competitive sample quality with orders-of-magnitude fewer network evaluations while maintaining robust ELBO estimates.

Correspondence to: Pascal Jutras-Dubé (`pjutrasd@purdue.edu`), Jiaru Zhang (`jiaru@purdue.edu`), and Ruqi Zhang (`ruqiz@purdue.edu`). Code is available at: `https://github.com/PascalJD/one-step-diffusion-samplers`

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

1 INTRODUCTION

Sampling from densities of the form

$$p_{\text{target}} = \frac{\rho}{Z}, \quad \text{with} \quad Z = \int_{\mathbb{R}^D} \rho(x)\mathrm{d}x \tag{1}$$

with ρ evaluable pointwise but Z intractable, is a central problem in machine learning (Hernández-Lobato and Adams, 2015; Neal, 1995) and statistics (Andrieu et al., 2003; Neal, 2001), and has applications in scientific fields like physics (Albergo et al., 2019; Noé et al., 2019; Wu et al., 2019), chemistry (Frenkel and Smit, 2002; Holdijk et al., 2024; Hollingsworth and Dror, 2018), and many other fields involving probabilistic models.

Many established sampling algorithms are inherently *iterative*, with the accuracy of the final samples depending heavily on the number of steps. Markov chain Monte Carlo (MCMC) methods converge asymptotically but require long chains in practice (MacKay, 2003; Robert, 1995), while recent diffusion-based samplers (Berner et al., 2024; Vargas et al., 2023; Zhang and Chen, 2022) guarantee finite-time convergence but still rely on hundreds of discretization steps, limiting their use in large models and resource-limited scenarios.

In this paper, we develop Self-Distilled One-Step Diffusion Samplers (OSDS) that preserve sample quality and deliver reliable evidence estimates. Our starting point is self-distillation by state consistency: a step-conditioned control is trained so that a single large probability-flow (PF) ODE step reproduces the composition of many small steps, enabling one/few-step transport in state space.

At the same time, we uncover a critical limitation: state consistency alone does not yield a principled ELBO. In diffusion samplers, ELBOs are typically computed from path-space likelihood ratio (Richter and Berner, 2024; Vargas et al., 2024). In the few-step regime, common discretizations are time-asymmetric, so the likelihood ratio becomes unstable and ELBO estimates collapse, even when samples look accurate.

To address this, we introduce a deterministic-flow (DF) importance weight. Prior samples are pushed forward by the learned PF ODE, and a log-Jacobian is accumulated along the flow; the resulting DF weight yields stable, accurate ELBOs in the one/few-step regime.

OSDS is trained by jointly optimizing: (i) state consistency to match a large step with the composition of small PF steps, and (ii) volume consistency to match the accumulated log-Jacobian across resolutions, ensuring geometric fidelity of the flow. The resulting deterministic map supports fast sampling and evidence estimation with the same small step budget. Our contributions can be summarized as follows:

- We show that RND path-space weights break in the few-step regime because common discrete integrators are time-asymmetric, inducing a forward-backward kernel mismatch and collapsing ELBO.
- We propose Self-Distilled One-Step Diffusion Samplers (OSDS), which learn a step-conditioned shortcut PF ODE via state and volume consistency, and derive a deterministic-flow importance weight from PF ODE change-of-variables that remains stable at one/few steps. To the best of our knowledge, this is the *first* sampler that achieves both high-quality sample generation and accurate statistical estimation in one/few steps.
- Across a wide range of challenging synthetic and Bayesian benchmarks, OSDS delivers competitive sample quality with orders-of-magnitude fewer network evaluations, while maintaining robust evidence estimation in the few-step regime.

2 RELATED WORK

Markov chain Monte Carlo samplers. Classical Monte Carlo samplers construct a Markov chain whose stationary distribution matches the target (Brooks et al., 2012). Prominent examples include Metropolis–Hastings (Hastings, 1970; Metropolis et al., 1953), Gibbs sampling (Geman and Geman, 1984), and overdamped Langevin methods (Parisi, 1981; Rossky et al., 1978). By exploiting geometry, Hamiltonian Monte Carlo improves exploration (Brooks et al., 2012; Duane et al., 1987; MacKay, 2003), and scalability is addressed by stochastic-gradient variants (Welling and Teh, 2011; Chen et al., 2014; Zhang et al., 2020b,a). Despite these advances, MCMC requires many transitions to produce high-quality samples, and convergence is only asymptotically guaranteed.

Normalizing flows. Normalizing flows (NFs) learn invertible maps that transform a simple base distribution into the target. Continuous-time variants (CNFs) evolve samples by integrating an ODE with an iterative solver (Chen et al., 2018b; Grathwohl et al., 2019). In the unnormalized-density setting considered here, flows are typically trained through variational objectives or combined with auxiliary procedures (e.g., annealing or SMC) (Wu et al., 2020; Arbel et al., 2021; Matthews et al., 2022).

Generative flow networks. GFlowNets (GFNs) amortize sampling from intractable distributions by learning policies that construct samples via multi-step trajectories (Bengio et al., 2021, 2023). Recent theory extends GFNs to continuous or hybrid spaces, connecting them to diffusion and flow matching (Lahlou et al., 2023; Zhang et al., 2024; Berner et al., 2025). While GFNs amortize training-time exploration, their inference remains iterative because a trajectory must be generated step by step.

Bridge samplers. Schrödinger bridge (SB) methods cast sampling as an entropic optimal transport problem, learning a stochastic process that bridges a prior to the target; see, e.g., SB samplers (Bernton et al., 2019) and diffusion Schrödinger bridges (DSB) (De Bortoli et al., 2021). Underdamped diffusion bridges (UDB) broaden this view to degenerate noise and exploit favorable convergence/numerical properties of underdamped dynamics (Blessing et al., 2025). Bridge-based samplers learn or estimate dynamics and thus require discrete-stepping SDE integration.

Diffusion-based samplers. A complementary control-theoretic perspective learns drifts that transport an easy prior to the target by solving a (stochastic) optimal control problem (Berner et al., 2024; Richter, 2021; Richter and Berner, 2024; Zhang and Chen, 2022). This viewpoint underlies many recent diffusion samplers (Doucet et al., 2022; Geffner and Domke, 2023; Vargas et al., 2023) and refinements (e.g., resampling (Chen et al., 2025), Hamiltonian couplings (Blessing et al., 2025), particle methods (Phillips et al., 2024), PDE-based evolution (Sun et al., 2024), simulation-free training (Akhound-Sadegh et al., 2024)). While these methods amortize some cost in training, inference still integrates dynamics over many steps.

Acceleration techniques for diffusion models. A parallel line of work reduces the number of reverse-time steps in data-driven diffusion models. Consistency models enable one-step generation by distilling a pre-trained diffusion model (Song et al., 2023) or can be trained from scratch (Song and Dhariwal, 2024; Lu and Song, 2025). Progressive distillation repeatedly halves the number of solver steps while preserving sample

quality (Salimans and Ho, 2022). More recently, shortcut models train a single network to "skip" multiple diffusion steps in one shot (Frans et al., 2025). Orthogonally, solver-level accelerations such as high-order solvers (Jolicoeur-Martineau et al., 2021; Lu et al., 2022; Karras et al., 2022) and straightened transport paths (Liu et al., 2023) permit large step sizes at inference. These accelerators assume access to data from the target distribution, are not designed to incorporate only pointwise evaluations of an unnormalized density, and typically do not provide estimates of the log-partition function.

Summary. Prior samplers rely on multi-step trajectories or time integration at inference. Recent accelerators for diffusion models reduce that step count, but they are data-driven and do not natively handle unnormalized targets or yield $\log Z$ estimates. By contrast, our method fully amortizes exploration during training and produces single-step samples at test time from unnormalized densities given only pointwise access to ρ, while supporting normalization estimates.

3 PRELIMINARIES

3.1 Diffusion Samplers and SDEs

Diffusion samplers aim to draw samples from a complex target density $p_{\text{target}} = \rho/Z$ by transporting them from a simpler prior density p_{prior}. This is modeled as a generative Stochastic Differential Equation (SDE) in $\mathbb{R}^D$ forward in time $t \in [0, T]$:

$$\mathrm{d}\mathbf{x}_t = \mu(\mathbf{x}_t, t)\mathrm{d}t + \sigma(t)\mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_0 \sim p_{\text{prior}}, \tag{2}$$

where $\mu(\mathbf{x}_t, t) \in \mathcal{U} \subset C(\mathbb{R}^D \times [0, T], \mathbb{R}^D)$ is a learnable drift, $\sigma(t) \in C([0, T], \mathbb{R})$ is a predefined diffusion coefficient, and $\mathrm{d}\mathbf{w}_t$ denotes the standard Wiener increments. The core objective is to learn the drift μ such that this generative process becomes the time-reversal of a fixed, linear noising SDE:

$$\mathrm{d}\mathbf{x}_t = \overleftarrow{f}(t)\mathbf{x}_t\mathrm{d}t + \overleftarrow{\sigma}(t)\mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_0 \sim p_{\text{target}}, \tag{3}$$

where $f \in C([0, T], \mathbb{R})$ and $\overleftarrow{f}(t) = f(T - t)$ are predefined. It is known that this time-reversal is achieved if the generative drift satisfies

$$\mu = \sigma\sigma^\top \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t) - f, \tag{4}$$

where $p_t(\mathbf{x}_t)$ is the density of $\mathbf{x}_t$ (Anderson, 1982; Föllmer, 1986; Nelson, 1967).

In practice, the time-marginals p_t are intractable, which means the ideal drift in equation (4) cannot be computed directly. Instead, the intractable scaled score term $\sigma\nabla_{\mathbf{x}} \log p_t$ is parameterized by a neural network expressed as a control function $u_\theta(x, t)$. It is trained by minimizing a divergence $D(\mathbb{P}|\overleftarrow{\mathbb{P}})$ between the path measures of the generative process ($\mathbb{P}$) and the noising process ($\overleftarrow{\mathbb{P}}$). This requires computing the likelihood ratio between the two continuous-time path measures, which is given by the following proposition, with the proof deferred to Appendix B.1. See also (Vargas et al., 2024, Proposition 2.2).

Proposition 1 (Forward-backward Radon-Nikodym derivative (RND))**.** *Let $\mathbb{P}$ be the path measure of the forward SDE in equation* (2) *with control u, and let $\overleftarrow{\mathbb{P}}$ be the path measure of the reverse-time SDE in equation* (3) *with terminal density p_{target}.*

Then for any path $\mathbf{x}$ we have

$$\log \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\overleftarrow{\mathbb{P}}}(\mathbf{x}) = \frac{1}{2}\int_0^T \left(\|\sigma f\|^2 - \|\sigma f + u_\theta\|^2\right)(\mathbf{x}_t, t)\mathrm{d}t + \int_0^T \left(\frac{u_\theta}{\sigma}\right)(\mathbf{x}_t, t) \cdot \mathrm{d}\mathbf{x}_t + \log \frac{p_{\text{prior}}(\mathbf{x}_0)}{p_{\text{target}}(\mathbf{x}_T)}. \tag{5}$$

Once trained, the optimized control u_θ allows generation of samples from p_{target} through forward simulations of equation (2). In practice, this continuous-time process must be discretized into finite steps $0 = t_1 < t_2 < \cdots < t_N = T$, introducing a trade-off between computational cost and accuracy.

3.2 The FB-RND Framework for Evidence Estimation

Another key task in diffusion samplers beyond sample generation is estimating the model evidence, $\log Z$, which is crucial for model comparison and selection. Diffusion samplers are able to provide this estimate. The theoretical foundation for this capability is the discrete Forward-Backward Radon-Nikodym Derivative (FB-RND) framework, which provides a valid Evidence Lower Bound (ELBO) on $\log Z$. This is formally stated in the following proposition:

Proposition 2 (Finite-step forward-backward bound)**.** *For every discrete trajectory $\mathbf{x}_{0:N}$ simulated under the forward discretization of equation* (2)*,*

$$\log \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\overleftarrow{\mathbb{P}}}(\mathbf{x}_{0:N}) = \log \frac{p_{\text{prior}}(\mathbf{x}_0)}{p_{\text{target}}(\mathbf{x}_N)} + \sum_{n=0}^{N-1} \log \frac{p_{n+1|n}(\mathbf{x}_{n+1}|\mathbf{x}_n)}{p_{n|n+1}(\mathbf{x}_n|\mathbf{x}_{n+1})}. \tag{6}$$

Taking expectation with respect to the forward discrete

path measure yields the lower bound

$$\mathbb{E}_{\mathbb{P}}\Big[\log \rho(\mathbf{x}_N) - \log p_{\text{prior}}(\mathbf{x}_0) + \sum_{n=0}^{N-1} \log \frac{p_{n+1|n}(\mathbf{x}_{n+1}|\mathbf{x}_n)}{p_{n|n+1}(\mathbf{x}_n|\mathbf{x}_{n+1})}\Big] \leq \log Z. \tag{7}$$

See Appendix B.2 for a proof and a Monte Carlo estimator of $\log Z$*, and Algorithm 3 for pseudocode.*

The bound shown by proposition 2 provides an effective evaluation approach of $\log Z$ in diffusion samplers.

4 ONE-STEP TRANSPORT AND EVIDENCE ESTIMATION

This section introduces Self-Distilled One-Step Diffusion Samplers (OSDS), a framework designed to achieve two complementary goals: (i) generating high-fidelity samples in a single or very few steps, and (ii) reliably estimating the partition function $\log Z$ with the same few-step budget. The core idea is to self-distill a diffusion sampler into a deterministic flow map induced by its PF ODE. OSDS is trained using two complementary consistency terms: a state-space self-distillation loss, which enforces that the transported states match across resolutions, and a volume consistency loss, which enforces that the log-volume change (log-determinant of the Jacobian) accumulated by a large step matches the sum of the composition of small steps. At test time, the resulting flow map is used for both single/few-step sampling and estimation of $\log Z$ via change of variables.

4.1 Learning One-Step Transport via State Consistency

Our self-distillation framework relies on a recursive consistency principle: a large step should behave like the composition of smaller steps. To anchor this recursion, we need a reliable small-step sampler. In our setting there is no dataset to reveal the modes of p_{target}; the sampler must discover them. A diffusion sampler does exactly this: its Brownian motion visits high-density regions, leading to coverage that a purely deterministic flow (or a likelihood-trained normalizing flow without data) would struggle to obtain.

RND base loss at small steps. At a base resolution $d_0 = T/N_0$, we train $u_\theta(x, t, d_0)$ as a diffusion sampler using the KL objective

$$\mathcal{L}_{\text{RND}} = \mathbb{E}\Big[\log \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\bar{\mathbb{P}}}\Big]. \tag{8}$$

where the likelihood ratio is given by the RND proposition 1. This ensures the model learns a valid diffusion sampler at fine resolution.

Self-distillation to larger steps. Let $\Psi_\theta : \mathbb{R}^D \times [0, T] \times (0, T] \to \mathbb{R}^D$ denote a generic numerical solver that advances the state of the PF ODE of the generative SDE equation (2):

$$\mathrm{d}\mathbf{x}_t = \big(\tfrac{1}{2}\sigma u_\theta - f\big)(\mathbf{x}_t, t, d)\mathrm{d}t, \quad \mathbf{x}_0 \sim p_{\text{prior}}, \tag{9}$$

from time t to $t{+}d$: $\widehat{x}_{t+d} = \Psi_\theta(x_t; t, d)$.

We form a teacher–student pair: the teacher is the composition of two half-steps under frozen parameters $\theta' = \text{stopgrad}(\theta)$,

$$x_{t+\frac{d}{2}} = \Psi_{\theta'}(x_t; t, \tfrac{d}{2}), \qquad x_{t+d} = \Psi_{\theta'}(x_{t+\frac{d}{2}}; t + \tfrac{d}{2}, \tfrac{d}{2}),$$

and the student is the single large step $\widehat{x}_{t+d} = \Psi_\theta(x_t; t, d)$. The self-distillation loss enforces

$$\mathcal{L}_{\text{state}} = \mathbb{E}\big[\|\widehat{x}_{t+d} - x_{t+d}\|^2\big] \tag{10}$$

with (t, d) sampled along powers of two from the training trajectories.

By optimizing $\mathcal{L}_{\text{RND}}$ and $\mathcal{L}_{\text{state}}$ jointly, the model learns to explore through stochastic diffusion steps and to compress this exploration into accurate large-step deterministic updates. Figure 1 illustrates the self-distillation procedure along the PF ODE.

4.2 Why Path-Space ELBO Degrades at Few Steps

The validity of the bound proposed in proposition 2 hinges on the use of a true time-reversed kernel, known as the *time-adjoint* kernel $p^\star_{n|n+1}$. Let $\widetilde{p}_{n|n+1}(\cdot \mid \mathbf{x}_{n+1})$ be any surrogate backward kernel with the same support as $p^\star_{n|n+1}$ (e.g., obtained by applying Euler–Maruyama to the reverse-time SDE). Replacing the time-adjoint kernel by $\widetilde{p}_{n|n+1}$ introduces an entropy gap in the per-step forward–backward term:

$$\begin{aligned}&\mathbb{E}\big[\log p_{n+1|n}(\mathbf{x}_{n+1}|\mathbf{x}_n) - \log \widetilde{p}_{n|n+1}(\mathbf{x}_n|\mathbf{x}_{n+1})\big] \\ &= -\,\mathbb{E}_{\mathbf{x}_{n+1}}\Big[\mathrm{KL}\big(p^\star_{n|n+1}(\cdot|\mathbf{x}_{n+1}) \| \widetilde{p}_{n|n+1}(\cdot|\mathbf{x}_{n+1})\big)\Big] \leq 0.\end{aligned} \tag{11}$$

While this gap is negligible for small step sizes, it becomes dominant in the few-step regime. For example, at $N = 1$ this single KL term can be very large because the forward EM mean and variance scale with $\beta(t_1)$ at the right endpoint and with the control u, whereas a surrogate backward that ignores these choices concentrates elsewhere. It catastrophically inflates the variance of the importance weights and collapses the ELBO estimates.

To further illustrate the severity of this failure, we provide a numerical example and more details in section B.3. For a simple 1D VP SDE with a single

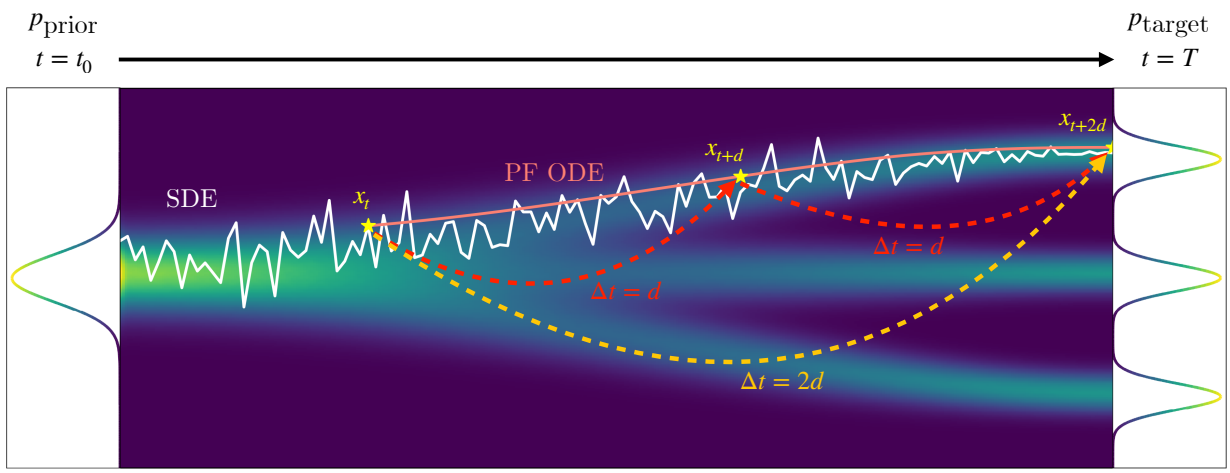

Figure 1: Graphical illustration of state-space self-distillation.

discretization step, we explicitly calculate and compare the true backward kernel with the surrogate "EM backward" kernel. The results show that their variances differ by nearly 50-fold. This discrepancy translates to a per-step cross-entropy gap of approximately -40 nats, providing clear evidence for the collapse of the ELBO estimator.

Therefore, to achieve both fast sampling and reliable model evaluation, an alternative route to $\log Z$ that does not rely on a fragile backward Markov kernel is required.

4.3 Deterministic-Flow Importance Weights

To keep the estimator well-conditioned at one or a few steps, we complement the state-consistency with a deterministic importance sampler built from the PF ODE of the learned dynamics. The key idea is to push forward prior samples through the single (or few) large PF step and evaluate a change-of-variables weight.

Fix a step size $d > 0$ (e.g., $d = T$ for a full-horizon transport). For $t \in [0, d]$, consider the PF ODE

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = b_\theta(\mathbf{x}_t, t), \;\; b_\theta(x,t) := \big(\tfrac{1}{2}\sigma u_\theta - f\big)(x,t,d), \quad (12)$$

and let $\phi_t : \mathbb{R}^D \to \mathbb{R}^D$ denote its flow map, i.e., $\phi_0(x) = x$ and $\frac{\mathrm{d}}{\mathrm{d}t}\phi_t(x) = b_\theta(\phi_t(x), t)$. The single-step deterministic transport is

$$T(x) := \phi_d(x).$$

We assume b is sufficiently regular so that $t \mapsto \phi_t$ is a diffeomorphic flow on $[0, d]$, see (Chen et al., 2018a, Section 6). We will use $x_0 \sim p_{\text{prior}}$ and set $y = T(x_0) = \phi_d(x_0)$.

Proposition 3 (Deterministic-flow IS weight, section C.1)**.** *Let $q := T_{\#}p_{\text{prior}}$ be the push-forward of the prior by the flow map $T = \phi_d$. Then for any $x_0 \in \mathbb{R}^D$,*

$$w(x_0) = \frac{\rho\big(T(x_0)\big)}{q\big(T(x_0)\big)} = \rho\big(\phi_d(x_0)\big)\frac{\big|\det \nabla T(x_0)\big|}{p_{\text{prior}}(x_0)}. \quad (13)$$

Moreover, $Z = \mathbb{E}_{x_0 \sim p_{\text{prior}}}\big[w(x_0)\big]$ so

$$\widehat{Z} = \frac{1}{M}\sum_{i=1}^{M} w\big(x_0^{(i)}\big) \;\; \textit{is an unbiased estimator of } Z. \quad (14)$$

For numerical stability we evaluate

$$\log w(x_0) = \log \rho(y) + \log\big|\det \nabla T(x_0)\big| - \log p_{\text{prior}}(x_0). \quad (15)$$

Computing $\log|\det \nabla T|$ via the PF ODE. Directly forming ∇T is expensive in high dimensions. For ODE flows one can compute the log-Jacobian determinant through the instantaneous change of variables identity (Chen et al., 2018a):

$$\frac{\mathrm{d}}{\mathrm{d}t}\log\big|\det \nabla\phi_t(x_0)\big| = \nabla\cdot b_\theta\big(\mathbf{x}_t, t\big) \;\; \text{along } \mathbf{x}_t = \phi_t(x_0). \quad (16)$$

Integrating equation (16) from 0 to d gives

$$\log\big|\det \nabla T(x_0)\big| = \int_0^d \nabla\cdot b_\theta\big(\mathbf{x}_t, t\big)\mathrm{d}t, \quad \mathbf{x}_t = \phi_t(x_0). \quad (17)$$

Hence one can evolve, in tandem with $\mathbf{x}_t$, a scalar volume accumulator ℓ_t solving the ODE

$$\dot{\ell}_t = \nabla\cdot b_\theta(\mathbf{x}_t, t), \quad \ell_0 = 0, \quad \Rightarrow \quad \log\big|\det \nabla T(x_0)\big| = \ell_d. \quad (18)$$

Practical computation. Combining equation (15) and equation (18), the single-sample log-weight is

$$\log w(x_0) = \log \rho\big(\mathbf{x}_d\big) + \ell_d - \log p_{\text{prior}}(x_0), \quad (19)$$

with $\dot{\mathbf{x}}_t = b_\theta(\mathbf{x}_t, t)$, $\mathbf{x}_0 = x_0$ and $\dot{\ell}_t = \nabla \cdot b_\theta(\mathbf{x}_t, t)$, $\ell_0 = 0$. In high dimensions the divergence $\nabla \cdot b$ can be obtained without forming Jacobians explicitly using standard stochastic trace estimators (e.g., Hutchinson, see section C.2 and (Grathwohl et al., 2019)). The ODE pair $(\mathbf{x}_t, \ell_t)$ can then be integrated with any accurate solver; the resulting $w(x_0)$ is plugged into equation (14).

Unlike the discrete forward/backward likelihood ratio equation (6), the deterministic weight does not require a reverse one-step kernel; it only needs the forward PF-ODE flow's Jacobian determinant, for which continuous-time change-of-variables is exact.

Volume consistency for accurate log-Jacobians. The accuracy of this deterministic estimator hinges on the reliability of the computed log-Jacobian. The RND base loss $\mathcal{L}_{RND}$ and the self-distillation loss $\mathcal{L}_{\text{state}}$ do not calibrate how densities transform, with $\mathcal{L}_{\text{state}}$ being a simple MSE on positions and insensitive to the geometric properties of the map. Therefore, two maps can land at the same state yet induce different local volume changes. To ensure the learned flow is geometrically consistent across different step sizes, we introduce an additional Volume Consistency Loss $\mathcal{L}_{\text{vol}}$.

Based on the principle that log-volumes are additive under composition, we extend the teacher-student framework to the log-Jacobian determinants. The teacher's accumulated log-volume is the sum of two half-step volumes:

$$\text{vol}_{\text{teacher}} = \text{vol}\big(\Psi_{\theta'}(x_t, t, \tfrac{d}{2})\big) + \text{vol}\big(\Psi_{\theta'}(x_{t+\frac{d}{2}}, t+\tfrac{d}{2}, \tfrac{d}{2})\big), \tag{20}$$

where $\text{vol}(\cdot)$ denotes the log-Jacobian of a map, computed via equation (17). The student's log-volume, $\text{vol}_{\text{student}} = \text{vol}(\Psi_\theta(x_t; t, d))$, is then trained to match the teacher's via a mean-squared error objective:

$$\mathcal{L}_{\text{vol}} = \mathbb{E}\Big[\big(\text{vol}_{\text{student}} - \text{vol}_{\text{teacher}}\big)^2\Big]. \tag{21}$$

This loss acts as a regularizer, penalizing mismatches in the accumulated log-Jacobian. This discourages exploding or vanishing local volumes, improves the numerical stability of the PF ODE integration, and yields better-conditioned deterministic weights for the estimator.

4.4 The Final OSDS Framework

By jointly optimizing all three objectives in our self-distillation framework, we present our final Self-Distilled One-Step Diffusion Sampler (OSDS) framework. It is trained by jointly minimizing a single, composite objective:

$$\mathcal{L}_{\text{OSDS}} = \mathcal{L}_{\text{RND}} + \mathcal{L}_{\text{state}} + \lambda_{\text{vol}}\mathcal{L}_{\text{vol}}, \tag{22}$$

Algorithm 1 OSDS Training

Require: densities ρ, p_{prior}; horizon T, SDE coeffs $f(t), \sigma(t)$; base steps N_0 ($d_0 = T/N_0$); step-conditioned control u_θ; weight λ_{vol}; iterations I; batch size B.

Ensure: trained parameters $\theta^\star$.

```
Initialize θ
for i = 1 to I do
    Draw {x_0^(b)}_{b=1}^B ~ p_prior
    Simulate paths {x_{0:N}^(b)}_{b=1}^B and compute L_RND via algorithm 3
    Sample anchors (x_t^(b), t) from paths and steps {d^(b)}_{b=1}^B
    Compute L_state and L_vol via algorithm 4.
    L_OSDS ← L_RND + L_state + λ_vol L_vol
    Update θ by SGD/Adam.
end for
```

where $\lambda_{\text{vol}} > 0$ is a weighting coefficient. This simultaneous optimization creates a powerful synergy: the $\mathcal{L}_{\text{RND}}$ term provides a stable, exploratory foundation; $\mathcal{L}_{\text{state}}$ distills this exploration into a spatially accurate shortcut; and $\mathcal{L}_{\text{vol}}$ ensures this shortcut is also geometrically consistent. A pseudocode summarizing the training procedure of OSDS is given by algorithm 1.

This principled training scheme produces a high-quality PF ODE flow map. At inference time, this allows for both rapid sample generation by evaluating the map, and stable estimation of $\log Z$ by using the deterministic-flow importance weights. Thus, the dual goals of speed and statistical reliability are simultaneously achieved. The sampling procedure is described in algorithm 2.

5 EXPERIMENTS

5.1 Setup

We evaluate the performance of OSDS on a diverse set of benchmarks. For synthetic targets, we consider the Funnel distribution, the 32-mode Many-Well distribution, and a Gaussian mixture model with 40 random modes. For real-world Bayesian inference, we include six standard benchmarks: CREDIT, SEEDS, CANCER, BROWNIAN, IONOSPHERE, and SONAR. The code is available at this GitHub repository.

Importance weights. All evaluation criteria are functions of unnormalized importance weights. For each sampler, step budget, and run we draw m trajectories and form a set of weights $\{w^{(i)}\}_{i=1}^m$ that compare the target ρ to the density induced by the sampler. The weight $w^{(i)}$ is either the discrete forward–backward (RND) path weight (proposition 2) or

Algorithm 2 OSDS Inference: single/few–step sampling and $\log Z$

Require: trained u_θ; densities ρ, p_{prior}; horizon T; test steps $N_{\text{test}} \in \{1, 2, 4, \ldots\}$; batch size M; solver Ψ.
Ensure: samples $\{y^{(i)}\}$; weights $\{w^{(i)}\}$; $\widehat{Z}$; ELBO.

Set step size $d \leftarrow T/N_{\text{test}}$
Set PF ODE $b_\theta \leftarrow (\frac{1}{2}\sigma u_\theta - f)$
for $i = 1$ **to** M **do**
  $x \sim p_{\text{prior}}$; $t \leftarrow 0$; $x_0 \leftarrow x$; logdet $\leftarrow 0$
  for $j = 1$ **to** N_{test} **do**
    Integrate PF ODE $x_t \leftarrow \Psi_\theta(x_t, t, d)$
    Accumulate $\ell \leftarrow \int_t^{t+d} \nabla \cdot b_\theta(x_t, \tau, d)\mathrm{d}\tau$
    logdet $\leftarrow$ logdet $+ \ell$; $t \leftarrow t+d$.
  end for
  $y^{(i)} \leftarrow x_t$
  $\log w^{(i)} \leftarrow \log \rho(y^{(i)}) + \text{logdet} - \log p_{\text{prior}}(x_0)$
  $w^{(i)} \leftarrow e^{\log w^{(i)}}$
end for
$\widehat{Z} \leftarrow \frac{1}{M}\sum_i w^{(i)}$; $\text{ELBO}_{\text{det}} \leftarrow \frac{1}{M}\sum_i \log w^{(i)}$.

the deterministic-flow (DF) change-of-variables weight (proposition 3).

Sinkhorn distance. To quantify sample quality when we have access to target samples, we report the entropy-regularized 2-Wasserstein (Sinkhorn) distance (Cuturi, 2013) between the empirical distribution of model samples and target samples. Lower values indicate closer agreement between the generated and target samples.

ELBO (reverse evidence lower bound). Following Blessing et al. (2024), we treat the sampler as an importance sampler with weights $w^{(i)}$ and define the evidence lower bound

$$\text{ELBO} := \mathbb{E}[\log w] \approx \frac{1}{m}\sum_{i=1}^{m} \log w^{(i)}.$$

This quantity lower-bounds $\log Z$ and is high when the importance weights are well calibrated and low-variance.

EUBO (forward evidence upper bound). When we can draw samples from the normalized target p_{target}, we can also report the forward evidence upper bound (EUBO) (Blessing et al., 2024). Let q denote the (unknown) model density and $w_{\text{fwd}}(x) = \rho(x)/q(x)$ the corresponding forward importance weight. Then

$$\begin{aligned}\text{EUBO} &:= \mathbb{E}_{x\sim p_{\text{target}}}\big[\log w_{\text{fwd}}(x)\big] \\ &= \log Z + D_{\text{KL}}\big(p_{\text{target}} \| q\big) \geq \log Z.\end{aligned}$$

Smaller EUBO therefore indicates better mode coverage of the target. In practice we approximate this expectation by Monte Carlo using the same sampler-induced weights.

Effective sample size (ESS). To assess how concentrated the weights are, we compute the normalized (reverse) effective sample size

$$\text{ESS} := \frac{\left(\sum_{i=1}^m w^{(i)}\right)^2}{m\sum_{i=1}^m \left(w^{(i)}\right)^2} \in (0, 1].$$

Values near 1 indicate nearly uniform weights (low variance), while values near 0 signal weight degeneracy.

Partition function error $|\Delta \log Z|$. Whenever the true partition function Z is known (for synthetic targets), we form the standard importance-weighted estimator

$$\widehat{Z} = \frac{1}{m}\sum_{i=1}^{m} w^{(i)}, \qquad \widehat{\log Z} = \log \widehat{Z},$$

and report the absolute error

$$\big|\Delta \log Z\big| = \big|\log Z - \widehat{\log Z}\big|.$$

Baselines and protocols. We compare against SMC (Del Moral et al., 2006), CRAFT (Matthews et al., 2022), CMCD (Vargas et al., 2024), PIS (Zhang and Chen, 2022), and DDS (Vargas et al., 2023).

For fairness, PIS, DDS, DIS, and our OSDS all share the same PIS-GRAD backbone (Zhang and Chen, 2022), with OSDS further including a small step-size embedding. Each method is evaluated over 20 independent runs, with 2,000 samples per run; we report mean ± standard deviation across runs. All samplers use the same right-endpoint Euler-Maruyama discretization. Further implementation details are deferred to section E.1.

5.2 Sample Quality

Sinkhorn distance. Table 1 summarizes Sinkhorn distance between model samples and the target on two synthetic benchmarks. Even with a single step, OSDS attains competitive distances: on Funnel it is within ∼0.8% of PIS while using 1/128 of the function evaluations; on MW OSDS is close in absolute terms to the best diffusion samplers (0.39 vs. 0.14 for PIS/DDS). These results indicate that our PF flow map preserves much of the sample quality typically associated with iterative diffusion/bridge samplers, despite using dramatically fewer steps.

Mode coverage. Figure 2 visualizes OSDS's single-step samples overlaid on energy contours for a Gaussian mixture model (GMM) with 40 random modes in

Table 1: Sinkhorn distance on synthetic targets. OSDS attains competitive sample quality in a single step.

Sinkhorn (↓)	NFE	Funnel (10D)	MW (5D)
SMC	128	149.35±4.73	20.71±5.33
CRAFT	128	133.42±1.04	11.47±0.90
CMCD	128	124.89±8.95	0.57±0.05
PIS	128	159.13±0.34	0.14±0.00
DDS	128	147.72±0.90	0.14±0.00
OSDS (ours)	1	160.36 ±0.64	0.39 ±0.00

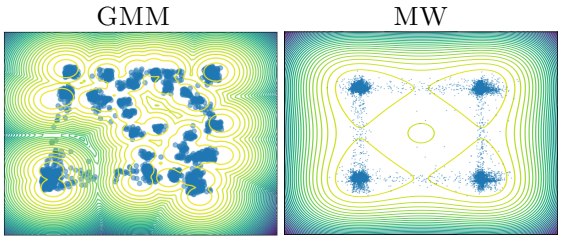

Figure 2: Single-step OSDS samples (blue) overlaid on target energy contours. Samples populate distinct high-density regions, illustrating broad mode coverage in one step.

two dimensions and a five-dimensional 32-mode ManyWell (MW). On GMM, mass is spread across the mixture components, while on MW the samples occupy all high-probability wells. These qualitative results mirror the Sinkhorn findings: OSDS maintains coverage of modes in one step, aligning samples with the high-density regions of the target energy.

Trading compute for sample quality. While our main comparisons focus on the single-step regime, OSDS also exposes a knob to trade additional NFEs for tighter inference. On the 40-mode GMM in Table 3, increasing the PF-ODE step budget improves the deterministic-flow (DF) weights and recovers the accuracy of the RND estimator in the small step regime.

Amortized inference cost. Although OSDS introduces a small distillation overhead during training, this cost is amortized at inference. In Appendix D, we derive an explicit NFE accounting that compares a baseline N-step diffusion sampler with OSDS. Our analysis shows that for moderate or large sampling workloads, the one-time distillation cost is quickly recovered, yielding substantial end-to-end computational savings.

5.3 Deterministic-Flow Importance Weights

We now evaluate the proposed deterministic-flow (DF) importance weights and compare them against the standard discrete forward-backward path weights (RND). We consider six Bayesian benchmarks from (Blessing et al., 2024) and the strongly multimodal 40-mode GMM.

Bayesian benchmarks. Table 2 reports ELBOs obtained from DF and RND weights for OSDS at two discretization budgets: a single PF-ODE step and a fine discretization, alongside PIS and DDS baselines.

In the *single-step* regime, DF remains numerically stable on all six datasets, yielding finite, reasonably tight ELBOs. In contrast, the RND estimator collapses: ELBO values are several orders of magnitude lower, reflecting the forward-backward kernel mismatch at coarse resolution. This matches our analysis in section B.3: when the discretization is too coarse, the surrogate backward kernel is no longer time-adjoint to the forward kernel, and path-space likelihood ratios become ill-conditioned, whereas DF sidesteps the backward kernel entirely.

In the *multi-step* regime, DF continues to improve and often surpasses diffusion baselines. OSDS (DF, 128) achieves the best ELBO on CREDIT, SEEDS, IONOSPHERE, and SONAR, and remains competitive on CANCER and BROWNIAN. At the same step budget, RND consistently yields lower ELBOs than DF, indicating higher-variance weights even when the discretization is fine. This suggests that enforcing geometric consistency via the PF ODE and volume tracking leads to better-conditioned importance weights and tighter evidence estimates.

Multimodal 40-mode GMM. To further stress-test the importance weights in a strongly multimodal setting, we consider the 40-mode Gaussian mixture model and compare DF and RND across a range of step budgets. Table 3 reports ELBO, EUBO, ESS, and $|\Delta \log Z|$ as a function of NFE for both estimators.

In the *few-step* regime (NFE $= 1, 2, 4, 8$), the RND estimator exhibits the breakdown predicted by our theory. ELBO values are extremely negative, the ESS is essentially zero, and $|\Delta \log Z|$ is large. By contrast, DF remains well behaved under exactly the same discretizations: ELBOs are moderate and improve rapidly with NFE, EUBO stays close to the ground-truth scale, ESS is consistently non-degenerate, and $|\Delta \log Z|$ is already below 1 for all NFE in this regime. This confirms that DF provides stable and accurate importance weights precisely where path-space RND weights become unreliable.

Table 2: ELBO from importance weights on six Bayesian benchmarks. DF denotes our proposed deterministic-flow change-of-variables weights (PF ODE) while RND denotes the discrete stochastic forward-backward path weights.

ELBO (↑)	Weight	NFE	Credit (25D)	Seeds (26D)	Cancer (31D)	Brown. (32D)	Iono.(35D)	Sonar (61D)
OSDS (ours)	DF	1	-777.22±8.64	-89.97±1.00	-2869±90	-11.37±0.12	-232.70±1.00	-304.40±3.00
OSDS (ours)	RND	1	-20758.59±540.69	-9892±2.7e+03	-10640±3e+02	-278.51±2.91	-1302±8.9	-2010±18.00
OSDS (ours)	DF	128	**-501.74**±0.53	**-44.35**±0.56	8.66±0.19	-11.58±0.13	**-86.68**±0.72	**-50.58**±0.56
OSDS (ours)	RND	128	-519.73±0.19	-79.74±0.10	2.90±0.16	-32.62±0.14	-124.83±0.24	-139.70±0.28
PIS	RND	128	-846.57±2.42	-88.92±2.05	**39.54**±5.30	NA	-125.03±0.69	-142.868±3.29
DDS	RND	128	-514.74±1.22	-75.21±0.21	20.00±0.690	**0.51**±0.23	-114.19±0.11	-121.22±5.99

Table 3: Comparison of our proposed deterministic-flow (DF) and discrete forward–backward (RND) importance weights on the 40-mode Gaussian mixture model across different step budgets (NFE).

	ELBO (↑)		EUBO (↓)		ESS (↑)		$\lvert\Delta \log Z\rvert$ (↓)	
NFE	DF	RND	DF	RND	DF	RND	DF	RND
1	-9.69±0.41	-11532.86±362.64	1.66±0.02	17.36±0.40	0.05±0.01	0.00±0.00	0.31±0.36	4.08±0.57
2	-3.36±0.20	-2646.03±84.56	1.78±0.02	5.608±0.07	0.13±0.03	0.00±0.00	0.70±0.07	2.57±0.50
4	-3.45±0.27	-66.89±3.22	2.08±0.02	3.28±0.04	0.41±0.03	0.02±0.01	0.97±0.03	1.47±0.38
8	1.39±0.16	-19.72±0.59	2.00±0.02	3.00±0.05	0.36±0.08	0.03±0.01	0.75±0.04	1.00±0.36
16	2.31±0.05	-5.43±0.13	2.05±0.03	2.18±0.04	0.61±0.03	0.06±0.02	0.99±0.02	0.32±0.22
32	2.64±0.04	-2.09±0.06	2.15±0.02	1.45±0.03	0.65±0.04	0.10±0.04	0.78±0.02	0.11±0.09
64	3.10±0.04	-1.98±0.06	2.91±0.02	1.83±0.06	0.64±0.01	0.17±0.05	0.77±0.05	0.05±0.05
128	3.59±0.29	-2.92±0.14	2.07±0.02	2.94±0.05	0.66±0.01	0.23±0.06	0.73±0.02	0.01±0.09

As the number of steps increases, the RND estimator gradually recovers. For NFE ≥ 16, its $|\Delta \log Z|$ becomes very small, reflecting the asymptotic correctness of the discrete forward-backward likelihood ratio when the integrator is sufficiently fine. However, even in this multi-step regime, RND weights remain high-variance: ESS is consistently lower than for DF, and the resulting ELBOs are also lower. DF, on the other hand, achieves strong ELBOs and ESS while maintaining $|\Delta \log Z|$ below 1 over the entire NFE range. Overall, these results show that DF importance weights are robust in the one/few-step regime, where discrete forward-backward estimators can catastrophically fail, and remain competitive in the multi-step regime.

5.4 Ablation on the volume-consistency weight

To assess the influence of the volume-consistency loss $\mathcal{L}_{\text{vol}}$, we ablate its weight λ_{vol} on one synthetic target (40-mode GMM) and one Bayesian benchmark (CREDIT). Table 4 reports $\lambda_{\text{vol}} = 0$ (no volume consistency) and $\lambda_{\text{vol}} = 0.25$. In all cases, adding a small amount of volume regularization improves the ELBO, with larger gains in the challenging one-step regime.

Table 4: Ablation on the volume-consistency weight λ_{vol} using deterministic-flow (DF) ELBO on the 40-mode GMM and the CREDIT Bayesian benchmark.

Task	NFE	λ_{vol}	ELBO (↑)
GMM	1	0	−10.74±0.41
GMM	1	0.25	**−9.69**±0.30
GMM	128	0	2.19±0.06
GMM	128	0.25	**3.59**±0.29
CREDIT	1	0	−1555.76±15.81
CREDIT	1	0.25	**−777.22**±8.64
CREDIT	128	0	−509.32±0.49
CREDIT	128	0.25	**−501.74**±0.53

CONCLUSION

We introduced Self-Distilled One-Step Diffusion Samplers (OSDS), a framework that reconciles fast inference with statistically sound evidence estimation for sampling from unnormalized densities. Our analysis identified a root cause of ELBO collapse in the few-step regime, the time-asymmetry of standard discrete integrators, which breaks the forward-backward likelihood ratio. OSDS addresses this by distilling many small probability-flow steps into a geometrically consistent shortcut map, enforced by state and volume consistency losses, and by estimating evidence via a deterministic change-of-variables weight that bypasses the fragile backward kernel. Empirically, OSDS delivers stable single-step ELBOs and strong sample quality.

ACKNOWLEDGEMENTS

This research is supported in part by NSF IIS-2508145 and Amazon Research Award.

References

Akhound-Sadegh, T., Rector-Brooks, J., Bose, A. J., Mittal, S., Lemos, P., Liu, C.-H., Sendera, M., Ravanbakhsh, S., Gidel, G., Bengio, Y., Malkin, N., and Tong, A. (2024). Iterated denoising energy matching for sampling from boltzmann densities. In *International Conference on Machine Learning*.

Albergo, M. S., Kanwar, G., and Shanahan, P. E. (2019). Flow-based Generative Models for Markov chain Monte Carlo in Lattice Field Theory. *Physical Review D*, 100(3):034515.

Anderson, B. D. O. (1982). Reverse-time Diffusion Equation Models. *Stochastic Processes and their Applications*, 12:313–326.

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. I. (2003). An Introduction to MCMC for Machine Learning. *Machine Learning*, 50(1-2):5–43.

Arbel, M., Matthews, A. G. D. G., and Doucet, A. (2021). Annealed flow transport monte carlo. In *International Conference on Machine Learning*.

Bengio, E., Jain, M., Korablyov, M., Precup, D., and Bengio, Y. (2021). Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation. In *Advances in Neural Information Processing Systems*.

Bengio, Y., Lahlou, S., Deleu, T., Hu, E. J., Tiwari, M., and Bengio, E. (2023). GFlowNet Foundations. *Journal of Machine Learning Research*, 24(210):1–55.

Berner, J., Richter, L., Sendera, M., Rector-Brooks, J., and Malkin, N. (2025). From Discrete-Time Policies to Continuous-Time Diffusion Samplers: Asymptotic Equivalences and Faster Training. *arXiv preprint arXiv:2501.06148*.

Berner, J., Richter, L., and Ullrich, K. (2024). An Optimal Control Perspective on Diffusion-based Generative Modeling. *Transactions on Machine Learning Research*.

Bernton, E., Heng, J., Doucet, A., and Jacob, P. E. (2019). Schrödinger bridge samplers. *arXiv preprint arXiv:1912.13170*.

Blessing, D., Berner, J., Richter, L., and Neumann, G. (2025). Underdamped Diffusion Bridges with Applications to Sampling. In *International Conference on Learning Representations*.

Blessing, D., Jia, X., Esslinger, J., Vargas, F., and Neumann, G. (2024). Beyond ELBOs: A Large-Scale Evaluation of Variational Methods for Sampling. In *International Conference on Machine Learning*.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs. `http://github.com/jax-ml/jax`. GitHub repository.

Brooks, S. P., Gelman, A., Jones, G. L., and Meng, X.-L. (2012). Handbook of Markov Chain Monte Carlo: Hardcover. *CHANCE*, 25:53–55.

Chen, J., Richter, L., Berner, J., Blessing, D., Neumann, G., and Anandkumar, A. (2025). Sequential Controlled Langevin Diffusions. In *International Conference on Learning Representations*.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2018a). Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems*.

Chen, T., Fox, E., and Guestrin, C. (2014). Stochastic Gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning*.

Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2018b). Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300.

De Bortoli, V., Thornton, J., Heng, J., and Doucet, A. (2021). Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling. In *Advances in Neural Information Processing Systems*.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68.

Doucet, A., Grathwohl, W., Matthews, A. G., and Strathmann, H. (2022). Score-based Diffusion Meets Annealed Importance Sampling. In *Advances in Neural Information Processing Systems*.

Duane, S., Kennedy, A., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222.

Föllmer, H. (1986). Time Reversal on Wiener Space. In *Stochastic Processes—Mathematics and Physics*, pages 119–129. Springer.

Frans, K., Hafner, D., Levine, S., and Abbeel, P. (2025). One Step Diffusion via Shortcut Models. In *International Conference on Learning Representations*. ICLR.

Frenkel, D. and Smit, B. (2002). *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, Amsterdam, The Netherlands.

Geffner, T. and Domke, J. (2023). Langevin Diffusion Variational Inference. In *International Conference on Artificial Intelligence and Statistics*.

Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., and Duvenaud, D. (2019). FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models. In *International Conference on Learning Representations*.

Hastings, W. K. (1970). Monte Carlo Sampling Methods using Markov Chains and their Applications. *Biometrika*, 57(1):97–109.

Hernández-Lobato, J. M. and Adams, R. (2015). Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. In *International Conference on Machine Learning*.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*.

Holdijk, L., Du, Y., Hooft, F., Jaini, P., Ensing, B., and Welling, M. (2024). Stochastic Optimal Control for Collective Variable Free Sampling of Molecular Transition Paths. *Advances in Neural Information Processing Systems*.

Hollingsworth, S. A. and Dror, R. O. (2018). Molecular Dynamics Simulation for All. *Neuron*, 99(6):1129–1143.

Jolicoeur-Martineau, A., Li, K., Piché-Taillefer, R., Kachman, T., and Mitliagkas, I. (2021). Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*.

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

Lahlou, S., Deleu, T., Lemos, P., Zhang, D., Volokhova, A., Hernández-García, A., Néhale Ezzine, L., Bengio, Y., and Malkin, N. (2023). A Theory of Continuous Generative Flow Networks. In *International Conference on Machine Learning*.

Liu, X., Gong, C., and Liu, Q. (2023). Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lu, C. and Song, Y. (2025). Simplifying, Stabilizing and Scaling Continuous-time Consistency Models. In *International Conference on Learning Representations*.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. (2022). Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*.

MacKay, D. J. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.

Matthews, A. G. d. G., Arbel, M., Rezende, D. J., and Doucet, A. (2022). Continual Repeated Annealed Flow Transport Monte Carlo. In *International Conference on Machine Learning*.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Neal, R. M. (1995). Bayesian Learning for Neural Networks.

Neal, R. M. (2001). Annealed Importance Sampling. *Statistics and Computing*, 11:125–139.

Nelson, E. (1967). *Dynamical Theories of Brownian Motion*. Princeton University Press, Princeton, NJ.

Noé, F., Köhler, J., and Wu, H. (2019). Boltzmann Generators – Sampling Equilibrium States of Many-Body Systems with Deep Learning. *Science*, 365.

Nüsken, N. and Richter, L. (2021). Solving high-dimensional Hamilton–Jacobi–Bellman PDEs using neural networks: perspectives from the theory of controlled diffusions and measures on path space. *Partial Differential Equations and Applications*, 2(4):48.

Øksendal, B. (2003). *Stochastic Differential Equations: An Introduction with Applications*. Universitext. Springer, 6 edition.

Parisi, G. (1981). Correlation Functions and Computer Simulations. *Nuclear Physics B*, 180(3):378–384.

Phillips, A., Dau, H.-D., Hutchinson, M. J., Bortoli, V. D., Deligiannidis, G., and Doucet, A. (2024). Particle Denoising Diffusion Sampler. In *International Conference on Machine Learning*.

Richter, L. (2021). *Solving high-dimensional PDEs, approximation of path space measures and importance sampling of diffusions*. Doctoral dissertation, BTU Cottbus-Senftenberg.

Richter, L. and Berner, J. (2024). Improved Sampling via Learned Diffusions. In *International Conference on Learning Representations*.

Robert, C. P. (1995). Convergence Control Methods for Markov Chain Monte Carlo Algorithms. *Statistical Science*, 10(3):231–253.

Rossky, P. J., Doll, J. D., and Friedman, H. L. (1978). Brownian Dynamics as Smart Monte Carlo Simulation. *The Journal of Chemical Physics*, 69(10):4628–4633.

Salimans, T. and Ho, J. (2022). Progressive Distillation for Fast Sampling of Diffusion Models. In *International Conference on Learning Representations*.

Song, Y. and Dhariwal, P. (2024). Improved Techniques for Training Consistency Models. In *International Conference on Learning Representations*.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. (2023). Consistency Models. In *International Conference on Machine Learning*.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.

Sun, J., Berner, J., Richter, L., Zeinhofer, M., Müller, J., Azizzadenesheli, K., and Anandkumar, A. (2024). Dynamical Measure Transport and Neural PDE Solvers for Sampling. *arXiv preprint arXiv:2407.07873*.

Vargas, F., Grathwohl, W. S., and Doucet, A. (2023). Denoising Diffusion Samplers. In *International Conference on Learning Representations*.

Vargas, F., Padhy, S., Blessing, D., and Nüsken, N. (2024). Transport Meets Variational Inference: Controlled Monte Carlo Diffusions. In *International Conference on Learning Representations*.

Welling, M. and Teh, Y. W. (2011). Bayesian Learning via Stochastic Gradient Langevin Dynamics. In *International Conference on Machine Learning*.

Wu, D., Wang, L., and Zhang, P. (2019). Solving Statistical Mechanics using Variational Autoregressive Networks. *Physical Review Letters*, 122(8):080602.

Wu, H., Köhler, J., and Noé, F. (2020). Stochastic normalizing flows. *Advances in Neural Information Processing Systems*, 33:5933–5944.

Zhang, D., Chen, R. T. Q., Liu, C.-H., Courville, A., and Bengio, Y. (2024). Diffusion Generative Flow Samplers: Improving Learning Signals through Partial Trajectory Optimization. In *International Conference on Learning Representations*.

Zhang, Q. and Chen, Y. (2022). Path Integral Sampler: A Stochastic Control Approach For Sampling. In *International Conference on Learning Representations*.

Zhang, R., Cooper, A. F., and De Sa, C. (2020a). AMAGOLD: Amortized Metropolis Adjustment for Efficient Stochastic Gradient MCMC. In *International Conference on Artificial Intelligence and Statistics*.

Zhang, R., Li, C., Zhang, J., Chen, C., and Wilson, A. G. (2020b). Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning. In *International Conference on Learning Representations*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes.** Mathematical setting, and explicit assumptions in section A, mathematical modeling and preliminaries in section 3 (SDEs, FB–RND).
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Yes.** Complexity and cost tradeoffs analyzed via NFEs tables 1 and 2 and break-even formula in section D.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Yes.** Anonymized code URL in section 5.1.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Yes.** Assumptions (Lipschitz/regularity, existence/uniqueness, bounded schedules) stated in section A and inside propositions if applicable.
   (b) Complete proofs of all theoretical results. **Yes.** Proofs for the key results propositions 1 to 3 are available in sections B.1, B.2 and C respectively.
   (c) Clear explanations of any assumptions. **Yes.** Assumptions reiterated and explained alongside each proposition and in the Notation/Assumptions section section A.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Yes.** Code and configuration files are in the github repor link in section 5.1; tables/figures reference the exact setups and metrics; repo contains run scripts.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes.** Training details (architecture, optimizer, EMA, clipping, hyperparameters) in section E.1; SDE/schedule in section E.3; metrics in section 5.1.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes.** Reported as mean $\pm$ std over 20 runs (section 5.1); metric definitions in section 5.1.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **Yes.** We use NVIDIA A6000 GPUs section E.1.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. **Yes.** Benchmarks are cited (section 5.1, Blessing et al. (2024)); baseline methods are cited throughout Related Work and tables. Any re-used code is also cited inside the repo.
   (b) The license information of the assets, if applicable. **Not Applicable.**
   (c) New assets either in the supplemental material or as a URL, if applicable. **Not Applicable.** No new datasets/models released beyond the anonymized code.
   (d) Information about consent from data providers/curators. **Not Applicable.**
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not Applicable.**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. **Not Applicable.**
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not Applicable.**
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not Applicable.**

Supplementary Materials

A Notation and Assumptions

Throughout we fix a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in[0,T]}, \mathbb{P})$ supporting a d-dimensional standard Wiener process $\mathbf{w} = (\mathbf{w}_t)_{t\in[0,T]}$. All stochastic integrals are in the Itô sense unless stated otherwise.

We denote by $\mathbf{x} = (\mathbf{x}_t)_{t\in[0,T]} \in C([0,T], \mathbb{R}^D)$ the forward diffusion process governed by the SDE equation (2). The law of the entire path $\mathbf{x}_{0:T}$ is written $\mathbb{P}^{\pi,u}$ when the initial density is π and the control (drift correction) is u. Analogously, $\overleftarrow{\mathbb{P}}^{\tau,v}$ denotes the law of the reverse-time SDE equation (3) with terminal density τ and control v.

The unnormalized target density is $\rho : \mathbb{R}^D \to (0,\infty)$ with unknown partition function $Z = \int \rho(x)\mathrm{d}x$. The normalized target is $p_{\text{target}} = \rho/Z$. The prior (reference) density is p_{prior}. Unless otherwise noted, expectations $\mathbb{E}[\cdot]$ are taken with respect to the forward path measure $\mathbb{P}^{\pi,u}$.

$\mathcal{U} \subset C(\mathbb{R}^D \times [0,T], \mathbb{R}^D)$ is the set of admissible controls. A subscript θ (e.g. u_θ) indicates a neural-network parameter vector $\theta \in \mathbb{R}^p$.

For numerical schemes we use a mesh $0 = t_0 < t_1 < \cdots < t_N = T$ and write $\Delta t_n = t_{n+1} - t_n$. The shorthand $\mathbf{x}_{0:N} := (\mathbf{x}_{t_0}, \ldots, \mathbf{x}_{t_N})$ denotes the sampled discrete path; $p_{n+1|n}$ refer to the corresponding Markov transition densities.

We assume that $x \mapsto f(x,t)$ and $x \mapsto u(x,t)$ are globally Lipschitz with at most linear growth, uniformly in $t \in [0,T]$. We assume $\sigma(t)$ is continuous and bounded away from 0 and ∞ on $[0,T]$. These standard conditions guarantee strong existence and uniqueness for equations (2) and (3) (Øksendal, 2003, Theorem 5.2.1), absolute continuity of the induced path measures, and finiteness of the Radon-Nikodym derivatives we use.

We use the dot notation $\dot{\mathbf{x}}_t := \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}_t$ for time derivatives. Given a (learned) vector field $b_\theta : \mathbb{R}^D \times [0,T] \to \mathbb{R}^D$, the probability flow ODE (PF ODE) associated with the generative SDE is

$$\dot{\mathbf{x}}_t = b_\theta(\mathbf{x}_t, t), \qquad \mathbf{x}_0 = x_0.$$

Its flow map $\phi_t : \mathbb{R}^D \to \mathbb{R}^D$ satisfies $\phi_0(x) = x$ and $\frac{\mathrm{d}}{\mathrm{d}t}\phi_t(x) = b_\theta(\phi_t(x), t)$. For a transport horizon $d > 0$ we write $T := \phi_d$ for the single-step deterministic map. We use $\nabla T(x)$ for the Jacobian, $J_T(x) := \nabla T(x)$, and $\log\det$ for the log-determinant.

Along a PF ODE trajectory we define the scalar accumulator ℓ_t by the auxiliary ODE

$$\dot{\ell}_t = \nabla \cdot b_\theta(\mathbf{x}_t, t), \qquad \ell_0 = 0,$$

so that the log-Jacobian of $T = \phi_d$ along the trajectory starting at x_0 is $\log|\det \nabla T(x_0)| = \ell_d$ (Chen et al., 2018a).

B Radon-Nikodym Derivative

B.1 Forward-backward Radon-Nikodym derivative

Consider forward and reverse-time stochastic processes on $\mathbb{R}^D$ over a time interval $[0,T]$, each described by the following SDEs:

$$\mathrm{d}\mathbf{x}_t = (f + \sigma u)(\mathbf{x}_t, t)\mathrm{d}t + \sigma(t)\mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_0 \sim \pi, \tag{23}$$

$$\mathrm{d}\mathbf{x}_t = (-\overleftarrow{f} + \overleftarrow{\sigma}\overleftarrow{v})(\mathbf{x}_t, t)\mathrm{d}t + \overleftarrow{\sigma}(t)\mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_0 \sim \tau. \tag{24}$$

Proposition 1 (Forward-backward Radon-Nikodym derivative), restated. For $u, v \in \mathcal{U} \subset C(\mathbb{R}^D \times [0,T], \mathbb{R}^D)$, $\mu \in C(\mathbb{R}^D \times [0,T], \mathbb{R}^D)$ and $\sigma \in C([0,T], \mathbb{R})$. The likelihood ratio between the path measures $\mathbb{P}^{\pi,u}$ and $\overleftarrow{\mathbb{P}}^{\tau,v}$ of the processes equation (23) and equation (24), respectively, is given by

$$\begin{aligned}\log \frac{\mathrm{d}\mathbb{P}^{\pi,u}}{\mathrm{d}\overleftarrow{\mathbb{P}}^{\tau,v}}(\mathbf{x}) = &\frac{1}{2}\int_0^T \left(\|\sigma f\|^2 - \|\sigma f + u\|^2\right)(\mathbf{x}_t, t)\mathrm{d}t + \int_0^T \left(\frac{u}{\sigma}\right)(\mathbf{x}_t, t)\cdot \mathrm{d}\mathbf{x}_t \\ &+ \frac{1}{2}\int_0^T \left(\|\sigma f + v\|^2 - \|\sigma f\|^2\right)(\mathbf{x}_t, t)\mathrm{d}t - \int_0^T \left(\frac{v}{\sigma}\right)(\mathbf{x}_t, t)\cdot \overleftarrow{\mathrm{d}}\mathbf{x}_t \\ &+ \log \frac{\pi(\mathbf{x}_0)}{\tau(\mathbf{x}_T)}.\end{aligned} \tag{25}$$

Proof. The result follows the structure of (Vargas et al., 2024, proof of Proposition 2.2). We introduce a reference process with control w and define the corresponding path measures $\mathbb{P}^{\pi,w}$, $\overleftarrow{\mathbb{P}}^{\tau,w}$. The Radon-Nikodym derivative can be decomposed as the forward and backward Radon-Nikodym derivatives:

$$\log \frac{\mathrm{d}\mathbb{P}^{\pi,u}}{\mathrm{d}\overleftarrow{\mathbb{P}}^{\tau,v}}(\mathbf{x}) = \log \frac{\mathrm{d}\mathbb{P}^{\pi,u}}{\mathrm{d}\mathbb{P}^{\pi,w}}(\mathbf{x}) + \log \frac{\mathrm{d}\overleftarrow{\mathbb{P}}^{\tau,w}}{\mathrm{d}\overleftarrow{\mathbb{P}}^{\tau,v}}(\mathbf{x}) + \log \frac{\pi(\mathbf{x}_0)}{\tau(\mathbf{x}_T)}. \tag{26}$$

The result then follows by applying Girsanov theorem (Nüsken and Richter, 2021, Section A.1) forward and backward in time. The forward term can be expressed as

$$\log \frac{\mathrm{d}\mathbb{P}^{\pi,u}}{\mathrm{d}\mathbb{P}^{\pi,w}}(\mathbf{x}) = \frac{1}{2}\int_0^T \left(\|\sigma f + w\|^2 - \|\sigma f + u\|^2\right)(\mathbf{x}_t, t)\mathrm{d}t + \int_0^T \left((u-w)\sigma^{-1}\right)(\mathbf{x}_t, t)\cdot \mathrm{d}\mathbf{x}_t$$

Similarly, the backward term can be expressed as:

$$\begin{aligned}\log \frac{\mathrm{d}\overleftarrow{\mathbb{P}}^{\tau,w}}{\mathrm{d}\overleftarrow{\mathbb{P}}^{\tau,v}}(\mathbf{x}) &= \log \frac{\mathrm{d}\mathbb{P}^{\tau,w}}{\mathrm{d}\mathbb{P}^{\tau,v}}(\mathbf{x}) \\ &\quad - \int_0^T \left((w-v)\sigma^{-1}\right)(\mathbf{x}_t, t)\cdot \mathrm{d}\mathbf{x}_t + \int_0^T \left((w-v)\sigma^{-1}\right)(\mathbf{x}_t, t)\cdot \overleftarrow{\mathrm{d}}\mathbf{x}_t \\ &= \frac{1}{2}\int_0^T \left(\|\sigma f + v\|^2 - \|\sigma f + w\|^2\right)(\mathbf{x}_t, t)\mathrm{d}t + \int_0^T \left((w-v)\sigma^{-1}\right)(\mathbf{x}_t, t)\cdot \overleftarrow{\mathrm{d}}\mathbf{x}_t\end{aligned}$$

Setting $w := 0$ and plugging into equation (26) yields the Radon-Nikodym derivative equation (5). □

B.2 Discrete forward-backward likelihood ratio and a lower bound on $\log Z$

We discretize the time interval $[0,T]$ by $0 = t_0 < t_1 < \cdots < t_N = T$.

Proposition 2 (Finite-step forward-backward bound), restated. Let $\mathbb{P}^{\pi,u}$ be the path measure of the forward SDE equation (23) driven by a (possibly approximate) control u, and let $\overleftarrow{\mathbb{P}}^{\tau,v}$ be the reverse-time SDE equation (24) with terminal density $\tau = \rho/Z$. Denote by

$$p_{n+1|n}(\mathbf{x}_{n+1} \mid \mathbf{x}_n), \qquad p_{n|n+1}(\mathbf{x}_n \mid \mathbf{x}_{n+1})$$

the one-step transition densities of any consistent discretization scheme applied to the two SDEs. Then for every discrete trajectory $\mathbf{x}_{0:N} := (\mathbf{x}_0, \ldots, \mathbf{x}_N)$ generated forward in time under $\mathbb{P}^{\pi,u}$,

$$\log \frac{\mathrm{d}\mathbb{P}^{\pi,u}}{\mathrm{d}\overleftarrow{\mathbb{P}}^{\tau,v}}(\mathbf{x}_{0:N}) = \log \frac{\pi(\mathbf{x}_0)}{\tau(\mathbf{x}_N)} + \sum_{n=0}^{N-1} \log \frac{p_{n+1|n}(\mathbf{x}_{n+1} \mid \mathbf{x}_n)}{p_{n|n+1}(\mathbf{x}_n \mid \mathbf{x}_{n+1})}. \tag{27}$$

Consequently

$$\mathbb{E}_{\mathbb{P}^{\pi,u}}\Big[\sum_{n=0}^{N-1} \log \frac{\rho(\mathbf{x}_N) p_{n+1|n}(\mathbf{x}_{n+1} \mid \mathbf{x}_n)}{\pi(\mathbf{x}_0) p_{n|n+1}(\mathbf{x}_n \mid \mathbf{x}_{n+1})}\Big] \le \log Z \tag{28}$$

with equality when the exact optimal drifts are used and the discretization is exact. The inequality holds for any step-size sequence and for any approximate control u.

Proof. Factorizing the discrete path densities, $\mathbb{P}^{\pi,u}(\mathbf{x}_{0:N}) = \pi(\mathbf{x}_0)\prod_{n=0}^{N-1} p_{n+1|n}(\mathbf{x}_{n+1} \mid \mathbf{x}_n)$ and $\overleftarrow{\mathbb{P}}^{\tau,v}(\mathbf{x}_{0:N}) = \tau(\mathbf{x}_N)\prod_{n=0}^{N-1} p_{n|n+1}(\mathbf{x}_n \mid \mathbf{x}_{n+1})$, gives equation (27) immediately. Taking the expectation of both sides with respect to $\mathbb{P}^{\pi,u}$ and noting that $\mathbb{E}_{\mathbb{P}^{\pi,u}}\big[\log \frac{\mathrm{d}\mathbb{P}^{\pi,u}}{\mathrm{d}\overleftarrow{\mathbb{P}}^{\tau,v}}\big] = \mathrm{KL}(\mathbb{P}^{\pi,u}\|\overleftarrow{\mathbb{P}}^{\tau,v}) \geq 0$ establishes equation (28). Because the Kullback–Leibler divergence is non-negative, the bound is always valid, even if u (or v) is sub-optimal or the numerical scheme is coarse. □

In discrete time, the Radon-Nikodym derivative between the path measures of a forward simulator and its reverse factorizes into a boundary term and a forward-over-backward product of one-step Markov kernels (equation (27)). Here $p_{n+1|n}$ is the forward one-step kernel of the chosen integrator for the generative SDE, while $p_{n|n+1}$ is the corresponding backward kernel (the conditional of the same discrete chain run backward, i.e., conditioned on the right endpoint). Crucially, $p_{n|n+1}$ is not in general equal to "the EM step of the reverse-time SDE". See (Blessing et al., 2025, Remark 3.2) for the same discrete RND viewpoint in the underdamped setting.

Practical Monte-Carlo estimator. Drawing M independent paths under the forward discretization and substituting them into equation (28) yields the unbiased estimator

$$\widehat{\log Z} := \frac{1}{M}\sum_{m=1}^{M}\sum_{n=0}^{N-1} \log \frac{\rho(\mathbf{x}_N^{(m)})p_{n+1|n}(\mathbf{x}_{n+1}^{(m)} \mid \mathbf{x}_n^{(m)})}{\pi(\mathbf{x}_0^{(m)})p_{n|n+1}(\mathbf{x}_n^{(m)} \mid \mathbf{x}_{n+1}^{(m)})}, \tag{29}$$

Drawing M paths yields an unbiased estimator of the ELBO (the LHS of equation 28). Since the ELBO lower-bounds $\log Z$, this provides a lower bound in expectation.

Because the proof only uses (i) absolute continuity of the discrete path measures and (ii) positivity of KL, it remains valid when the continuous SDE is replaced by any approximate simulator. Using a more accurate scheme tightens the bound but can never violate equation (28).

Setting $N = 1$ in Prop. 2 gives the weight

$$\log \frac{\mathrm{d}\mathbb{P}^{\pi,u}}{\mathrm{d}\overleftarrow{\mathbb{P}}^{\tau,v}}((\mathbf{x}_0, \mathbf{x}_N)) = \log \frac{\pi(\mathbf{x}_0)}{\tau(\mathbf{x}_N)} + \log \frac{p_{N|0}(\mathbf{x}_N \mid \mathbf{x}_0)}{p_{0|N}(\mathbf{x}_0 \mid \mathbf{x}_N)}, \tag{30}$$

i.e. the log-ratio of two Gaussian kernels plus the prior/target term. Because it is still a likelihood ratio, the expectation remains an ELBO even when the single step is large.

Line–integral discretization. Many diffusion papers compute the path log-weight directly from the Girsanov formula (Berner et al., 2024; Richter and Berner, 2024),

$$\log \frac{\mathrm{d}\mathbb{P}^{\pi,u}}{\mathrm{d}\overleftarrow{\mathbb{P}}^{\tau,v}} = \int_0^T (u+v)\cdot\Big(u + \frac{v-u}{2} + \nabla\cdot(\sigma v - \mu)\Big)(\mathbf{x}_t, t)\mathrm{d}t + \int_0^T \big(u+v\big)(\mathbf{x}_t, t)\cdot \mathrm{d}\mathbf{w}_t + \log\frac{\pi(\mathbf{x}_0)}{\tau(\mathbf{x}_T)}. \tag{31}$$

Discretized as an approximate Riemann sum, this divergence-based Radon-Nikodym derivative does not guarantee a lower-bound for the log-normalization constant (Blessing et al., 2025). When sampling using $N = 1$ steps, the inner running cost of the line-integral discretization vanishes and the estimation collapses to the boundary ratios.

B.3 Why EM breaks at one step

EM is popular, but fails at coarse discretization. With many small steps, Euler-Maruyama (EM) discretization produces nearly time-symmetric pairs, so each log-ratio term is close to zero and the importance weights are well-behaved.

Write $\Delta t_n = t_{n+1} - t_n$. For the forward SDE equation (2), EM gives the forward kernel

$$p_{n+1|n}(x'|x) = \mathcal{N}\Big(x'; x + \big(f(t_n)x + \sigma(t_n)u(x,t_n)\big)\Delta t_n, \sigma(t_n)^2\Delta t_n I\Big). \tag{32}$$

Applied to the reverse-time SDE equation (3), the right-point EM update gives

$$\widetilde{p}_{n|n+1}(x|x') = \mathcal{N}\Big(x; x' - \overleftarrow{f}(t_{n+1})x'\Delta t_n, \overleftarrow{\sigma}(t_{n+1})^2\Delta t_n I\Big). \tag{33}$$

Both kernels are consistent as $\max_n \Delta t_n \to 0$. At coarse resolution, however, they are highly time-asymmetric, since equation (32) is calibrated at (x, t_n) and equation (33) at (x', t_{n+1}).

When Δt_n is large, $\widetilde{p}_{n|n+1}$ in equation (33) is not the time-adjoint of equation (32).

Definition 1 (Time-adjoint.)**.** Given a forward kernel $p_{n+1|n}$ and a marginal π_n, the time-reversed kernel is

$$p^\star_{n|n+1}(x|x') = \frac{\pi_n(x) p_{n+1|n}(x'|x)}{\int \pi_n(z) p_{n+1|n}(x'|z) \mathrm{d}z}.$$

We call $p_{n+1|n}$ and $p_{n|n+1}$ *time-adjoint* if $p_{n|n+1} = p^\star_{n|n+1}$ for the π_n transported by $p_{n+1|n}$.

If $p_{n|n+1} = p^\star_{n|n+1}$, then the discrete forward–backward (FB) Radon-Nikodym ratio equation (6) is an exact likelihood ratio for the discrete path measures and its expectation gives a valid ELBO.

Replacing $p^\star_{n|n+1}$ by $\widetilde{p}_{n|n+1}$ introduces a cross-entropy gap at each step:

$$\mathbb{E}\big[\log p_{n+1|n}(\mathbf{x}_{n+1}|\mathbf{x}_n) - \log \widetilde{p}_{n|n+1}(\mathbf{x}_n|\mathbf{x}_{n+1})\big] = -\mathbb{E}_{\mathbf{x}_{n+1}}\Big[\mathrm{KL}\big(p^\star_{n|n+1}(\cdot|\mathbf{x}_{n+1}) \| \widetilde{p}_{n|n+1}(\cdot|\mathbf{x}_{n+1})\big)\Big] \le 0.$$

At $N = 1$ this single KL term can be very large because the forward EM mean and variance scale with $\beta(t_1)$ at the right endpoint and with the control u, whereas a surrogate backward that ignores these choices concentrates elsewhere. The ELBO and estimation of $\log Z$ become overly conservative, even when the samples remain visually plausible.

A 1D example. Consider the VP SDE family with constant $\beta > 0$ and $\sigma(t) = \sigma_0\sqrt{\beta}$, and take $u_\theta \equiv 0$ for clarity. We discretize $[0, 1]$ with a single step $\Delta t = 1$.

The forward SDE uses drift $+\frac{1}{2}\beta x$, so one EM step yields the linear-Gaussian update

$$x_1 = Ax_0 + \eta, \qquad A := 1 + \tfrac{1}{2}\beta, \quad \eta \sim \mathcal{N}(0, Q), \quad Q := \beta\sigma_0^2.$$

Suppose $x_0 \sim \mathcal{N}(0, \Sigma_0)$ with $\Sigma_0 = \sigma_0^2$ (the usual VP prior).

Then the true backward Markov kernel of this discrete forward chain is the Kalman posterior

$$p_{\text{true}}(x_0|x_1) = \mathcal{N}\big(Kx_1, \Sigma_{\text{post}}\big), \qquad K := \frac{\Sigma_0 A}{A^2\Sigma_0 + Q} = \frac{1 + \frac{1}{2}\beta}{(1 + \frac{1}{2}\beta)^2 + \beta},$$

$$\Sigma_{\text{post}} = \Sigma_0 - \frac{\Sigma_0^2 A^2}{A^2\Sigma_0 + Q} = \sigma_0^2 \cdot \frac{\beta}{(1 + \frac{1}{2}\beta)^2 + \beta}.$$

By contrast, the "EM backward" kernel one obtains by applying a right-endpoint EM step to the reverse direction is

$$p_{\text{em-bwd}}(x_0|x_1) = \mathcal{N}\big((1 - \tfrac{1}{2}\beta)\, x_1,\ Q\big).$$

For a typical value $\beta = 10$ and $\sigma_0^2 = 1$ we get

$$A = 6, \quad Q = 10, \quad K = \frac{6}{36 + 10} \approx 0.1304, \quad \Sigma_{\text{post}} = \frac{10}{46} \approx 0.217.$$

The two conditionals differ by more than an order of magnitude in both mean and variance:

$$\text{mean: } Kx_1 \approx 0.13x_1 \quad \text{vs} \quad (1 - \tfrac{1}{2}\beta)x_1 = -4x_1,$$

$$\text{variance: } \Sigma_{\text{post}} \approx 0.217 \quad \text{vs} \quad Q = 10.$$

Hence for fixed x_1, $p_{\text{true}}(x_0|x_1)$ is a tight Gaussian near $0.13x_1$, while $p_{\text{em-bwd}}(x_0|x_1)$ is an extremely broad Gaussian centered at $-4\,x_1$.

The per-step mismatch can be quantified exactly:

$$\mathbb{E}_{x_0|x_1}[\log p_{\text{true}}(x_0 \mid x_1) - \log p_{\text{EM-bwd}}(x_0 \mid x_1)] = \mathrm{KL}(p_{\text{true}} \| \, p_{\text{EM-bwd}}) \ \geq 0$$

The conditional KL: $\mathbb{E}\big[\mathrm{KL}(p^\star(\cdot \mid x_1) \| p_{\text{EM-bwd}}(\cdot \mid x_1))\big] \approx 40$ nats. By equation 11, this subtracts roughly 40 nats from the per-step FB-RND contribution when we use $p_{\text{EM-bwd}}$ instead of the time-adjoint $p^\star$.

Therefore, replacing the true discrete backward kernel by "EM backward" breaks the forward-backward identity at one step and catastrophically degrades the ELBO and $\log Z$ estimator.

In discrete time, the FB Radon-Nikodym ratio requires the backward Markov kernel of the same discrete chain used forward. At $N = 1$, EM's right-point forward step and its right-point "backward" step are not a time-adjoint pair, so the per-step log-ratio is dominated by a large cross-entropy term between the true and surrogate conditionals. This is negligible at small Δt but becomes dominant at one or a few steps.

C Deterministic-flow importance weights for single-step estimation of Z

C.1 Deterministic $\log Z$ estimate

We show how to compute an importance weight from a deterministic transport induced by the probability-flow (PF) ODE. This enables a single-step estimator of the partition function $Z = \int_{\mathbb{R}^D} \rho(y)\mathrm{d}y$ using samples from the prior p_{prior}.

Proposition 3 (Deterministic-flow IS weight), restated Let $q := T_{\#} p_{\text{prior}}$ be the push-forward of the prior by the flow map $T = \phi_d$. Then for any $x_0 \in \mathbb{R}^D$,

$$w(x_0) = \frac{\rho\big(T(x_0)\big)}{q\big(T(x_0)\big)} = \rho\big(\phi_d(x_0)\big)\frac{\big|\det \nabla T(x_0)\big|}{p_{\text{prior}}(x_0)}. \tag{34}$$

Moreover,

$$Z = \mathbb{E}_{x_0 \sim p_{\text{prior}}}\big[w(x_0)\big], \qquad \widehat{Z} = \frac{1}{M}\sum_{i=1}^{M} w\big(x_0^{(i)}\big) \ \text{ is an unbiased estimator of } Z. \tag{35}$$

Proof. Because T is a diffeomorphism, the change of variables formula gives

$$q(y) = p_{\text{prior}}\big(T^{-1}(y)\big)|\det \nabla T(T^{-1}(y))|^{-1}.$$

Substituting $y = T(x_0)$ yields

$$q(T(x_0)) = \frac{p_{\text{prior}}(x_0)}{|\det \nabla T(x_0)|},$$

which implies equation (34). Then

$$Z = \int \rho(y)\mathrm{d}y = \int \rho(T(x))|\det \nabla T(x)|\mathrm{d}x = \mathbb{E}_{x_0 \sim p_{\text{prior}}}\left[\frac{\rho(T(x_0))|\det \nabla T(x_0)|}{p_{\text{prior}}(x_0)}\right],$$

which is equation (35). Unbiasedness of $\widehat{Z}$ follows by linearity of expectation. □

C.2 Hutchinson trace estimator

Evaluating $\nabla \cdot b_\theta(\mathbf{x}_t, t) = \mathrm{tr}\big(\nabla_x b_\theta(\mathbf{x}_t, t)\big)$ naïvely is $O(d^2)$ and requires forming Jacobians explicitly. Following Grathwohl et al. (2019), we estimate the trace in linear time using the Hutchinson identity:

$$\mathrm{tr}(A) = \mathbb{E}_{\boldsymbol{\varepsilon}}\big[\boldsymbol{\varepsilon}^\top A \boldsymbol{\varepsilon}\big],$$

for any square matrix A and zero-mean probe $\boldsymbol{\varepsilon}$ with $\mathrm{Cov}(\boldsymbol{\varepsilon}) = I$ (e.g. Rademacher or standard Gaussian). Applied to $A = \nabla_x b_\theta(\mathbf{x}_t, t)$, this gives the unbiased estimator

$$\nabla \cdot b_\theta(\mathbf{x}_t, t) \approx \boldsymbol{\varepsilon}^\top \big(\nabla_x b_\theta(\mathbf{x}_t, t)\big)\boldsymbol{\varepsilon},$$

which can be computed with a pair of vector-Jacobian products (VJPs) at roughly the cost of one forward/backward evaluation.

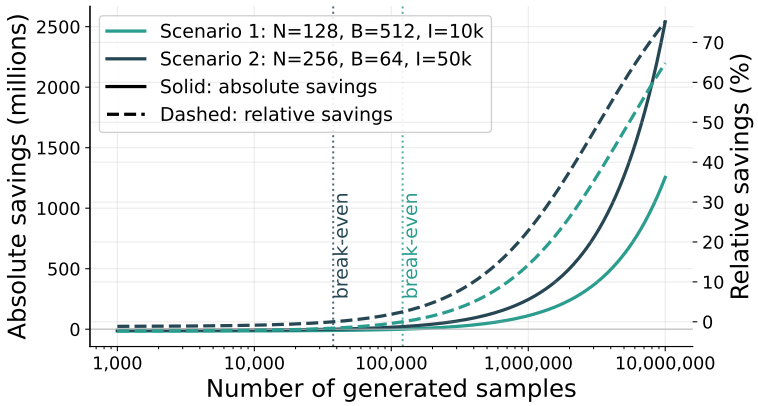

Figure 3: Absolute and relative NFE savings of OSDS over the baseline diffusion samplers PIS, DDS as a function of the number of generated samples (log scale).

D Cost-benefit analysis of one-step inference

A practical question is when the additional NFEs spent on the state- and volume-consistency branches pay off at inference time. Let N denote the number of Euler-Maruyama steps used by a baseline diffusion sampler, B the training batch size and I the number of training iterations. A baseline method uses $\mathrm{NFE}_{\mathrm{base}}(S) = IBN + SN$ network evaluations to train and then generate S samples. OSDS reuses the same N-step discretization for the FB-RND base loss but adds three extra evaluations per iteration for the teacher-student branches, and it applies a single PF-ODE step per test-time sample. Its total cost is $\mathrm{NFE}_{\mathrm{OSDS}}(S) = IB(N{+}3) + S$. The total NFE savings after drawing S samples are therefore $\Delta_{\mathrm{NFE}}(S) = \mathrm{NFE}_{\mathrm{base}}(S) - \mathrm{NFE}_{\mathrm{OSDS}}(S) = (N-1)S - 3IB$, so OSDS becomes strictly cheaper once $S > S_{\mathrm{break}} := 3IB/(N-1)$. For $S \gg S_{\mathrm{break}}$ the training term becomes negligible as the relative savings approach $1 - 1/N$.

Figure 3 illustrates $\Delta_{\mathrm{NFE}}(S)$ and the corresponding relative savings for two representative configurations. Scenario 1 uses N=128, B=512, and I=10k iterations, corresponding to a moderately sized run. Scenario 2 uses a more expensive baseline with N=256, B=64, and I=50k. These plots highlight that the distillation overhead is a one-time cost: once the sampler is reused for thousands to millions of draws, OSDS yields substantial end-to-end computational savings.

E Implementation Details

E.1 Model architecture and optimization

All methods use a PISGRAD backbone of identical size (same depth/width and Fourier features for time) (Zhang and Chen, 2022). Our variant (step-conditioned) augments the backbone with a small embedding for the step size d: we compute Fourier features for both t and d, concatenate them, and feed the result into the time coder.

We use AdamW (Loshchilov and Hutter, 2017) with decoupled weight decay 0.1 and global-norm gradient clipping (threshold 1.0 by default). All implementations are in JAX/Flax/Optax (Bradbury et al., 2018). We maintain an EMA of parameters with decay 0.999. We train and evaluate using a NVIDIA A6000 GPU.

The learning rate is tuned for all methods in $\{0.001, 0.0001, 0.00001\}$, following Blessing et al. (2024). We use the model checkpoints with the best moving average ELBO.

Algorithm 3 Discrete forward–backward sampling ELBO/FB-RND (proposition 2)

Require: trained θ; densities ρ, π; steps N; transition densities $p_{n+1|n}$ and time-adjoint $p_{n|n+1}$ defined by the discrete chain.

Ensure: path $x_{0:N}$, path weight w, and ELBO contribution $\log w$.

$x_0 \sim \pi$
$x_n \leftarrow x_0$
$\log w \leftarrow 0$
for $n = 0$ **to** $N-1$ **do**
  $x_{n+1} \sim p_{n+1|n}(x_{n+1} \mid x_n)$
  $\log w \leftarrow \log w + \log p_{n+1|n}(x_{n+1} \mid x_n) - \log p_{n|n+1}(x_n \mid x_{n+1})$
  $x_n \leftarrow x_{n+1}$
end for
$x \leftarrow x_n$
$\log w \leftarrow \log w + \log \rho(x_N) - \log \pi(x_0)$
return $x_{0:N}$, $w = e^{\log w}$, $\log w$

E.1.1 Jacobian regularization

To stabilize the deterministic flow during distillation, we add a small Jacobian-norm penalty on the step-conditioned field used by the student. Let $J_u(x,t,d) = \partial u_\theta(x,t,d)/\partial x \in \mathbb{R}^{d\times d}$ denote the Jacobian w.r.t. the state. We define

$$L_{\text{Jac}}(x_t;t,d) = \mathbb{E}_v\Big[\frac{1}{d}\|J_u(x_t,t,d)v\|_2^2\Big] \approx \frac{1}{d}\|J_u(x_t,t,d)v\|_2^2, \tag{36}$$

where v is a single Rademacher probe ($v_i \sim \text{Unif}\{\pm 1\}$). This is a Hutchinson-style estimator: $\mathbb{E}_v\|Jv\|_2^2 = \text{tr}(J^\top J) = \|J\|_F^2$, so L_{Jac} penalizes the (scaled) Frobenius norm of the Jacobian and hence the local Lipschitz constant of u_θ (we absorb the $1/d$ scaling into the weight). We compute $J_u(x_t,t,d)v$ with a single JVP at the start of the student's step x_t.

The full distillation objective is then

$$L = L_{\text{state}} + \lambda_{\text{trace}} L_{\text{trace}} + \lambda_{\text{Jac}} L_{\text{Jac}},$$

This regularization mirrors the JVP/trace machinery used in continuous/ODE flows: Hutchinson probes are standard for Jacobian traces in, and controlling Lipschitzness improves numerical stability and reduces stiffness (which otherwise increases solver work). See the instantaneous change-of-variables and trace/JVP discussion in CNFs (Chen et al., 2018a) and FFJORD (Grathwohl et al., 2019).

E.2 Loss algorithmic details

This section spells out how we instantiate the three computations used by OSDS: (i) the fine-resolution forward–backward Radon-Nikodym (RND) training loss, (ii) the teacher-student self-distillation losses (state and volume), and (iii) the discrete forward–backward (FB–RND) bound used for evaluation when a reliable backward kernel is available.

Algorithm 3 evaluates the discrete FB likelihood ratio (proposition 2, equations (6) and (7)). Algorithm 4 implements the shortcut self-distillation objective (equations (10), (21) and (22)).

The self-distillation loss adds a minimal computation overhead of three additional NFEs per iteration (out of 128, 256 or 512, for instance).

E.3 VP SDE and time schedule

Unless specified otherwise, we use the variance–preserving (VP) noising SDE (Ho et al., 2020; Song et al., 2021)

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\mathrm{d}t + \sqrt{\beta(t)}\sigma_0\mathrm{d}\mathbf{w}_t, \qquad t \in [0,1], \tag{37}$$

where $\sigma_0 > 0$ is a constant scale and $\beta(t)$ is a nonnegative schedule. In our experiments we use a linear schedule $\beta(t) = \beta_{\min} + t(\beta_{\max} - \beta_{\min})$ with $\beta_{\min} = 0.01$ and $\beta_{\max} = 10$.

Algorithm 4 Self-distillation losses

Require: Anchors $\{(x_t^{(b)}, t)\}_{b=1}^{B}$; per-anchor steps $\{d^{(b)}\}_{b=1}^{B}$; control u_θ; PF ODE drift $b_\theta(x, \tau, d)$; integrator Ψ that can integrate an augmented ODE $(\dot{x}, \dot{\ell})$ and return both final state and log-volume; frozen teacher params $\theta' = \texttt{stopgrad}(\theta)$; shared Hutchinson probe ε.

Ensure: $\mathcal{L}_{\text{state}}$, $\mathcal{L}_{\text{vol}}$.

$\text{sum}_{\text{state}} \leftarrow 0; \quad \text{sum}_{\text{vol}} \leftarrow 0$
for $b = 1$ **to** B **do**
  $x \leftarrow x_t^{(b)}; \quad t_0 \leftarrow t; \quad d \leftarrow d^{(b)}$
  Define teacher augmented drift (half-step):
    $g_{\theta'}^{(d/2)}(x, \ell, \tau) \;=\; \big(b_{\theta'}(x, \tau, \frac{d}{2}), \textsc{Hutchinson}(\nabla_x b_{\theta'}(x, \tau, \frac{d}{2}), \varepsilon)\big)$
  Teacher half-step 1: $(\tilde{x}, \tilde{v}) \leftarrow \Psi\big(g_{\theta'}^{(d/2)}, (x, 0), t_0, \frac{d}{2}\big)$
  Teacher half-step 2: $(x_{t+d}^{\text{teach}}, \bar{v}) \leftarrow \Psi\big(g_{\theta'}^{(d/2)}, (\tilde{x}, 0), t_0 + \frac{d}{2}, \frac{d}{2}\big)$
  $v_{\text{teach}} \leftarrow \tilde{v} + \bar{v}$
  Define student augmented drift (full-step):
    $g_\theta^{(d)}(x, \ell, \tau) = \big(b_\theta(x, \tau, d), \textsc{Hutchinson}(\nabla_x b_\theta(x, \tau, d), \varepsilon)\big)$
  Student single step: $(\widehat{x}_{t+d}, v_{\text{stud}}) \leftarrow \Psi\big(g_\theta^{(d)}, (x, 0), t_0, d\big)$
  $\text{sum}_{\text{state}} \leftarrow \text{sum}_{\text{state}} + \|\widehat{x}_{t+d} - x_{t+d}^{\text{teach}}\|^2$
  $\text{sum}_{\text{vol}} \leftarrow \text{sum}_{\text{vol}} + (v_{\text{stud}} - v_{\text{teach}})^2$
end for
$\mathcal{L}_{\text{state}} \leftarrow \text{sum}_{\text{state}}/B; \quad \mathcal{L}_{\text{vol}} \leftarrow \text{sum}_{\text{vol}}/B$
return $\mathcal{L}_{\text{state}}$, $\mathcal{L}_{\text{vol}}$

For all methods we use the same cosine schedule for the noise scale. For N total steps and index $k \in \{0, \ldots, N\}$, define $t_k := k/N$. With hyperparameters $\sigma_{\max} > \sigma_{\min} > 0$, $s \geq 0$, and exponent $p \geq 1$, we use

$$\sigma(t_k) = \frac{1}{2}\big(\sigma_{\max} - \sigma_{\min}\big)\cos^p\Big(\frac{\pi}{2}\frac{1 + s - t_k}{1 + s}\Big) + \frac{1}{2}\sigma_{\min}. \tag{38}$$

Limitations

Our work has several limitations. First, our empirical study focuses on synthetic energy landscapes and low- to mid-dimensional Bayesian benchmarks (up to around 60 dimensions). We do not include molecular systems such as Lennard-Jones clusters or alanine dipeptide, which are standard in recent work on Boltzmann sampling.

Second, OSDS is built on top of a diffusion-based PIS-GRAD backbone trained at a relatively fine time discretization. Our method amortizes this training into one- or few-step inference, but it does not reduce the cost of learning the teacher itself, and we do not explore alternative backbones or simulation-free training.

Finally, our deterministic-flow importance weights rely on estimating log-Jacobian divergences via Hutchinson-style trace estimators along the PF ODE. While our experiments suggest that this is numerically stable in the dimensionalities we consider, we do not provide a systematic study of variance and bias as the state dimension grows.