Few-Step Boltzmann Generators via Scalable Likelihood Flow Maps

RuiKang OuYang[1,*] **Hanlin Yu**[2,†] **Xinyue Ai**[3,†] **Yutong He**[4]
Nicholas M. Boffi[4] **Pradeep Ravikumar**[4] **José Miguel Hernández-Lobato**[1]
Max Simchowitz[4,5] **Benjamin Kurt Miller**[6] **Omar Chehab**[4]
[1]University of Cambridge [2]University of Helsinki [3]Peking University
[4]Carnegie Mellon University [5]Amazon FAR [6]FAIR at Meta
[*]Correspondence: ro352@cam.ac.uk. [†]Equal contributions.

Abstract

Recent progress in flow-based generative modeling has led to models that output high-quality samples while using only a small number of function evaluations. However, at present, there is a lack of similar advances in estimating the model likelihood. In particular, most existing methods either rely on restrictive architectures that enable exact calculations, or use stochastic approximations such as Hutchinson's trace estimator that introduce substantial variance. In this work, we introduce SCALABLE LIKELIHOOD DISTILLATION OF FLOW MAPS (SCALLOP). SCALLOP builds on the recently proposed F2D2, a likelihood flow map model that can generate samples and their densities in a small number of function evaluations. While F2D2 uses Hutchinson's estimator during training, we introduce an alternative and more scalable likelihood distillation objective that is Hutchinson-free and admits a vectorized formulation. Empirically, we demonstrate the effectiveness of SCALLOP as a Boltzmann generator in molecular science, and further validate its benefits on image datasets. SCALLOP significantly reduces both training variance and training time while consistently improving performance compared to F2D2, and is competitive with the state-of-the-art while achieving up to a $10\times$ speedup in inference over the fastest baseline. Codes are available in https://github.com/tonyauyeung/scallop.

1 Introduction

While the generative modeling community has focused on improving the empirical quality of generated samples and the speed of generation (Song et al., 2023; Song and Dhariwal, 2024; Lu and Song, 2025; Geng et al., 2025; Boffi et al., 2026), numerous applications, notably in the sciences, require access to the model likelihood, e.g., for free energy estimation and enhanced sampling methods (Torrie and Valleau, 1977; Grubmüller, 1995; Laio and Parrinello, 2002; Barducci et al., 2008; Noé et al., 2019; Rissanen et al., 2025; Nam et al., 2025; Xie et al., 2026). To satisfy this requirement, early approaches restricted model architectures to ensure strict invertibility (Papamakarios et al., 2021), which inherently limits expressivity. Conversely, continuous flow- and diffusion-based models—along with their few-step distilled counterparts—offer far greater expressivity but do not trivially admit scalable likelihood evaluation. Computing their likelihoods remains bottlenecked by either prohibitively expensive Jacobian calculations (Chen et al., 2019; Rehman et al., 2026) or high-variance stochastic estimators (Grathwohl et al., 2018).

Current Approaches and Limitations Overall, the generative modeling literature has focused on learning a transport map from Gaussian samples to data samples, and then using the change-of-variables formula to compute the corresponding likelihood. Using that formula is often problematic: it requires computing costly quantities based on the neural network, during inference or training.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Likelihoods of normalizing flows (Rezende and Mohamed, 2015; Papamakarios et al., 2021) have limited expressivity and can be slow to compute. Normalizing flows parameterize transport maps as compositions of invertible neural network layers and compute likelihoods through the static change-of-variables formula, which requires evaluating their Jacobian log-determinants. To make this computation cheap, namely linear in the data dimension, the neural network has strong architectural constraints. These constraints can substantially reduce expressivity (Gao et al., 2020), particularly in applications with important symmetries such as molecular and materials modeling, where autoregressive layers are often unsuitable. For practical applications, expressive architectures require many neural network layers. For example, normalizing flows based on scalable transformer-based architectures (Zhai et al., 2025) have been successfully applied to atomic systems (Tan et al., 2026a,b; Schebek et al., 2025). Still, at inference time, they often suffer from a trade-off in speed between generating samples and computing likelihoods.

Computing likelihoods from continuous normalizing flows (CNFs) (Chen et al., 2019) typically suffers from high variance. CNFs remove the architectural constraints of normalizing flows by parameterizing the velocity field of a continuous transport process. Likelihood computation then requires integrating the divergence of the velocity field along trajectories through the instantaneous change-of-variables formula (Chen et al., 2019). In practice, this divergence is typically estimated using Hutchinson's trace estimator (Grathwohl et al., 2018), whose variance can become prohibitively large in high dimensions: maintaining constant precision may require computational budgets scaling quadratically with the data dimension. While variance-reduction techniques have been developed (Meyer et al., 2021; Liu et al., 2025), they are not mainstream.

Recent CNF distillation methods struggle to reconcile fast sampling with scalable and tractable likelihood evaluation. For instance, while flow map models (Boffi et al., 2025) successfully compress sampling into a single forward pass, they do not natively support tractable likelihood evaluation. Unlike standard normalizing flows, these models lack explicit architectural constraints and are not guaranteed to be bijective. To address this, subsequent works attempt to enforce model invertibility via regularization techniques (Draxler et al., 2024; Rehman et al., 2026); however, likelihood evaluation under these models remains computationally expensive—scaling cubically with the data dimension ($\mathcal{O}(d^3)$) due to the required Jacobian log-determinant calculation. To bypass this bottleneck, Ai et al. (2026) frame likelihood evaluation as a parallel distillation task, jointly amortizing both the sampling process and the likelihood estimation into a few network evaluations. Nevertheless, training this joint architecture still relies on Hutchinson-based divergence estimators, thereby inheriting their characteristic high variance. Alternatively, energy-based parameterizations of CNFs (Aggarwal et al., 2025; OuYang et al., 2026a; Plainer et al., 2026; He et al., 2026; Guth et al., 2026) enable likelihood estimation in a single network evaluation, but do not support few-step generation.

Contributions. In this work, we build on F2D2 (Ai et al., 2026) that extends flow maps into *likelihood flow maps* that distill both the sampling procedure and the likelihood computation into a single neural network evaluation. We propose SCALABLE LIKELIHOOD DISTILLATION OF FLOW MAPS (🫳 SCALLOP), which replaces the likelihood distillation loss in F2D2 with an alternative, based on reformulating the change-of-density along the flow. Our contributions are as follows:

- We introduce a *Hutchinson-free*, vectorized likelihood-distillation objective with the same population gradients as divergence matching for an exact marginal velocity field (Propositions 2 and 6). Lower training variability is supported empirically by Figures 1 and 7.
- Empirical improvements over the F2D2 model, on which SCALLOP is based, are substantial: SCALLOP can reduce training variance by up to $100\times$, converges faster, and consistently obtains better test set performance.
- In molecular domains, SCALLOP is much faster than state-of-the-art Boltzmann Generators at inference time: $10\times$ faster than normalizing flows (Tan et al., 2026a) while achieving better performance, and $100\times$ faster than the comparable FALCON (Rehman et al., 2026).
- These empirical gains translate to higher-dimensional image domains, where SCALLOP estimates more accurate likelihoods than F2D2, while being faster and more stable to train.

2 Background

We motivate our study with a common setting: A practitioner is interested in sampling from an unnormalized distribution π to approximate some macroscopic property like free energy differences, denoted O. π takes on a Boltzmann distribution and the observable O is computed in expectation:

$$\pi(x) = \frac{1}{Z}\exp(-U(x)), \quad \mathbb{E}_\pi[O(X)] = \int O(x)\pi(x)\,dx, \tag{1}$$

where $U : \mathbb{R}^d \to \mathbb{R}$ is the energy and Z is a normalizing constant. Since drawing unbiased samples from the Boltzmann distribution is difficult, Boltzmann Generators (Noé et al., 2019) fit a generative model p^θ to a potentially biased, accessible distribution p_{data} with similarities to π. Samples from p^θ are then used to approximate $\mathbb{E}_\pi[O(X)]$ with self-normalized importance sampling (SNIS):

$$\mathbb{E}_\pi[O(X)] = \mathbb{E}_{p^\theta}\left[\frac{\pi(X)}{p^\theta(X)}O(X)\right] \approx \frac{\sum_{n=1}^N w_n O(x_n)}{\sum_{m=1}^N w_m}, \quad \text{where } \begin{cases} w_n = \dfrac{\exp(-U(x_n))}{p^\theta(x_n)}, \\ x_n \sim p^\theta. \end{cases} \tag{2}$$

SNIS requires evaluating the probability density of the generated sample x under p^θ, commonly referred to as the model likelihood. Provided no ambiguity arises, we use the terms *sample density* and *model likelihood* interchangeably throughout this paper. Presently, the most expressive generative architectures, such as continuous-time flow models (Song et al., 2021; Lipman et al., 2023), require computationally expensive simulations to evaluate exact densities. Prior works either restrict themselves to less expressive models with tractable likelihoods, or rely on the Hutchinson trace estimator during training or inference. In this work, we present an alternative to the latter: *a novel distillation objective for likelihood flow maps that enables simultaneous sample generation and density approximation from continuous-time models without requiring Hutchinson's trace estimator at any stage*. Beyond our primary motivation, efficient likelihood estimation is of broad interest across machine learning, with applications extending to reward tilting and model composition.

Continuous Flows as Boltzmann Generators. A continuous flow is an Ordinary Differential Equation (ODE) that transports samples from a base distribution $p_0 = \mathcal{N}(0, I)$ to the data distribution $p_1 = p_{\text{data}}$, through a sequence of intermediate distributions $(p_t)_{t\in[0,1]}$. It can be used to generate samples and their densities by integrating

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_t \\ \log p_t^\theta(x_t) \end{bmatrix} = \begin{bmatrix} v_t^\theta(x_t) \\ -\nabla \cdot v_t^\theta(x_t) \end{bmatrix}, \quad \text{with } x_0 \sim p_0 \text{ and } \log p_0^\theta(x_0) = \log p_0(x_0), \tag{3}$$

where $\nabla\cdot$ is the divergence operator. The upper equation for generating samples is called the *Probability Flow ODE* (PF-ODE) and the lower equation for generating their log-densities is commonly called the *instantaneous change-of-variables* equation (Chen et al., 2019).

In many machine learning applications, the intermediate distributions $(p_t)_t$ are defined by the user, for example, using a *linear interpolation* between samples, $x_t = (1-t)x_0 + tx_1$ with $x_0 \sim p_0$ and $x_1 \sim p_1$. We will use this choice as a default throughout this paper. Flow matching (FM) (Lipman et al., 2023; Liu, 2022) learns a corresponding velocity field $v_t^\theta(x)$ and Diffusion Models (DMs) (Hyvärinen, 2005; Ho et al., 2020; Song et al., 2021) learn marginal scores $(\nabla \log p_t^\theta)_t$ which are a simple reparameterization of the velocity field, following

$$v_t(x_t) = \frac{x_t}{t} + \frac{1-t}{t}\nabla \log p_t(x_t). \tag{4}$$

Note that evaluating the density using Equation (3) requires the exact computation of $\nabla \cdot v_t^\theta$, which is infeasible in high-dimensional spaces. A popular approximation is to use Hutchinson's trace estimator (Hutchinson, 1990), which estimates $\nabla \cdot v_t^\theta(x_t) = \mathbb{E}_\epsilon[\epsilon^\top \nabla v_t^\theta(x_t)\epsilon]$ with a zero-mean random vector ϵ, independent of x_t, satisfying $\mathbb{E}[\epsilon\epsilon^\top] = I$, where $\nabla v_t^\theta(x_t)\epsilon$ can be efficiently computed using the Jacobian-Vector Product (JVP) operation. However, this estimator suffers from high variance, and the JVP still introduces some computational overhead.

Flow Maps for Few-step Sampling. Generating samples and their densities by numerical integration of Equation (3) requires hundreds of function evaluations. To reduce this computational

inefficiency, flow maps (Frans et al., 2025; Boffi et al., 2025; Geng et al., 2025) directly integrate the *sample generation* given by the upper part of Equation (3). In practice, flow map models are neural networks, trained to satisfy the consistency constraint

$$\Phi(x_t, t, s) = x_t + \int_t^s v_\tau(x_\tau)\mathrm{d}\tau = x_s. \tag{5}$$

Training objectives for the flow map are decomposed into two terms. For $t = s$, the flow map satisfies $\Phi(x_t, t, t) = x_t$, and matching $\partial_s \Phi(x_t, t, s)|_{s=t} = v_t(x_t)$ gives the classic FM objective; for $t < s$, the consistency is imposed via (1) the Lagrangian equation: $\mathrm{d}_s \Phi(x_t, t, s) = v_s(x_s)$, (2) the Eulerian equation: $\mathrm{d}_t \Phi(x_t, t, s) = 0$, or (3) the semigroup property: $\Phi(\Phi(x_t, t, r), r, s) = \Phi(x_t, t, s)$ for any $t < r < s$. For a trained flow map with parameter θ, the generation process is solved by iteratively applying $\hat{x}_{t_{i+1}} = \Phi_\theta(\hat{x}_{t_i}, t_i, t_{i+1})$ with $0 = t_0 < ... < t_N = 1$ and $\hat{x}_{t_0} \sim p_0$, involving $N \ll 1000$ function evaluations (NFEs).

Extending flow map models so that they simultaneously integrate the *likelihood generation* given by the bottom part of Equation (3) was recently proposed by Ai et al. (2026).

3 Scalable Likelihood Distillation in Flow Maps

We focus on learning a flow map-based, few-step generative model with co-evaluation of the model likelihood. Following the terminology of Ai et al. (2026), we call such a family *likelihood flow maps* and introduce a scalable training objective after reviewing the recent `F2D2` (Ai et al., 2026).

Throughout the paper, we use the linear interpolation $x_t = (1-t)x_0 + tx_1$ with $x_0 \sim \mathcal{N}(0, I)$ to construct the corrupted samples. This defines a sequence of marginal distributions $(p_t)_t$ that could be induced by a velocity field $(v_t)_t$ through solving the ODE $\mathrm{d}x_t = v_t(x_t)\mathrm{d}t$. Although v_t is typically intractable, we assume a distillation setting with access to an approximate velocity field v_t^ϕ. We omit ϕ for simplicity and assume $v_t^\phi \approx v_t$.

3.1 Likelihood Flow Maps Enable Few-Step Likelihood Evaluation

We first define the *likelihood flow map*, which is an operator that outputs both the generated samples and the associated log-density change jointly.

Definition 1 (Likelihood Flow Map) *Given a density-augmented PF-ODE as described in Equation (3), the likelihood flow map $\Psi : \mathbb{R}^d \times [0,1]^2 \to \mathbb{R}^{d+1}$ is the solution operator that maps any state at time t to its corresponding state at time s, with $s > t$, as well as the log-density change:*

$$\Psi(x_t, t, s) = \begin{bmatrix} x_t \\ 0 \end{bmatrix} + \int_t^s \begin{bmatrix} v_\tau(x_\tau) \\ -\nabla \cdot v_\tau(x_\tau) \end{bmatrix} \mathrm{d}\tau = \begin{bmatrix} x_s \\ \log p_s(x_s) - \log p_t(x_t) \end{bmatrix}. \tag{6}$$

Ai et al. (2026) propose to parameterize the likelihood flow map through the skip-connection mechanism

$$\Psi_\theta(x_t, t, s) = \begin{bmatrix} x_t \\ 0 \end{bmatrix} + (s-t) \begin{bmatrix} u_\theta(x_t, t, s) \\ D_\theta(x_t, t, s) \end{bmatrix} \Longrightarrow \begin{cases} u_\theta(x_t, t, t) = v_t(x_t) \\ D_\theta(x_t, t, t) = -\nabla \cdot v_t(x_t) \end{cases}, \tag{7}$$

where u_θ and D_θ are neural networks that share a subset of their parameters. When $s \to t$, this formulation naturally recovers the instantaneous velocity and instantaneous log-density-change. For clarity, we will use $f_\theta(x_t, t, s) = \begin{bmatrix} u_\theta(x_t, t, s) \\ D_\theta(x_t, t, s) \end{bmatrix}$ and $b_t(x_t) = \begin{bmatrix} v_t(x_t) \\ -\nabla \cdot v_t(x_t) \end{bmatrix}$.

3.2 How to Train a Likelihood Flow Map?

According to the instantaneous behavior as described by Equation (7), training a likelihood flow map (Ai et al., 2026) is analogous to training a standard flow map (Boffi et al., 2025): we simply allocate an extra dimension for the log-density. Under the skip-connection parameterization, training a likelihood flow map is essentially learning to satisfy $(s-t)f_\theta(x_t, t, s) \approx \int_t^s b(x_\tau, \tau)\mathrm{d}\tau$. The training objective integrates the following *three losses*:

$$\mathcal{L}_{\mathrm{F2D2}}(\theta) = \int_0^1 w_v(t)\mathcal{L}_v(\theta; t) + w_{\mathrm{DM}}(t)\mathcal{L}_{\mathrm{DM}}(\theta; t)\mathrm{d}t + \int_0^1 \int_0^s w(t, s)\mathcal{L}_{\mathrm{SD}}(\theta; t, s)\mathrm{d}t\mathrm{d}s, \tag{8}$$

where $w : [0,1]^2 \to \mathbb{R}^+$ and $w_v, w_{\mathrm{DM}} : [0,1] \to \mathbb{R}^+$ are weighting functions. The first two losses $\mathcal{L}_v$ and $\mathcal{L}_{\mathrm{DM}}$ are instantaneous losses written as

$$\text{(Flow Matching)} \quad \mathcal{L}_v(\theta; t) = \mathbb{E}_{x_0,x_1}\Big[\|u_\theta(x_t,t,t) - v_t(x_t)\|^2\Big], \tag{9}$$

$$\text{(Divergence Matching)} \quad \mathcal{L}_{\mathrm{DM}}(\theta; t) = \mathbb{E}_{x_0,x_1}\Big[|D_\theta(x_t,t,t) + \nabla \cdot v_t(x_t)|^2\Big], \tag{10}$$

with $x_t = (1-t)x_0 + tx_1$. The third loss $\mathcal{L}_{\mathrm{SD}}$ is called the self-distillation loss and has three different instantiations: (1) the Lagrangian equation: $\mathrm{d}_s\Phi(x_t,t,s) = v_s(x_s)$, (2) the Eulerian equation: $\mathrm{d}_t\Phi(x_t,t,s) = 0$, or (3) the semigroup property: $\Phi(\Phi(x_t,t,r),r,s) = \Phi(x_t,t,s)$ for any $t < r < s$. Adapted to the likelihood flow map setting, these three consistency constraints read:

(1) the Lagrangian self-distillation (LSD), *i.e.* $\mathrm{d}_s\Psi(x_t,t,s) = b_s(x_s)$

$$\mathcal{L}_{\mathrm{LSD}}(\theta; t, s) = \mathbb{E}_{x_0,x_1}\Big[\big\|f_\theta(x_t,t,s) + (s-t)\mathrm{sg}\big(\partial_s f_\theta(x_t,t,s)\big) - \mathrm{sg}\big(f_\theta(x_s,s,s)\big)\big\|^2\Big]; \tag{11}$$

(2) the Eulerian self-distillation (ESD), *i.e.* $\mathrm{d}_t\Psi(x_t,t,s) = 0$

$$\mathcal{L}_{\mathrm{ESD}}(\theta; t, s) = \mathbb{E}_{x_0,x_1}\Big[\Big\|f_\theta(x_t,t,s) - \mathrm{sg}\big(f_\theta(x_t,t,t)\big) - (s-t)\mathrm{sg}\big(\nabla f_\theta(x_t,t,s)\cdot b_t(x_t) + \partial_t f_\theta(x_t,t,s)\big)\Big\|^2\Big]; \tag{12}$$

(3) the progressive self-distillation (PSD), *i.e.* $\Psi(\Psi(x_t,t,u),u,s) = \Psi(x_t,t,s)\ \forall u \in [t,s]$

$$\mathcal{L}_{\mathrm{PSD}}(\theta; t, s) = \int_t^s \mu(u)\mathbb{E}_{x_0,x_1}\Big[\Big\|f_\theta(x_t,t,s) - \mathrm{sg}\Big(\frac{u-t}{s-t}f_\theta(x_t,t,u) + \frac{s-u}{s-t}f_\theta\big(\Psi_\theta(x_t,t,u),u,s\big)\Big)\Big\|^2\Big]\mathrm{d}u, \tag{13}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient and $\mu : [0,1] \to \mathbb{R}^+$ is another weighting function for u which is usually chosen to concentrate at the midpoint, $\mu(u) = \delta(u - \frac{t+s}{2})$ (Frans et al., 2025).

Ai et al. (2026) proposed F2D2, a likelihood flow map trained with the above objectives. Despite the mathematical simplicity, F2D2 relies on Hutchinson's trace estimator (Hutchinson, 1990) to estimate the divergence term $\nabla \cdot v_t$ in b_t to replace the computationally expensive $\mathcal{L}_{\mathrm{DM}}$ term, namely *Hutchinson Divergence Matching*:

$$\mathcal{L}_{\mathrm{HDM}}(\theta; t) = \mathbb{E}_{x_0,x_1}\mathbb{E}_\epsilon\left[\left|D_\theta(x_t,t,t) + \epsilon^\top \nabla v_t(x_t)\epsilon\right|^2\right], \tag{14}$$

where ϵ is independent of (x_0, x_1), has zero mean and satisfies $\mathbb{E}[\epsilon\epsilon^\top] = I$, as for $\mathcal{N}(0, I)$, and it can be shown that $\arg\min_\theta \mathcal{L}_{\mathrm{HDM}}(\theta; t) = \arg\min_\theta \mathcal{L}_{\mathrm{DM}}(\theta; t)$. Though $\nabla v_t(x_t)\epsilon$ can be implemented through a `JVP`, making it more efficient than the exact divergence calculation, it remains computationally expensive during training, especially in large-scale systems, and suffers from high variance.

3.3 Scalable Likelihood Distillation via Conditional Divergence Matching

The computational bottleneck in `F2D2` is introduced by having to estimate $-\nabla \cdot v_t(x_t) = \mathrm{d}_t \log p_t(x_t)$ whenever $b_t(x_t)$ needs to be calculated, which also suffers from high variance. Inspired by the success of Flow Matching (Lipman et al., 2023; Liu, 2022), which leverages a conditional objective to bypass the intractable velocity $v_t(x_t)$ during training, we raise the following question:

Is there any conditional objective that allows us to bypass the divergence calculation/estimation?

Our answer is positive. The key derivation is given by the chain rule: given x_t generated by a known velocity field v_t, the chain rule reads as

$$\mathrm{d}_t \log p_t(x_t) = \nabla \log p_t(x_t) \cdot v_t(x_t) + \partial_t \log p_t(x_t); \tag{15}$$

this identity is also used by Choi et al. (2021) for density ratio estimation.

Remark 1 *Alternatively, Equation (15) can be derived from the continuity equation as stated in Appendix A.3.*

It is well known that the Stein score $\nabla \log p_t$ can be learned via *Denoising Score Matching* (DSM, Hyvärinen, 2005; Vincent, 2011; Ho et al., 2020; Song et al., 2021), while the time score $\partial_t \log p_t$ can be learned via *Conditional time Score Matching* (C*t*SM, Yu et al., 2025; Guth et al., 2026). Combining both, we propose the *Conditional Divergence Matching* (CDM) objective, which bypasses the divergence operator:

$$\mathcal{L}_{\mathrm{CDM}}(\theta; t) = \mathbb{E}_{\xi, x_t}\left[\left| D_\theta(x_t, t, t) - v_t(x_t) \cdot \nabla \log p_t(x_t|\xi) - \partial_t \log p_t(x_t|\xi)\right|^2\right] \tag{16}$$

where ξ is a conditioning variable such that conditional time and space scores are available in closed form. For example, when using the interpolation $x_t = (1-t)x_0 + tx_1$, then conditioning on $\xi = x_1$ leaves only one source of randomness in x_0, which is Gaussian. It follows that $p_t(x_t|\xi)$ becomes tractable, and plugging it into the above yields the following proposition:

Proposition 2 (Conditional Divergence Matching) *Let $(p_t)_t$ be the laws of $x_t = (1-t)x_0 + tx_1$, where $x_0 \sim p_0 = \mathcal{N}(0, I)$ and $x_1 \sim p_1$ are independent. Let $v_t(x) = \mathbb{E}[x_1 - x_0 \mid x_t = x]$ be the corresponding marginal velocity field, and $D_\theta : \mathbb{R}^d \times [0,1]^2 \to \mathbb{R}$ the parameterized divergence predictor. The Conditional Divergence Matching objective*

$$\mathcal{L}_{\mathrm{CDM}}(\theta; t) = \mathbb{E}_{x_0, x_1}\left[\left| D_\theta(x_t, t, t) + \frac{x_0}{1-t} \cdot \big(v_t(x_t) - (x_1 - x_0)\big) - \frac{1}{1-t} d\right|^2\right] \tag{17}$$

equals the Divergence Matching objective $\mathcal{L}_{\mathrm{DM}}(\theta; t)$ up to a constant independent of θ, for all $t \in [0, 1)$ for which the losses are finite. Hence, $\nabla_\theta \mathcal{L}_{\mathrm{CDM}}(\theta; t) = \nabla_\theta \mathcal{L}_{\mathrm{DM}}(\theta; t)$.

Remark 3 *For an approximate pretrained velocity field, Appendix A.8 quantifies the resulting likelihood-estimation and population importance-sampling errors under suitable regularity and density-ratio assumptions.*

Remark 4 *The CDM objective can also be derived by first expressing the velocity as the posterior expectation over the conditional velocity, and then manipulating the divergence operator to obtain the divergence-operation-free training objective. Details are provided in Appendix A.4.*

Remark 5 *While the velocity v_t and the Stein score $\nabla \log p_t$ are connected by Equation (4) when $p_0 = \mathcal{N}(0, I)$ and $x_t = (1-t)x_0 + tx_1$, the mapping between the two becomes nontrivial for general stochastic interpolants, as stated in Equation (28). Hence, we treat $\nabla \log p_t$ as intractable. This can introduce additional variance to the training objective in the $\mathcal{N}(0, I)$ base cases. However, we found that it has little effect on performance in practice .*

3.4 Vectorized Divergence Matching

Since the divergence is the trace of the velocity Jacobian, predicting its diagonal entries provides a componentwise training signal whose sum recovers the scalar target. To regress vectors instead of scalars, we propose the following vectorized variants of the divergence matching objectives: (1) *Vectorized Divergence Matching* (DM-v), (2) *Vectorized Hutchinson Divergence Matching* (HDM-v), and (3) *Vectorized Conditional Divergence Matching* (CDM-v). In particular, the effective training target changes from the negative divergence $-\nabla \cdot v_t$ to the negative diagonal entries of the Jacobian $-\mathrm{diag}(\nabla v_t)$. For notational clarity, we use the symbolic notation $\nabla \odot v_t$ to denote $\mathrm{diag}(\nabla v_t)$. In particular, the divergence predictor is now parameterized to output a vector, i.e. $D_\theta : \mathbb{R}^d \times [0,1]^2 \to \mathbb{R}^d$:

$$\mathcal{L}_{\text{DM-v}}(\theta; t) = \mathbb{E}_{x_0, x_1}\left[\left\| D_\theta(x_t, t, t) + \nabla \odot v_t(x_t)\right\|^2\right], \tag{18}$$

$$\mathcal{L}_{\text{HDM-v}}(\theta; t) = \mathbb{E}_{x_0, x_1, \epsilon}\left[\left\| D_\theta(x_t, t, t) + \epsilon \odot \nabla v_t(x_t)\epsilon\right\|^2\right], \tag{19}$$

$$\mathcal{L}_{\text{CDM-v}}(\theta; t) = \mathbb{E}_{x_0, x_1}\left[\left\| D_\theta(x_t, t, t) + \frac{x_0}{1-t} \odot \big(v_t(x_t) - (x_1 - x_0)\big) - \frac{1}{1-t}\mathbf{1}_d\right\|^2\right]. \tag{20}$$

Proposition 6 (Vectorized Divergence Matching) *Let $(v_t)_t$ and $(p_t)_t$ be defined as in Proposition 2, and $D_\theta : \mathbb{R}^d \times [0,1]^2 \to \mathbb{R}^d$ be the parameterized **vectorized** divergence predictor. The vectorized learning objectives defined in Equations (18) to (20) are related as*

$$\arg\min_\theta \mathcal{L}_{\text{HDM-v}}(\theta; t) = \arg\min_\theta \mathcal{L}_{\text{DM-v}}(\theta; t) \quad \textit{and} \quad \nabla_\theta \mathcal{L}_{\text{CDM-v}}(\theta; t) = \nabla_\theta \mathcal{L}_{\text{DM-v}}(\theta; t), \tag{21}$$

for all $t \in [0, 1)$ for which the losses are finite. When the predictor family can represent the target, an optimal $\theta^* = \arg\min_\theta \mathcal{L}_{\text{DM-v}}(\theta; t)$ recovers

$$D_{\theta^*}(x_t, t, t) = -\nabla \odot v_t(x_t) \quad \textit{and} \quad \mathbf{1}_d \cdot D_{\theta^*}(x_t, t, t) = -\nabla \cdot v_t(x_t). \tag{22}$$

The same vectorization technique is also used by Ou et al. (2025) for learning the covariance of the denoising kernel in Diffusion models, and Yu et al. (2025) for learning the time score.

Connection to Second-Order Score Matching. Our CDM-v objective is related to the second-order Score Matching objective (Meng et al., 2021). At $0 < t < 1$, recall the relationship between the velocity and score described in Equation (4):

$$v_t(x_t) = \frac{x_t}{t} + \frac{1-t}{t}\nabla \log p_t(x_t) \Longrightarrow \nabla v_t(x_t) = \frac{I}{t} + \frac{1-t}{t}\nabla^2 \log p_t(x_t). \tag{23}$$

With access to $\nabla \log p_t$, $\nabla^2 \log p_t$ can be learned with a conditional loss (Meng et al., 2021). Hence, learning the diagonal entries of $\nabla^2 \log p_t(x_t)$ only is equivalent to our vectorized conditional divergence matching up to an independent constant. We leave the detailed background and discussion in Appendix A.7.

3.5 SCALLOP

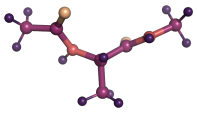
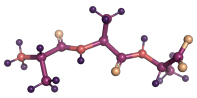
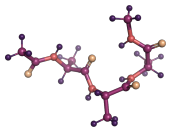
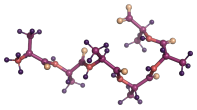
Combining the training objectives from Section 3.2 with our *Conditional Divergence Matching* objective (Section 3.3) and the vectorization technique (Section 3.4), we propose `SCALLOP`, a scalable and efficient pipeline for training a likelihood flow map. The complete training objective is written in Equation (24). Again, we consider a distillation setting following Ai et al. (2026), where a pretrained velocity field $v_t^\phi \approx v_t$ is available. When no ambiguity arises, we write v_t for v_t^ϕ for notational simplicity. We focus on modeling molecular systems, where specific symmetries must be respected. Related works are discussed in Section 1 and expanded on in Appendix B.

Parameterization. We use the vectorized objective, CDM-v, to learn the log-density-change part of the likelihood flow map. Internally, we parameterize the vectorized map $\Psi_\theta^{\text{vec}}(x_t, t, s) = [x_t; \mathbf{0}_d] + (s-t)f_\theta(x_t, t, s)$, where $f_\theta(x_t, t, s) = [u_\theta(x_t, t, s); D_\theta(x_t, t, s)]$, $u_\theta, D_\theta : \mathbb{R}^d \times [0, 1]^2 \to \mathbb{R}^d$, $x_t + (s-t)u_\theta(x_t, t, s)$ gives the generated samples, and $(s-t)D_\theta(x_t, t, s)$ gives the vectorized log-density change. Summing its components recovers the scalar log-density change in Ψ_θ. We employ shared encoding layers for feature extraction, while specifying two readout heads for the samples and log-density-change respectively, analogous to Ai et al. (2026). As a byproduct of the vectorization trick, we employ the same architecture for these two readout heads. In particular, we follow Rehman et al. (2026) to use the standard Diffusion Transformer block (DiT, Peebles and Xie, 2023) with an additional time embedding head to adapt to the flow map setting. We use 6 shared DiT blocks, and 3 DiT blocks for each of the sample readout head and log-density-change readout head.

Enforcing Soft Symmetries for Molecular Systems. The generated samples should be equivariant to rotation and translation of the input, namely SE(3)-equivariant, and the predicted log-density-change should be SE(3)-invariant. Following Tan et al. (2026a); Rehman et al. (2026), we subtract the mean of the data to enforce translational equivariance, while employing random rotation to softly impose equivariance of the rotations. The scalar divergence target is rotation-invariant. We sum the coordinatewise predictions to estimate it and use rotation augmentation to encourage this symmetry. As explained in Appendix A.6, the Jacobian diagonal is not generally an ordinary rotation-equivariant vector, and augmentation alone does not guarantee exact symmetry.

Learning Objective and Stabilizing Training. Following Boffi et al. (2025); Ai et al. (2026), we use learned weights $\exp(-w_\psi(t, s))$ for self-distillation and $\exp(-w_\psi(t, t))$ for flow matching, where ψ denotes a shallow scalar-valued neural network. However, for the divergence matching part, the objective has high variance when $t \to 1$ as studied by Song et al. (2021); Yu et al. (2025) as well as a singularity at $t = 1$. To stabilize training, we uniformly sample t from $[0, 1 - \text{eps}]$ with $\text{eps} = 10^{-5}$, as well as employ a weighting $\exp(-w_\psi(t, t))(1-t)$ for the divergence matching part. For the self-distillation part, (s, t) is sampled uniformly on the upper triangle $t < s$ of the square

Table 1: Results on ALA-N systems, with $N = 2, 3, 4, 6$. Results are obtained from three random seeds with 2×10^5 samples, reported as `mean` $\pm$ `std`. The best results are in **bold**, and the second-best results are underlined. Values of SBG-IS, FALCON, and FALCON-A are from Rehman et al. (2026).

		SBG-IS	FALCON	FALCON-A	F2D2	SCALLOP
	ESS ↑	0.030±0.012	0.067±0.013	**0.097**±0.007	0.066±0.001	0.088±0.001
ALA-2	$\mathcal{E}$-$\mathcal{W}_2$ ↓	0.873±0.338	**0.225**±0.104	0.512±0.038	0.958±0.114	0.555±0.116
($d = 66$)	$\mathbb{T}$-$\mathcal{W}_2$ ↓	0.439±0.129	0.402±0.021	**0.180**±0.005	0.383±0.022	0.310±0.027
	NFE	1	4	4	4	4
	ESS ↑	0.052±0.013	0.077±0.004	**0.104**±0.004	0.079±0.001	0.088±0.001
ALA-3	$\mathcal{E}$-$\mathcal{W}_2$ ↓	0.758±0.506	**0.544**±0.013	1.385±0.182	2.155±0.033	1.596±0.028
($d = 99$)	$\mathbb{T}$-$\mathcal{W}_2$ ↓	0.502±0.016	0.452±0.011	0.343±0.004	0.219±0.001	**0.212**±0.001
	NFE	1	8	8	8	8
	ESS ↑	0.046±0.014	0.055±0.003	**0.094**±0.007	0.053±0.001	0.067±0.001
ALA-4	$\mathcal{E}$-$\mathcal{W}_2$ ↓	1.068±0.495	**0.686**±0.047	2.929±0.068	4.209±0.066	1.545±0.033
($d = 126$)	$\mathbb{T}$-$\mathcal{W}_2$ ↓	0.969±0.067	**0.858**±0.077	1.094±0.034	1.435±0.023	1.399±0.039
	NFE	1	8	8	8	8
	ESS ↑	0.034±0.015	0.060±0.017	**0.077**±0.007	0.033±0.001	0.042±0.001
ALA-6	$\mathcal{E}$-$\mathcal{W}_2$ ↓	1.021±0.239	**0.892**±0.311	1.211±0.105	10.140±0.065	5.845±0.101
($d = 189$)	$\mathbb{T}$-$\mathcal{W}_2$ ↓	1.431±0.085	1.256±0.132	1.163±0.112	1.143±0.039	**1.120**±0.014
	NFE	1	16	16	16	16

$[0, 1 - \text{eps}]^2$. Overall, the practical training objective reads

$$\mathcal{L}_{\text{SCALLOP}}(\theta) = \int_0^1 \Big[e^{-w_\psi(t,t)}\big(\mathcal{L}_v(\theta;t) + (1-t)\mathcal{L}_{\text{CDM-v}}(\theta;t)\big) + w_\psi(t,t)\Big]\mathrm{d}t + \int_0^1 \int_0^s \Big[e^{-w_\psi(t,s)}\mathcal{L}_{\text{SD}}(\theta;t,s) + w_\psi(t,s)\Big]\mathrm{d}t\mathrm{d}s. \quad (24)$$

Fast Generation and Likelihood Evaluation. Given a discrete time schedule $0 = t_0 < ... < t_K = 1$, inference in SCALLOP is analogous to that in a standard flow map, with a small amount of additional overhead due to calling the log-density-change readout head. In particular, we start by sampling $x_0 \sim p_0$ and evaluating $\log p_0(x_0)$. We then iteratively update $x_{t_{i+1}} = x_{t_i} + (t_{i+1} - t_i)u_\theta(x_{t_i}, t_i, t_{i+1})$ and $\log p_{t_{i+1}}(x_{t_{i+1}}) = \log p_{t_i}(x_{t_i}) + (t_{i+1} - t_i)\mathbf{1}_d \cdot D_\theta(x_{t_i}, t_i, t_{i+1})$; see Algorithm 1 for details. It is notable that, unlike methods with approximate invertibility (Draxler et al., 2024; Rehman et al., 2026), SCALLOP does not require calculating or estimating the log-determinant of the Jacobian of the generator for likelihood evaluation, but instead requires a single forward pass of the neural network. Overall, the few-step and high-order-free nature of SCALLOP enables efficient sample generation alongside density evaluation.

4 Experiments

Instantiation. We optimize Equation (24) to train SCALLOP. In particular, we use the LSD loss (Equation (11)) as the self-distillation loss, and leave the other alternatives for future work.

Molecular Systems. We conduct experiments on several alanine systems of varying lengths, including Alanine Dipeptide (ALA-2), Alanine Tripeptide (ALA-3), Alanine Tetrapeptide (ALA-4), and Alanine Hexapeptide (ALA-6). Data are obtained from simulating Molecular Dynamics in implicit solvent, with the AMBER-14 force field. Details of the simulation are provided in Appendix C. Following Tan et al. (2026a); Rehman et al. (2026), models are trained on the biased dataset. The goal is to correct the model using self-normalized importance sampling, enabling it to sample equilibrium conformations corresponding to the Boltzmann distribution.

Baselines. We benchmark SCALLOP against the most promising discrete- and continuous-flow baselines identified by Rehman et al. (2026): SBG (Tan et al., 2026a) and FALCON (Rehman et al., 2026). Specifically, to better focus on likelihood evaluation, we compare against the SBG with standard SNIS (denoted as SBG-IS). Although an SMC-based variant (SBG-SMC) exists and can be seamlessly integrated with both FALCON and SCALLOP, we exclude it from our current benchmark to ensure a fair comparison with SNIS, leaving the exploration of SCALLOP-SMC for future work.

Metrics. We evaluate SCALLOP and the baselines using standard SNIS. To measure sampling efficiency, we report the Effective Sample Size (ESS). To quantify the effectiveness of the reweighted samples, we compute the 2-Wasserstein distance for both the distributions of energy ($\mathcal{E}$-$\mathcal{W}_2$) and torsional angle ($\mathbb{T}$-$\mathcal{W}_2$). Specifically, $\mathcal{E}$-$\mathcal{W}_2$ is sensitive to extreme energy values—often caused by atoms being too close to one another—and thus reflects local structural refinements. Conversely, because torsional angles act as the primary collective variables capturing the most information in alanine systems, $\mathbb{T}$-$\mathcal{W}_2$ measures global structural quality. Further details are provided in Appendix C.

4.1 SCALLOP as a Boltzmann Generator

Comparable Performance against SOTA. The performance of SCALLOP and the other baselines is reported in Table 1. Across different systems, SCALLOP consistently outperforms F2D2 across all considered metrics, while achieving highly competitive results against the state-of-the-art FALCON and FALCON-A. Specifically, SCALLOP excels at capturing the global conformational space, achieving the best $\mathbb{T}$-$\mathcal{W}_2$ on the ALA-3 and ALA-6 systems. Furthermore, it consistently yields a high ESS, indicating substantial overlap with the target distribution and a strong resistance to mode collapse—a conclusion further corroborated by its strong $\mathbb{T}$-$\mathcal{W}_2$ performance.

While FALCON generally achieves a lower $\mathcal{E}$-$\mathcal{W}_2$, SCALLOP drastically improves upon F2D2, nearly halving the $\mathcal{E}$-$\mathcal{W}_2$ error in larger systems like ALA-4 and ALA-6. More importantly, higher $\mathcal{E}$-$\mathcal{W}_2$ values do not undermine the overall validity of the drawn samples, as the Wasserstein-2 distance is notoriously sensitive to energy outliers caused by microscopic structural deviations. The low $\mathbb{T}$-$\mathcal{W}_2$ demonstrates that SCALLOP correctly captures the global torsional angle structure, confirming that the macroscopic metastable states and their statistical weights are accurately recovered. Because the underlying conformational macrostates are correct, the remaining local structural noise that inflates the $\mathcal{E}$-$\mathcal{W}_2$ can be efficiently mitigated by applying a computationally cheap local relaxation to align the local structures.

Improved Training Stability against F2D2. Figure 1 compares training on ALA-2. The left panel shows loss-gradient norms, with variance estimated via a sliding window; the middle panel shows the Signal-to-Noise Ratio (SNR) of the loss gradients, defined as $10\log_{10}(|\text{mean}|/\text{std})$. SCALLOP exhibits consistently higher log SNR, indicating greater training stability. The right panel reports ESS against training GPU-hours, showing convergence to a superior optimum.

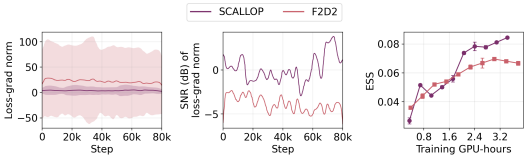


Figure 1: Training behavior comparison between SCALLOP and F2D2 on ALA-2.

Efficient Training and Likelihood Evaluation. We evaluate training and inference times on a single NVIDIA RTX PRO 6000 GPU, detailed in Figure 2. The left panel reports training times evaluated at a batch size of 1024 with the same number of network updates. By avoiding the JVP used in Hutchinson's trace estimator, SCALLOP achieves a roughly 10% speedup compared to F2D2.

The right panel reports the time required to generate 2×10^5 samples alongside their densities. Due to the memory-intensive full Jacobian computations, FALCON requires heavily restricted

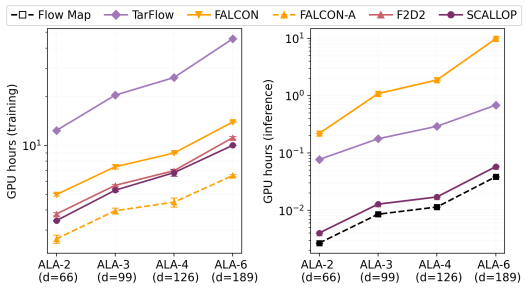


Figure 2: **(Left)** Training time comparison; **(Right)** Inference time comparison, for generating 2×10^5 samples with their densities.

Table 2: Evaluation on CelebA-64. The reference flow-matching model achieves BPD 1.75 with 1024 Euler steps and FID 2.48 with 200 Euler steps. $|\Delta|$ denotes the absolute deviation from reference BPD and Err denotes the mean absolute error against the reference per-sample BPD estimates. All models are trained with the LSD loss (Equation (11)).

Steps	Flow map				F2D2				SCALLOP			
	NLL	$\lvert\Delta\rvert\downarrow$	Err $\downarrow$	FID $\downarrow$	NLL	$\lvert\Delta\rvert\downarrow$	Err $\downarrow$	FID $\downarrow$	NLL	$\lvert\Delta\rvert\downarrow$	Err $\downarrow$	FID $\downarrow$
1	-69.83	71.58	71.64	12.96	1.64	**0.11**	0.53	6.94	1.60	0.15	**0.17**	**6.91**
2	-32.72	34.47	34.46	6.32	1.73	**0.02**	0.33	3.86	1.69	0.06	**0.12**	**3.81**
4	-14.89	16.64	16.59	4.04	1.75	**0.00**	0.24	2.75	1.75	**0.00**	**0.12**	**2.70**
8	-6.78	8.53	8.45	3.33	1.64	0.11	0.18	2.41	1.68	**0.07**	**0.10**	**2.37**

batch sizes $(1000, 1000, 500, 100)$ for ALA-$(2, 3, 4, 6)$, respectively, leading to inference costs that scale steeply with dimensionality. Conversely, SCALLOP and F2D2 estimate likelihoods via appended DiT blocks and possess nearly identical inference costs. This approach bypasses the Jacobian bottleneck, supports large batch sizes (10^4 for all systems), and adds modest overhead over a baseline Flow Map that lacks likelihood evaluation.

4.2 Effectiveness and Scalability beyond Molecules

To test whether the benefits of SCALLOP generalize beyond molecular data, we evaluate it on unconditional image generation using CelebA-64 (Liu et al., 2015). Following the F2D2 evaluation protocol, we report FID for sample quality and BPD, i.e., negative log-likelihood measured in bits per dimension, for likelihood estimation. We further report the absolute deviation from the 1024-step teacher BPD and the mean absolute per-sample BPD error with respect to the same teacher reference. All metrics are evaluated using 1, 2, 4, and 8 Euler steps for both sampling and likelihood computation. As shown in Table 2, SCALLOP achieves BPD and FID comparable to F2D2 across all steps, while consistently reducing the per-sample BPD error. Figure 7 further shows that SCALLOP has more stable training dynamics, with lower variance in both training loss and gradient norm compared with F2D2 over 60k training steps. Moreover, its loss computation takes only 83.1% of the wall-clock time of F2D2, demonstrating improved computational efficiency. Additional details are provided in Appendix E.

5 Discussion

In conclusion, we proposed SCALABLE LIKELIHOOD DISTILLATION OF FLOW MAPS (SCALLOP), an efficient procedure for training few-step models to jointly generate samples and their densities. This capability allows SCALLOP to serve as a Boltzmann generator for sampling from unnormalized densities, where it is competitive with the state-of-the-art while being significantly faster to evaluate at inference time.

Limitations. Despite its empirical success, SCALLOP has two main limitations. First, exact equivalence between conditional and marginal divergence matching assumes a perfectly learned velocity field. Under the regularity and density-ratio assumptions in Appendix A.8, Proposition 9 bounds log-density error along the teacher's continuous-time trajectories, separating teacher error from error in fitting the instantaneous likelihood head. Theorem 10 bounds the difference in observable expectations between the interpolation and teacher marginals. Corollary 11 bounds the bias of a population importance-weighted expectation targeting p_t when the proposal density is estimated by a fitted head. These guarantees do not cover finite-sample self-normalization or errors introduced by few-step distillation. Second, while SCALLOP can generate samples and their densities in 1 step, it practically uses $\sim$10 steps for high-dimensional tasks.

Acknowledgments and Disclosure of Funding

RKOY acknowledges the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/L016516/1 for the University of Cambridge Centre for Doctoral Training, the Cambridge Centre

for Analysis. HY acknowledges the research environment provided by ELLIS Institute Finland and was supported by the Research Council of Finland Flagship programme: Finnish Center for Artificial Intelligence FCAI. PR acknowledges the support of AFRL and DARPA via FA8750-23-2-1015, ONR via N00014-23-1-2368, and NSF via IIS-1955532. JMHL acknowledges support from EPSRC funding under grant EP/Y028805/1. JMHL also acknowledges support from a Turing AI Fellowship under grant EP/V023756/1.

This work was performed using HPC resources from GENCI–IDRIS (AD011015234R1). This project acknowledges the resources provided by the Cambridge Service for Data-Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (www.csd3.cam.ac.uk), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1), and DiRAC funding from the Science and Technology Facilities Council (www.dirac.ac.uk).

Competing interests. The authors declare no competing financial interests beyond the funding and affiliations disclosed above.

References

R. Aggarwal, J. Chen, N. M. Boffi, and D. R. Koes. Boltznce: Learning likelihoods for boltzmann generation with stochastic interpolants and noise contrastive estimation, 2025. URL https://arxiv.org/abs/2507.00846.

X. Ai, Y. He, A. Gu, R. Salakhutdinov, J. Kolter, N. Boffi, and M. Simchowitz. Joint distillation for fast likelihood evaluation and sampling in flow-based models. In *International Conference on Learning Representations*, 2026.

M. Albergo, N. M. Boffi, and E. Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

A. Barducci, G. Bussi, and M. Parrinello. Well-tempered metadynamics: a smoothly converging and tunable free-energy method. *Physical review letters*, 100(2):020603, 2008.

N. M. Boffi, M. S. Albergo, and E. Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation, 2025. URL https://arxiv.org/abs/2505.18825.

N. M. Boffi, M. S. Albergo, and E. Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. In *Annual Conference on Neural Information Processing Systems*, 2026.

V. D. Bortoli, M. Hutchinson, P. Wirnsberger, and A. Doucet. Target score matching, 2024. URL https://arxiv.org/abs/2402.08667.

R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations, 2019. URL https://arxiv.org/abs/1806.07366.

K. Choi, C. Meng, Y. Song, and S. Ermon. Density ratio estimation via infinitesimal classification. *arXiv e-prints*, 2021.

M. Dibak, L. Klein, A. Krämer, and F. Noé. Temperature steerable flows and Boltzmann generators, 2021. URL https://arxiv.org/abs/2108.01590.

F. Draxler, P. Sorrenson, L. Zimmermann, A. Rousselot, and U. Köthe. Free-form flows: Make any architecture a normalizing flow, 2024. URL https://arxiv.org/abs/2310.16624.

K. Frans, D. Hafner, S. Levine, and P. Abbeel. One step diffusion via shortcut models, 2025. URL https://arxiv.org/abs/2410.12557.

R. Gao, E. Nijkamp, D. P. Kingma, Z. Xu, A. M. Dai, and Y. N. Wu. Flow contrastive estimation of energy-based models. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 7515–7525, 2020. doi: 10.1109/CVPR42600.2020.00754.

Z. Geng, M. Deng, X. Bai, J. Z. Kolter, and K. He. Mean flows for one-step generative modeling, 2025. URL https://arxiv.org/abs/2505.13447.

W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models, 2018. URL https://arxiv.org/abs/1810.01367.

H. Grubmüller. Predicting slow structural transitions in macromolecular systems: Conformational flooding. *Physical Review E*, 52(3):2893, 1995.

F. Guth, Z. Kadkhodaie, and E. P. Simoncelli. Learning normalized image densities via dual score matching, 2026. URL https://arxiv.org/abs/2506.05310.

M. U. Gutmann and A. Hyvärinen. Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics. *Journal of Machine Learning Research*, 13(11): 307–361, 2012. URL http://jmlr.org/papers/v13/gutmann12a.html.

J. He, J. M. Hernández-Lobato, Y. Du, and F. Vargas. Rne: plug-and-play diffusion inference-time control and energy-based training, 2026. URL https://arxiv.org/abs/2506.05668.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

M. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 19(2):433–450, 1990. doi: 10.1080/03610919008812866. URL https://doi.org/10.1080/03610919008812866.

A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. URL http://jmlr.org/papers/v6/hyvarinen05a.html.

L. Kish. Confidence intervals for clustered samples. *American Sociological Review*, 22(2):154–165, 1957.

L. Klein, A. Krämer, and F. Noe. Equivariant flow matching. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 59886–59910. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/bc827452450356f9f558f4e4568d553b-Paper-Conference.pdf.

J. Köhler, L. Klein, and F. Noe. Equivariant flows: Exact likelihood generative learning for symmetric densities. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5361–5370. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/kohler20a.html.

J. Köhler, A. Krämer, and F. Noé. Smooth normalizing flows. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/167434fa6219316417cd4160c0c5e7d2-Abstract.html.

A. Laio and M. Parrinello. Escaping free-energy minima. *Proceedings of the national academy of sciences*, 99(20):12562–12566, 2002.

Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Q. Liu. Rectified flow: A marginal preserving approach to optimal transport, 2022. URL https://arxiv.org/abs/2209.14577.

X. Liu, H. Du, W. Deng, and R. Zhang. Optimal stochastic trace estimation in generative modeling. In Y. Li, S. Mandt, S. Agrawal, and E. Khan, editors, *Proceedings of the 28th international conference on artificial intelligence and statistics*, volume 258 of *Proceedings of machine learning research*, pages 4600–4608. PMLR, May 2025. URL https://proceedings.mlr.press/v258/liu25k.html.

Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

C. Lu and Y. Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LyJi5ugyJx.

C. Meng, Y. Song, W. Li, and S. Ermon. Estimating High Order Gradients of the Data Distribution by Denoising. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25359–25369. Curran Associates, Inc., 2021.

R. A. Meyer, C. Musco, C. Musco, and D. P. Woodruff. Hutch++: Optimal stochastic trace estimation. In *2021 Symposium on Simplicity in Algorithms (SOSA)*, pages 142–155. Society for Industrial and Applied Mathematics, 2021. doi: 10.1137/1.9781611976496.16.

J. Nam, B. Máté, A. P. Toshev, M. Kaniselvan, R. Gómez-Bombarelli, R. T. Chen, B. Wood, G.-H. Liu, and B. K. Miller. Enhancing diffusion-based sampling with molecular collective variables. *arXiv preprint arXiv:2510.11923*, 2025.

F. Noé, S. Olsson, J. Köhler, and H. Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019. doi: 10.1126/science.aaw1147. tex.eprint: https://www.science.org/doi/pdf/10.1126/science.aaw1147.

Z. Ou, M. Zhang, A. Zhang, T. Z. Xiao, Y. Li, and D. Barber. Improving probabilistic diffusion models with optimal diagonal covariance matching, 2025. URL https://arxiv.org/abs/2406.10808.

R. OuYang, L. Grenioux, and J. M. Hernández-Lobato. A diffusive classification loss for learning energy-based generative models, 2026a. URL https://arxiv.org/abs/2601.21025.

R. OuYang, B. Qiang, and J. M. Hernández-Lobato. Bnem: A boltzmann sampler based on bootstrapped noised energy matching, 2026b. URL https://arxiv.org/abs/2409.09787.

G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57): 1–64, 2021.

W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4195–4205, October 2023.

M. Plainer, H. Wu, L. Klein, S. Günnemann, and F. Noé. Consistent sampling and simulation: Molecular dynamics with energy-based diffusion models, 2026. URL https://arxiv.org/abs/2506.17139.

D. Rehman, T. Akhound-Sadegh, A. Gazizov, Y. Bengio, and A. Tong. FALCON: Few-step accurate likelihoods for continuous flows. In *The fourteenth international conference on learning representations*, 2026.

D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

S. Rissanen, R. OuYang, J. He, W. Chen, M. Heinonen, A. Solin, and J. M. Hernández-Lobato. Progressive tempering sampler with diffusion, 2025. URL https://arxiv.org/abs/2506.05231.

M. Schebek, F. Noé, and J. Rogal. Scalable boltzmann generators for equilibrium sampling of large-scale materials, 2025. URL https://arxiv.org/abs/2509.25486.

Y. Song and P. Dhariwal. Improved techniques for training consistency models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=WNzy9bRDvG.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

Y. Song, P. Dhariwal, M. Chen, and I. Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32211–32252. PMLR, 2023.

C. B. Tan, A. J. Bose, C. Lin, L. Klein, M. M. Bronstein, and A. Tong. Scalable equilibrium sampling with sequential boltzmann generators, 2026a. URL https://arxiv.org/abs/2502.18462.

C. B. Tan, M. Hassan, L. Klein, S. Syed, D. Beaini, M. M. Bronstein, A. Tong, and K. Neklyudov. Amortized sampling with transferable normalizing flows, 2026b. URL https://arxiv.org/abs/2508.18175.

A. Tong, K. Fatras, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, and Y. Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. Expert Certification.

G. M. Torrie and J. P. Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of computational physics*, 23(2):187–199, 1977.

P. Vincent. A Connection Between Score Matching and Denoising Autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

Y. Xie, L. Winkler, L. Sun, S. Lewis, A. E. Foster, J. J. Luna, T. Hempel, M. Gastegger, Y. Chen, I. Zaporozhets, et al. Enhanced diffusion sampling: Efficient rare event sampling and free energy calculation with diffusion models. *arXiv preprint arXiv:2602.16634*, 2026.

H. Yu, A. Klami, A. Hyvarinen, A. Korba, and O. Chehab. Density ratio estimation with conditional probability paths. In *International Conference on Machine Learning*, 2025.

H. Yu, R. OuYang, P. Kaushik, A. Klami, M. U. Gutmann, and O. Chehab. Learning energy-based models from stochastic interpolants using spatiotemporal differences, 2026. URL https://arxiv.org/abs/2605.26850.

S. Zhai, R. Zhang, P. Nakkiran, D. Berthelot, J. Gu, H. Zheng, T. Chen, M. A. Bautista, N. Jaitly, and J. Susskind. Normalizing flows are capable generative models, 2025. URL https://arxiv.org/abs/2412.06329.

Appendix

A Conditional Estimators for Flow-based Model's Likelihood: Proofs and Connections **16**

A.1 Desiderata for a Consistent Objective 17

A.2 Derivation through Total Derivative 17

A.3 Derivation through the Continuity Equation 18

A.4 Derivation through Manipulating the Trace Operator 18

A.5 Connection between Two Estimators, and Beyond 19

A.6 Vectorized Estimators 20

A.7 Connection with Higher-Order Score Matching 20

A.8 Error Quantification 22

B Expanded Discussion of Related Works **28**

C Experimental Details for Molecules **29**

C.1 Details of Training 29

C.2 Details of Inference 30

C.3 Details of Alanine Systems 30

C.4 Details of Metrics 31

D Additional Experimental Results for Molecules **31**

E Experimental Details for Images **31**

F Additional Experimental Results for Images **33**

Notation and Terminology. Throughout this appendix, we operate within the framework of generative modeling via continuous-time paths. Here, x_0 and x_1 denote the endpoint data variables, x_t represents the intermediate state at time $t \in [0,1]$, and z is the latent noise variable driving the process. Unless otherwise specified, conditional expectations of the form $\mathbb{E}[\cdot|x_t]$ are taken over the joint posterior distribution $p(x_0, x_1|x_t)$ of the endpoints given the intermediate state. At times with $\gamma_t > 0$, the noise z is uniquely determined once (x_0, x_1, x_t) are known, so expectations involving the noise (such as $\mathbb{E}[z|x_t]$) are implicitly evaluated under this same posterior.

We use "estimator" for both a conditional training target and its exact population conditional mean, distinguishing them when needed. Mini-batches of joint interpolation samples give Monte Carlo estimates of the training loss; they do not directly evaluate the posterior mean at a fixed x_t. The population identities below describe the target of conditional regression, rather than a claim that a finite trained network is unbiased.

A Conditional Estimators for Flow-based Model's Likelihood: Proofs and Connections

In this section, we provide conditional estimators for estimating the model density, i.e. likelihoods, of samples generated by the following ODE in $\mathbb{R}^d$

$$\mathrm{d}x_t = v_t(x_t)\mathrm{d}t, \tag{25}$$

which starts from the base distribution and induces the laws $(p_t^v)_t$, with the aim of matching the target p_1. While many solutions exist, we focus on the one that induces the same laws, $(p_t)_t$ as the following *Stochastic Interpolant*

$$x_t = \alpha_t x_1 + \beta_t x_0 + \gamma_t z, \tag{26}$$

where (x_0, x_1) are sampled from any coupling distribution π with marginals p_0, p_1, respectively, α, β, γ are non-negative functions satisfying certain conditions, and z follows a standard Gaussian distribution $\mathcal{N}(0, I)$, independently of (x_0, x_1). All conditional Gaussian score identities below are restricted to times with $\gamma_t > 0$ and assume the differentiability and integrability needed to interchange differentiation and integration.

Remark 7 *By choosing $p_0 = \delta_0$, i.e. $X_0 \equiv 0$, the above formulation recovers the standard setting in* Diffusion Models *(DM) and* Flow Matching *(FM).*

Remark 8 *While the velocity field generating a given marginal probability path p_t is not unique, the canonical marginal velocity field induced by the stochastic interpolant in Equation (26), $(v_t^*)_{t\in[0,1]}$, is uniquely defined via conditional expectation (Albergo et al., 2025):*

$$\begin{aligned} v_t^*(x_t) &= \mathbb{E}[\dot{\alpha}_t x_1 + \dot{\beta}_t x_0 + \dot{\gamma}_t z \mid x_t] \\ &= \mathbb{E}[\dot{\alpha}_t x_1 + \dot{\beta}_t x_0 \mid x_t] + \dot{\gamma}_t \mathbb{E}[z \mid x_t]. \end{aligned} \tag{27}$$

Defining $f_t(x_t) := \mathbb{E}[\dot{\alpha}_t x_1 + \dot{\beta}_t x_0 \mid x_t]$ and applying Tweedie's formula to the Gaussian noise term, we can express the marginal velocity as:

$$v_t^*(x_t) = f_t(x_t) - \dot{\gamma}_t \gamma_t \nabla \log p_t(x_t). \tag{28}$$

An imperfect velocity $v_t \neq v_t^*$ can change the marginal path, but need not do so: if $\nabla \cdot [p_t(v_t - v_t^*)] = 0$ at all times, the same marginals solve both continuity equations. For example, for radial p_t in dimension at least two, a field $\delta_t(x) = a(\|x\|^2)Jx$, with $J^\top = -J$ and smooth compactly supported a, is tangential to the density level sets and has zero divergence. It can be nonzero while preserving p_t. The general mismatch is quantified in Appendix A.8.

In the following content, we aim to estimate the likelihood induced by Equation (25), which is

$$\log p_t^v(x_t) = \log p_0(x_0) + \int_0^t \mathrm{d}_u \log p_u^v(x_u)\mathrm{d}u, \quad \text{where} \tag{29}$$

$$\mathrm{d}_u \log p_u^v(x_u) = -\nabla \cdot v_u(x_u) = -\mathrm{tr}\big(\nabla v_u(x_u)\big), \tag{30}$$

and $(x_u)_{u\in[0,t]}$ follow the dynamics in Equation (25). For clarity, we present the estimators and connections based on the following assumption:

Assumption 1 *We assume $v_t = v_t^*$, which implies $p_t^v = p_t^{v^*}$ and $\nabla \log p_t^v = \nabla \log p_t^{v^*}$.*

Under Assumption 1, we denote p_t as the marginals for clarity.

A.1 Desiderata for a Consistent Objective

At fixed t, let $Y = f_t(x_t|\xi)$ be a scalar or vector conditional target, $m_t(x_t) = \mathbb{E}[Y \mid x_t]$ its population conditional mean, and $D_\theta(x_t, t)$ a predictor. Assume the target and sampling law are fixed independently of θ and the squared losses are finite. Conditional expectation gives the loss decomposition (Tong et al., 2024; Yu et al., 2025)

$$\mathbb{E}\|D_\theta - Y\|^2 = \mathbb{E}\|D_\theta - m_t\|^2 + \mathbb{E}\|Y - m_t\|^2. \tag{31}$$

To see this, expand $D_\theta - Y = (D_\theta - m_t) + (m_t - Y)$; the cross term has expectation zero because $\mathbb{E}[m_t - Y \mid x_t] = 0$. The final term is independent of θ, so conditional and marginal regression have the same minimizers and, when differentiation under the expectation is justified, the same predictor gradients. The identity also holds after averaging over t with a fixed nonnegative weight $\lambda(t)$, provided the weighted losses are finite. Thus it suffices to construct a tractable conditional target whose mean is the desired marginal quantity.

A.2 Derivation through Total Derivative

For clarity, we define the following notation throughout the paper:

$$\tilde{v}_t(x_t|x_0, x_1) := \dot{\alpha}_t x_1 + \dot{\beta}_t x_0 + \dot{\gamma}_t z, \quad \text{where } z = \frac{x_t - (\alpha_t x_1 + \beta_t x_0)}{\gamma_t}, \tag{32}$$

and $v_t(x_t) = \mathbb{E}[\tilde{v}_t(x_t|x_0, x_1)|x_t]$.

To estimate $\mathrm{d}_t \log p_t(x_t)$, one could expand it through total derivative

$$\begin{aligned}
\mathrm{d}_t \log p_t(x_t) &= \partial_t \log p_t(x_t) + \nabla \log p_t(x_t) \cdot \mathrm{d}_t x_t && (33)\\
&= \partial_t \log p_t(x_t) + \nabla \log p_t(x_t) \cdot v_t(x_t), && (34)
\end{aligned}$$

where both $\nabla \log p_t(x_t)$ and $\partial_t \log p_t(x_t)$ can be expressed as conditional expectations (Song et al., 2021; Yu et al., 2025), respectively, as follows:

$$\begin{aligned}
\nabla \log p_t(x_t) &= \mathbb{E}\big[\nabla_{x_t} \log p(x_t|x_0, x_1)|x_t\big] && (35)\\
&= -\frac{1}{\gamma_t}\mathbb{E}\big[z|x_t\big], && (36)\\
\partial_t \log p_t(x_t) &= \mathbb{E}\big[\partial_t \log p(x_t|x_0, x_1)|x_t\big] && (37)\\
&= \frac{1}{\gamma_t}\mathbb{E}\Big[-d\dot{\gamma}_t + z \cdot \Big(\dot{\alpha}_t x_1 + \dot{\beta}_t x_0 + \dot{\gamma}_t z\Big)\,\Big|x_t\Big] && (38)\\
&= \frac{1}{\gamma_t}\mathbb{E}\Big[-d\dot{\gamma}_t + z \cdot \tilde{v}_t(x_t|x_0, x_1)\Big|x_t\Big]. && (39)
\end{aligned}$$

Therefore, the total derivative can be expressed as a conditional expectation

$$\mathrm{d}_t \log p_t(x_t) = -\mathbb{E}\Big[\frac{z}{\gamma_t} \cdot \Big(v_t(x_t) - \tilde{v}_t(x_t|x_0, x_1)\Big) + \frac{\dot{\gamma}_t}{\gamma_t} d\Big|x_t\Big], \tag{40}$$

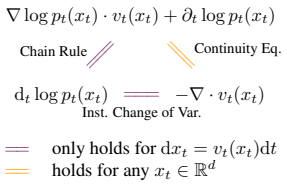
which can be learned efficiently without any high-order derivatives given access to v_t.

A.3 Derivation through the Continuity Equation

Alternatively, Equation (15) can be derived from the (log) continuity equation

$$\partial_t p_t(x_t) + \nabla \cdot (v_t(x_t) p_t(x_t)) = 0 \Leftrightarrow \partial_t \log p_t(x_t) = -\nabla \log p_t(x_t) \cdot v_t(x_t) - \nabla \cdot v_t(x_t). \quad (41)$$

By the instantaneous change-of-variable (Chen et al., 2019), $\mathrm{d}_t \log p_t(x_t) = -\nabla \cdot v_t(x_t)$, we recover Equation (15). Thus, the chain rule and continuity equation yield the instantaneous change-of-variables formula, providing an alternative proof to Chen et al. (2019). These relations are visualized in the diagram on the right-hand side.



A.4 Derivation through Manipulating the Trace Operator

On the other hand, we could then expand the Jacobian of the velocity, $J_v(x_t) = \nabla v_t(x_t)$, as follows

$$\begin{aligned}
\nabla v_t(x_t) &= \nabla_{x_t} \int p(x_0, x_1|x_t) \tilde{v}_t(x_t|x_0, x_1) dx_0 dx_1 \\
&\overset{(i)}{=} \int \Big(\tilde{v}_t(x_t|x_0, x_1) \nabla_{x_t} p(x_0, x_1|x_t)^\top + p(x_0, x_1|x_t) \nabla_{x_t} \tilde{v}_t(x_t|x_0, x_1) \Big) dx_0 dx_1 \\
&\overset{(ii)}{=} \int p(x_0, x_1|x_t) \Big(\tilde{v}_t(x_t|x_0, x_1) \nabla_{x_t} \log p(x_0, x_1|x_t)^\top + \nabla_{x_t} \tilde{v}_t(x_t|x_0, x_1) \Big) dx_0 dx_1 \\
&\overset{(iii)}{=} \int p(x_0, x_1|x_t) \Big(\tilde{v}_t(x_t|x_0, x_1) \nabla_{x_t} \log p(x_t|x_0, x_1)^\top \\
&\qquad - \tilde{v}_t(x_t|x_0, x_1) \nabla_{x_t} \log p_t(x_t)^\top + \nabla_{x_t} \tilde{v}_t(x_t|x_0, x_1) \Big) dx_0 dx_1, \quad (42)
\end{aligned}$$

where (i) is by $\nabla(pf) = f(\nabla p)^\top + p \nabla f$ for any vector-valued function f; (ii) is according to $\nabla p = p \nabla \log p$; and (iii) is based on $\nabla_{x_t} \log p(x_0, x_1|x_t) = \nabla_{x_t} \log p(x_t|x_0, x_1) - \nabla \log p_t(x_t)$. By plugging

$$\nabla_{x_t} \log p(x_t|x_0, x_1) = \frac{\alpha_t x_1 + \beta_t x_0 - x_t}{\gamma_t^2} = -\frac{z}{\gamma_t}, \quad (43)$$

$$\nabla_{x_t} \tilde{v}_t(x_t|x_0, x_1) = \nabla_{x_t} \Big(\dot{\alpha}_t x_1 + \dot{\beta}_t x_0 + \dot{\gamma}_t \frac{x_t - \alpha_t x_1 - \beta_t x_0}{\gamma_t} \Big) = \frac{\dot{\gamma}_t}{\gamma_t} I, \quad (44)$$

one can simply write the Jacobian as a conditional expectation:

$$\nabla v_t(x_t) = \mathbb{E}\big[\tilde{J}_v(x_t|x_0, x_1) | x_t \big], \quad (45)$$

where

$$\begin{aligned}
\tilde{J}_v(x_t|x_0, x_1) &= -\big(\dot{\alpha}_t x_1 + \dot{\beta}_t x_0 + \dot{\gamma}_t z \big) \Big(\nabla \log p_t(x_t) + \frac{z}{\gamma_t} \Big)^\top + \frac{\dot{\gamma}_t}{\gamma_t} I \quad (46) \\
&= -\tilde{v}_t(x_t|x_0, x_1) \Big(\nabla \log p_t(x_t) + \frac{z}{\gamma_t} \Big)^\top + \frac{\dot{\gamma}_t}{\gamma_t} I. \quad (47)
\end{aligned}$$

The divergence is then obtained by applying a trace operator:

$$\nabla \cdot v_t(x_t) = \mathrm{tr}\Big(\mathbb{E}\big[\tilde{J}_v(x_t|x_0,x_1)|x_t\big]\Big) = \mathbb{E}\Big[\mathrm{tr}\Big(\tilde{J}_v(x_t|x_0,x_1)\Big)|x_t\Big], \tag{48}$$

where

$$\mathrm{tr}\Big(\tilde{J}_v(x_t|x_0,x_1)\Big) = -\big(\dot{\alpha}_t x_1 + \dot{\beta}_t x_0 + \dot{\gamma}_t z\big)\cdot\Big(\nabla \log p_t(x_t) + \frac{z}{\gamma_t}\Big) + \frac{\dot{\gamma}_t}{\gamma_t} d \tag{49}$$

$$= -\tilde{v}_t(x_t|x_0,x_1)\cdot\Big(\nabla \log p_t(x_t) + \frac{z}{\gamma_t}\Big) + \frac{\dot{\gamma}_t}{\gamma_t} d. \tag{50}$$

Example 1. When $p_0 = \delta_0$, i.e. $X_0 \equiv 0$, the formulation reduces to a standard Diffusion model or Flow matching. By further choosing the optimal path, i.e. $x_t = tx_1 + (1-t)z$, and noting that

$$v_t(x_t) = \frac{x_t}{t} + \frac{1-t}{t}\nabla \log p_t(x_t). \tag{51}$$

One could simply write the above conditional Jacobian as

$$\tilde{J}_v(x_t|x_0,x_1) = -\big(x_1 - z\big)\Big(\frac{tv_t(x_t) - x_t}{1-t} + \frac{z}{1-t}\Big)^\top - \frac{1}{1-t} I \tag{52}$$

$$= -\big(x_1 - z\big)\Big(\frac{tv_t(x_t) - t(x_1 - z)}{1-t}\Big)^\top - \frac{1}{1-t} I \tag{53}$$

$$= -\frac{t}{1-t}\big(x_1 - z\big)\Big(v_t(x_t) - \big(x_1 - z\big)\Big)^\top - \frac{1}{1-t} I. \tag{54}$$

A.5 Connection between Two Estimators, and Beyond

We now demonstrate that the two estimators for $\mathrm{d}_t \log p_t(x_t)$ introduced in Appendices A.2 and A.4 are mathematically equivalent, differing only by a zero-mean term. Recall the two formulations:

$$\mathrm{d}_t \log p_t(x_t) = -\mathbb{E}\Big[\frac{z}{\gamma_t}\cdot\Big(v_t(x_t) - \tilde{v}_t(x_t|x_0,x_1)\Big) + \frac{\dot{\gamma}_t}{\gamma_t} d \Big| x_t\Big] \tag{55}$$

$$= -\mathbb{E}\Big[-\tilde{v}_t(x_t|x_0,x_1)\cdot\Big(\nabla \log p_t(x_t) + \frac{z}{\gamma_t}\Big) + \frac{\dot{\gamma}_t}{\gamma_t} d \Big| x_t\Big]. \tag{56}$$

Notice that for any vector-valued function $h_t(x_t)$, according to Equation (27), we have

$$\mathbb{E}\Big[h_t(x_t)\cdot\big(v_t(x_t) - \tilde{v}_t(x_t|x_0,x_1)\big)\Big|x_t\Big] = h_t(x_t)\cdot\Big(v_t(x_t) - \mathbb{E}\big[\tilde{v}_t(x_t|x_0,x_1)|x_t\big]\Big)$$
$$= h_t(x_t)\cdot\Big(v_t(x_t) - v_t(x_t)\Big) = 0. \tag{57}$$

Then Equation (55) can be written as

$$-\mathbb{E}\Big[\Big(\frac{z}{\gamma_t} + \nabla \log p_t(x_t)\Big)\cdot\Big(v_t(x_t) - \tilde{v}_t(x_t|x_0,x_1)\Big) + \frac{\dot{\gamma}_t}{\gamma_t} d \Big| x_t\Big] \tag{58}$$

$$= -\mathbb{E}\Big[-\tilde{v}_t(x_t|x_0,x_1)\cdot\Big(\frac{z}{\gamma_t} + \nabla \log p_t(x_t)\Big) + \frac{\dot{\gamma}_t}{\gamma_t} d \Big| x_t\Big]$$
$$- v_t(x_t)\cdot\Big(\underbrace{\frac{1}{\gamma_t}\mathbb{E}[z|x_t] + \nabla \log p_t(x_t)}_{\overset{(i)}{=} -\nabla \log p_t(x_t) + \nabla \log p_t(x_t) = 0}\Big), \tag{59}$$

where (i) is by Tweedie's formula.

A.6 Vectorized Estimators

We first conclude that the vectorized estimators obtained by replacing the inner-product $\cdot$ in Equations (55) and (56) with an elementwise product $\odot$ are unbiased estimators for $\mathrm{diag}(-\nabla v_t(x_t))$.

The conclusion can be simply obtained from the derivation of the conditional trace estimator, Equation (56), where

$$\mathrm{diag}\big(ab^\top\big) = a \odot b \quad \text{and} \quad \mathrm{tr}\big(ab^\top\big) = a \cdot b. \tag{60}$$

The conclusion also applies to the conditional total derivative estimator, Equation (55), since the difference between Equations (55) and (56) is a term with zero expectation as stated in their connection.

Transformation of the vectorized target. If the velocity is rotation-equivariant, its Jacobian satisfies $J_v(Qx) = QJ_v(x)Q^\top$ for an orthogonal block rotation Q. The trace is invariant, but the diagonal generally does not transform as an ordinary vector: $\mathrm{diag}(QJQ^\top) \neq Q\,\mathrm{diag}(J)$. For instance, $J = I$ has unchanged diagonal while $Q\mathbf{1}_d$ need not equal $\mathbf{1}_d$. Thus the likelihood head learns coordinatewise diagonal targets whose sum estimates an invariant scalar. Rotation augmentation encourages the desired learned symmetry, but does not by itself guarantee exact equivariance or invariance.

A.7 Connection with Higher-Order Score Matching

We connect the conditional estimators with the second-order Tweedie identity. For $x \sim p_{\mathrm{data}}$ and $q_\sigma(\tilde{x} \mid x) = \mathcal{N}(\tilde{x}; x, \sigma^2 I)$ with $\sigma > 0$, let $\tilde{p}$ be the marginal density of $\tilde{x}$ and write $\otimes^2 x = xx^\top$. We assume the differentiability and moment conditions needed for the conditional moments below.

According to Equation (28) and under Assumption 1, the Jacobian of velocity can be written as

$$\nabla v_t(x_t) = \nabla f_t(x_t) - \dot{\gamma}_t \gamma_t \nabla^2 \log p_t(x_t). \tag{61}$$

The second-order Tweedie identity (Meng et al., 2021) gives

$$\mathbb{E}[\otimes^2 x|\tilde{x}] = \sigma^2 I + \sigma^4 \nabla^2 \log \tilde{p}(\tilde{x}) + \otimes^2\big(\tilde{x} + \sigma^2 \nabla \log \tilde{p}(\tilde{x})\big) \tag{62}$$

$$\Longrightarrow \nabla^2 \log \tilde{p}(\tilde{x}) = \frac{1}{\sigma^4}\Big(\mathbb{E}[\otimes^2 x|\tilde{x}] - \sigma^2 I - \otimes^2\big(\tilde{x} + \sigma^2 \nabla \log \tilde{p}(\tilde{x})\big)\Big). \tag{63}$$

Let $x = \alpha_t x_1 + \beta_t x_0$. Then we have

$$\tilde{p}(\tilde{x}) = \int \pi(x_0, x_1)\delta(\alpha_t x_1 + \beta_t x_0 - x)\mathcal{N}(\tilde{x}; x, \gamma_t^2 I) dx dx_0 dx_1 = p_t(\tilde{x}). \tag{64}$$

Hence, by applying the above second-order Tweedie's formula, the Hessian of $\log p_t(x_t)$ can be written as

$$\nabla^2 \log p_t(x_t) = \frac{1}{\gamma_t^4}\Big(\mathbb{E}[\otimes^2\big(\alpha_t x_1 + \beta_t x_0\big)|x_t] - \gamma_t^2 I - \otimes^2\big(x_t + \gamma_t^2 \nabla \log p_t(x_t)\big)\Big) \tag{65}$$

For clarity, we focus on the example of Diffusion models, i.e. $p_0 = \delta_0$. In such a case, the velocity and score can be related as

$$v_t(x_t) = \frac{\dot{\alpha}_t}{\alpha_t} x_t + \left(\frac{\dot{\alpha}_t}{\alpha_t} - \frac{\dot{\gamma}_t}{\gamma_t}\right)\gamma_t^2 \nabla \log p_t(x_t), \tag{66}$$

and therefore

$$\nabla v_t(x_t) = \frac{\dot{\alpha}_t}{\alpha_t} I + \Big(\frac{\dot{\alpha}_t}{\alpha_t} - \frac{\dot{\gamma}_t}{\gamma_t}\Big)\gamma_t^2 \nabla^2 \log p_t(x_t). \tag{67}$$

For clarity, we denote $c_t = \dot{\alpha}_t/\alpha_t - \dot{\gamma}_t/\gamma_t$. The Hessian can be simplified as follows:

$$\nabla^2 \log p_t(x_t) = \frac{1}{\gamma_t^4}\Big(\mathbb{E}[\alpha_t^2 x_1 x_1^\top | x_t] - \gamma_t^2 I - \otimes^2\big(x_t + \gamma_t^2 \nabla \log p_t(x_t)\big)\Big). \tag{68}$$

Since $\mathrm{d}_t \log p_t(x_t)$ is the negative trace of the Jacobian of v_t, we focus on the diagonal entry of the Hessian:

$$\begin{aligned}\operatorname{diag}\big(\nabla^2 \log p_t(x_t)\big) = \frac{1}{\gamma_t^4}\Big(&\mathbb{E}[\alpha_t^2 \odot^2 x_1 | x_t] - \gamma_t^2 \mathbf{1}_d - \odot^2 x_t \\ &- 2\gamma_t^2 x_t \odot \nabla \log p_t(x_t) - \gamma_t^4 \odot^2 \nabla \log p_t(x_t)\Big),\end{aligned} \tag{69}$$

where we denote $\odot^2$ as 2-fold elementwise product for clarity. And notice that

$$\begin{aligned}\mathbb{E}[\alpha_t^2 \odot^2 x_1 | x_t] &= \mathbb{E}[\odot^2 (x_t - \gamma_t z) | x_t] && (70)\\ &= \odot^2 x_t - 2\gamma_t x_t \odot \mathbb{E}[z|x_t] + \gamma_t^2 \mathbb{E}[\odot^2 z | x_t] && (71)\\ &\overset{(i)}{=} \odot^2 x_t + 2\gamma_t^2 x_t \odot \nabla \log p_t(x_t) + \gamma_t^2 \mathbb{E}[\odot^2 z | x_t], && (72)\end{aligned}$$

where (i) is by Tweedie's formula. The diagonal of the Hessian can be simplified as

$$\operatorname{diag}\big(\nabla^2 \log p_t(x_t)\big) = \mathbb{E}\Big[\odot^2\Big(\frac{1}{\gamma_t} z\Big)\Big| x_t\Big] - \odot^2 \nabla \log p_t(x_t) - \frac{1}{\gamma_t^2}\mathbf{1}_d. \tag{73}$$

Further notice that

$$\begin{aligned}\mathbb{E}\Big[\odot^2\Big(\frac{1}{\gamma_t} z\Big)\Big| x_t\Big] &= \mathbb{E}\Big[\odot^2\Big(\frac{1}{\gamma_t} z + \nabla \log p_t(x_t)\Big)\Big| x_t\Big] \\ &\quad - 2\mathbb{E}\Big[\frac{1}{\gamma_t} z \Big| x_t\Big] \odot \nabla \log p_t(x_t) - \odot^2 \nabla \log p_t(x_t) && (74)\\ &\overset{(i)}{=} \mathbb{E}\Big[\odot^2\Big(\frac{1}{\gamma_t} z + \nabla \log p_t(x_t)\Big)\Big| x_t\Big] + \odot^2 \nabla \log p_t(x_t), && (75)\end{aligned}$$

where (i) is again by Tweedie's formula. We have the simplification

$$\begin{aligned}\operatorname{diag}\big(\nabla^2 \log p_t(x_t)\big) &= \mathbb{E}\Big[\odot^2\Big(\frac{1}{\gamma_t} z + \nabla \log p_t(x_t)\Big)\Big| x_t\Big] - \frac{1}{\gamma_t^2}\mathbf{1}_d && (76)\\ &= \mathbb{E}\Big[\frac{z}{\gamma_t} \odot \Big(\frac{z}{\gamma_t} + \nabla \log p_t(x_t)\Big)\Big| x_t\Big] - \frac{1}{\gamma_t^2}\mathbf{1}_d. && (77)\end{aligned}$$

Therefore,

$$\operatorname{diag}\big(\nabla v_t(x_t)\big) = \frac{\dot\alpha_t}{\alpha_t}\mathbf{1}_d + c_t\gamma_t^2\mathbb{E}\Big[\frac{z}{\gamma_t}\odot\Big(\frac{z}{\gamma_t}+\nabla\log p_t(x_t)\Big)\Big|x_t\Big] - c_t\mathbf{1}_d \tag{78}$$

$$\overset{(i)}{=} \mathbb{E}\Big[\frac{z}{\gamma_t}\odot\Big(c_t\gamma_t z + v_t(x_t) - \frac{\dot\alpha_t}{\alpha_t}x_t\Big)\Big|x_t\Big] + \frac{\dot\gamma_t}{\gamma_t}\mathbf{1}_d \tag{79}$$

$$\overset{(ii)}{=} \frac{1}{\gamma_t}\mathbb{E}\Big[z\odot\Big(v_t(x_t) - \tilde v_t(x_t|x_0,x_1)\Big)\Big|x_t\Big] + \frac{\dot\gamma_t}{\gamma_t}\mathbf{1}_d, \tag{80}$$

where (i) follows from the relationship between velocity and score stated in Equation (66) and (ii) is by

$$\frac{\dot\alpha_t}{\alpha_t}x_t - c_t\gamma_t z = \frac{\dot\alpha_t}{\alpha_t}(\alpha_t x_1 + \cancel{\gamma_t z}) - \Big(\cancel{\frac{\dot\alpha_t}{\alpha_t}} - \frac{\dot\gamma_t}{\gamma_t}\Big)\gamma_t z = \dot\alpha_t x_1 + \dot\gamma_t z = \tilde v_t(x_t|x_0,x_1). \tag{81}$$

Remarkably, it recovers the estimator derived from the conditional total derivative, Equation (40), in the diffusion setting.

Example. When choosing the optimal path, i.e. $\alpha_t = t$ and $\gamma_t = 1-t$, the diagonal of Jacobian results in

$$\operatorname{diag}\big(\nabla v_t(x_t)\big) = \frac{1}{1-t}\mathbb{E}\Big[z\odot\Big(v_t(x_t) - (x_1 - z)\Big)\Big|x_t\Big] - \frac{1}{1-t}\mathbf{1}_d, \tag{82}$$

A.8 Error Quantification

A.8.1 Notation and Assumptions

We work on $\mathbb{R}^d$. Let v_t^* denote the ground-truth velocity field generating the probability path $(p_t)_{t\in[0,1]}$ through the continuity equation $\partial_t p_t + \nabla\cdot(p_t v_t^*) = 0$, and let v_t denote the velocity learned by the model, with marginal p_t^v satisfying $\partial_t p_t^v + \nabla\cdot(p_t^v v_t) = 0$. We assume throughout that the densities are strictly positive and that the densities and velocity fields are sufficiently regular for the continuity equations and the characteristic identities below to hold pointwise. In particular, the learned ODE admits a global diffeomorphic flow ϕ_t such that $\phi_{t\#}p_0 = p_t^v$. Define the velocity error

$$\delta_t(x) := v_t(x) - v_t^*(x). \tag{83}$$

We take $p_0 = \mathcal{N}(0, I)$ and adopt the parameterization $x_t = \alpha_t x_1 + \gamma_t z$, where $x_1 \sim p_1$ and $z \sim \mathcal{N}(0, I)$ are independent. The schedules satisfy $\alpha_0 = 0$, $\gamma_0 = 1$, $\alpha_1 = 1$, $\gamma_1 = 0$, $\alpha_t > 0$ for $t \in (0, 1]$, and $\gamma_t > 0$ for $t \in [0, 1)$. We define the following quantities:

$$e_t^2 := \mathbb{E}_{p_t}\big[\|\delta_t(x_t)\|^2\big], \qquad B_t := \|\delta_t\|_{L^\infty(\mathbb{R}^d)}, \qquad S_t^2 := \mathbb{E}_{p_t}\big[\|\mathcal{D}_t\delta_t(x_t)\|^2\big], \tag{84}$$

where $\|\cdot\|$ is the Euclidean norm and L^∞ denotes the essential supremum (with the Euclidean norm for vector-valued functions), and

$$\mathcal{D}_t f := \Delta f + \nabla\log p_t\cdot\nabla f \;=\; p_t^{-1}\nabla\cdot\big(p_t\,\nabla f\big) \tag{85}$$

is the generator of the overdamped Langevin diffusion with invariant measure p_t, applied componentwise to vector fields. We make the following assumptions:

Assumption 2 *$B_t < \infty$ for $t \in [0, 1]$.*

Assumption 3 *For every $t \in [0,1]$, the components of δ_t lie in the domain of $\mathcal{D}_t$, $S_t < \infty$, and the weighted integration-by-parts identity*

$$\mathbb{E}_{p_t}\big[\|\nabla \delta_t(x_t)\|_F^2\big] = -\mathbb{E}_{p_t}\big[\delta_t(x_t) \cdot \mathcal{D}_t \delta_t(x_t)\big] \tag{86}$$

holds. A sufficient condition is that δ_t is twice continuously differentiable, all displayed integrals are absolutely convergent, and the boundary term in the divergence theorem vanishes.

Assumption 4 *The density p_1 is continuously differentiable, the integration by parts used in the target-score identity below has no boundary term, and its fourth-order score moment is finite:*

$$J_4(p_1) := \mathbb{E}_{p_1}[\|\nabla \log p_1(x_1)\|^4] < \infty. \tag{87}$$

We further define the following score factor for notational convenience:

$$G_t := \min\Big(\frac{\sqrt{d(d+2)}}{\gamma_t^2}, \frac{\sqrt{J_4(p_1)}}{\alpha_t^2} \Big), \tag{88}$$

with the extended-value convention that any term with zero denominator is interpreted as $+\infty$. In particular, $G_t < \infty$, $G_0 = \sqrt{d(d+2)}$, and $G_1 = \sqrt{J_4(p_1)}$.

Outline of the proof. We first bound the local weighted-divergence error and propagate it along the learned flow, including the residual of a fitted likelihood head. The ideal integrated surrogate equals the interpolation density exactly, so its population importance-sampling estimand reduces to an expectation under the teacher's sampling law. We then bound the teacher's observable bias and the population importance-sampling error of a fitted head, and state their dependence on assumed learning rates.

A.8.2 Pointwise Error

Along trajectories of the learned ODE $\mathrm{d}x_t = v_t(x_t)\mathrm{d}t$, the instantaneous change-of-variables formula gives

$$\mathrm{d}_t \log p_t^v(x_t) = \partial_t \log p_t^v(x_t) + \nabla \log p_t^v(x_t) \cdot v_t(x_t) = -\nabla \cdot v_t(x_t). \tag{89}$$

Because both $\nabla \log p_t^v$ and $\partial_t \log p_t^v$ are intractable, we replace them by $\nabla \log p_t$ and $\partial_t \log p_t$ respectively to obtain the estimator

$$\widetilde{\mathrm{d}_t \log p_t^v}(x_t) := \partial_t \log p_t(x_t) + \nabla \log p_t(x_t) \cdot v_t(x_t), \tag{90}$$

where both $\nabla \log p_t$ and $\partial_t \log p_t$ could be estimated by a posterior expectation over the dataset (Song et al., 2021; Yu et al., 2025).

We further introduce the material derivative of the data density in the direction of the ground-truth velocity field. The superscript v^* below specifies this direction and does not denote a density different from p_t:

$$\begin{aligned} \mathrm{d}_t \log p_t^{v^*}(x) &:= \partial_t \log p_t(x) + \nabla \log p_t(x) \cdot v_t^*(x) && (91) \\ &= -\nabla \cdot v_t^*(x), && (92) \end{aligned}$$

where the last equality follows from the continuity equation $\partial_t \log p_t + \nabla \log p_t \cdot v_t^* = -\nabla \cdot v_t^*$.

Define the pointwise error of the estimator as

$$\mathcal{E}_t(x_t) := \widetilde{\mathrm{d}_t \log p_t^v}(x_t) - \mathrm{d}_t \log p_t^v(x_t). \tag{93}$$

Adding and subtracting $\mathrm{d}_t \log p_t^{v^*}(x_t)$, we can decompose the error as

$$\mathcal{E}_t(x_t) = \underbrace{\widetilde{\mathrm{d}_t \log p_t^v}(x_t) - \mathrm{d}_t \log p_t^{v^*}(x_t)}_{\text{score–velocity mismatch}} + \underbrace{\mathrm{d}_t \log p_t^{v^*}(x_t) - \mathrm{d}_t \log p_t^v(x_t)}_{\text{model-flow mismatch}}. \tag{94}$$

The first term is given by

$$\begin{aligned}
\widetilde{\mathrm{d}_t \log p_t^v}(x_t) - \mathrm{d}_t \log p_t^{v^*}(x_t) &= \partial_t \log p_t(x_t) + \nabla \log p_t(x_t) \cdot v_t(x_t) \\
&\quad - \partial_t \log p_t(x_t) - \nabla \log p_t(x_t) \cdot v_t^*(x_t) \qquad (95) \\
&= \nabla \log p_t(x_t) \cdot \delta_t(x_t). \qquad (96)
\end{aligned}$$

The second term is given by

$$\begin{aligned}
\mathrm{d}_t \log p_t^{v^*}(x_t) - \mathrm{d}_t \log p_t^v(x_t) &= -\nabla \cdot v_t^*(x_t) - \big(-\nabla \cdot v_t(x_t)\big) \qquad (97) \\
&= \nabla \cdot \delta_t(x_t). \qquad (98)
\end{aligned}$$

Therefore,

$$\mathcal{E}_t(x_t) = \nabla \log p_t(x_t) \cdot \delta_t(x_t) + \nabla \cdot \delta_t(x_t). \tag{99}$$

Equivalently, the error is the weighted divergence

$$\mathcal{E}_t = p_t^{-1} \nabla \cdot (p_t \delta_t). \tag{100}$$

A nonzero velocity error can therefore have zero scalar CDM error. If $\nabla \cdot (p_t \delta_t) = 0$ throughout the path, then p_t also solves the learned continuity equation; uniqueness with the common initial condition gives $p_t^v = p_t$.

Lemma 1 (Mean-square error of the estimator) *Under Assumptions 2 to 4 and the flow and interpolation regularity stated above, define*

$$\eta_t := \|\mathcal{E}_t\|_{L^2(p_t)}, \qquad K_t := \sqrt{e_t}\big(\sqrt{S_t} + \sqrt{B_t G_t}\big). \tag{101}$$

Then, for $t \in [0, 1]$,

$$\mathbb{E}_{p_t}[\mathcal{E}_t^2] = \eta_t^2 \le K_t^2 \le 2e_t(S_t + B_t G_t). \tag{102}$$

Proof. We first establish the score and velocity moment estimates, then control the full weighted divergence without bounding its two summands separately. For $t < 1$, differentiation of the Gaussian convolution gives

$$\nabla \log p_t(x_t) = -\gamma_t^{-1} \mathbb{E}[z \mid x_t], \qquad \mathbb{E}_{p_t} \|\nabla \log p_t\|^4 \le \frac{\mathbb{E}\|z\|^4}{\gamma_t^4} = \frac{d(d+2)}{\gamma_t^4}. \tag{103}$$

For $0 < t < 1$, integration by parts against p_1 instead gives the target score identity (Bortoli et al., 2024),

$$\nabla \log p_t(x_t) = \alpha_t^{-1} \mathbb{E}[\nabla \log p_1(x_1) \mid x_t], \qquad \mathbb{E}_{p_t} \|\nabla \log p_t\|^4 \le \frac{J_4(p_1)}{\alpha_t^4}. \tag{104}$$

The latter bound holds directly at $t = 1$. Conditional Jensen's inequality justifies both moment bounds. Thus, with the endpoint convention in Equation (88),

$$\big(\mathbb{E}_{p_t} \|\nabla \log p_t\|^4\big)^{1/2} \le G_t, \qquad \big(\mathbb{E}_{p_t} \|\delta_t\|^4\big)^{1/2} \le B_t e_t. \tag{105}$$

Suppress t and write $g = \nabla \log p$. The weighted integration-by-parts identity and Cauchy–Schwarz imply

$$A^2 := \mathbb{E}_p \|\nabla \delta\|_F^2 = -\mathbb{E}_p[\delta \cdot \mathcal{D}\delta] \le eS, \tag{106}$$

$$C^2 := \mathbb{E}_p[\|g\|^2 \|\delta\|^2] \le \big(\mathbb{E}_p \|g\|^4\big)^{1/2} \big(\mathbb{E}_p \|\delta\|^4\big)^{1/2} \le GBe. \tag{107}$$

Set $F = \sqrt{p}\,\delta$ and let $\|\cdot\|_2$ denote the unweighted $L^2(\mathrm{d}x)$ norm. Then $F \in H^1(\mathbb{R}^d; \mathbb{R}^d)$: $\|F\|_2 = e$ and the two terms in its weak derivative are square-integrable by the preceding estimates. The product rule yields

$$\sqrt{p}\,\mathcal{E} = \nabla \cdot F + \tfrac{1}{2}\sqrt{p}\, g \cdot \delta, \tag{108}$$

$$\nabla F = \sqrt{p}\,\nabla \delta + \tfrac{1}{2}\sqrt{p}\,\delta g^\top. \tag{109}$$

For an H^1 vector field, $\|\nabla \cdot F\|_2 \le \|\nabla F\|_{2,F}$, where the second norm uses the Frobenius norm. Indeed, Plancherel's identity and $|\xi \cdot \widehat{F}(\xi)|^2 \le \|\xi\|^2 \|\widehat{F}(\xi)\|^2$ prove this integrated inequality. Applying the triangle inequality twice gives

$$\eta = \|\sqrt{p}\,\mathcal{E}\|_2 \le \|\nabla F\|_{2,F} + \tfrac{1}{2}C \le A + C \le \sqrt{eS} + \sqrt{eBG} = K. \tag{110}$$

Squaring and using $(a+b)^2 \le 2a^2 + 2b^2$ completes the proof. □

A.8.3 Error of Likelihood Estimation

Recall the estimator $\widetilde{\mathrm{d}_t \log p_t^v}$ from Equation (90). We turn it into a surrogate log-density by integrating it along the learned trajectory ϕ (where $x_t = \phi_t(x_0)$, $x_0 \sim p_0$), matching the exact log-density at the same initial condition:

$$\log \tilde{p}_t^v\big(\phi_t(x_0)\big) := \log p_0(x_0) + \int_0^t \widetilde{\mathrm{d}_s \log p_s^v}\big(\phi_s(x_0)\big)\, ds, \tag{111}$$

$$\log p_t^v\big(\phi_t(x_0)\big) = \log p_0(x_0) + \int_0^t \mathrm{d}_s \log p_s^v\big(\phi_s(x_0)\big)\, ds. \tag{112}$$

The first integrand is exactly the derivative of the interpolation log-density along the learned trajectory:

$$\widetilde{\mathrm{d}_s \log p_s^v}(\phi_s(x_0)) = \frac{\mathrm{d}}{\mathrm{d}s} \log p_s(\phi_s(x_0)). \tag{113}$$

Since $p_0^v = p_0$ and ϕ_t is a global diffeomorphism, the chain rule and the common initial condition give the identities

$$\tilde{p}_t^v = p_t, \qquad \Phi_t := \log \tilde{p}_t^v - \log p_t^v = \log \frac{p_t}{p_t^v}. \tag{114}$$

Thus ideal population CDM integrated along an approximate teacher tracks the interpolation density, which need not equal the teacher's sampling density. Subtracting the two trajectory integrals gives

$$\Phi_t(\phi_t(x_0)) = \int_0^t \mathcal{E}_s(\phi_s(x_0))\, \mathrm{d}s. \tag{115}$$

All statements are conditional on the learned functions. Write $m_t := \widetilde{\mathrm{d}_t \log p_t^v}$ and let D_t^θ be a fitted scalar instantaneous head. Define its residual, training error, and total local error by

$$r_{D,t}(x) := D_t^\theta(x) - m_t(x), \qquad \varepsilon_{D,t}^2 := \mathbb{E}_{p_t^v} |r_{D,t}|^2, \tag{116}$$

$$\mathcal{E}_t^\theta := D_t^\theta - \mathrm{d}_t \log p_t^v = r_{D,t} + \mathcal{E}_t. \tag{117}$$

Its integrated log-density estimate and error are

$$\log \tilde{p}_t^{v,\theta}(\phi_t(x_0)) := \log p_0(x_0) + \int_0^t D_s^\theta(\phi_s(x_0))\,\mathrm{d}s, \tag{118}$$

$$\Phi_t^\theta := \log \tilde{p}_t^{v,\theta} - \log p_t^v. \tag{119}$$

The positive function $\tilde{p}_t^{v,\theta}$ need not integrate to one.

Proposition 9 (Likelihood error along the learned flow) *Under the assumptions of Lemma 1, let $M_s^* := \|p_s^v/p_s\|_{L^\infty(p_s)}$ be finite for almost every $s \in [0,t]$. For a fitted head with finite integrals below,*

$$\|\Phi_t^\theta\|_{L^2(p_t^v)} \le \int_0^t \left(\varepsilon_{D,s} + \sqrt{M_s^*}\,\eta_s\right)\mathrm{d}s$$

$$\le A_t^\theta := \int_0^t \left(\varepsilon_{D,s} + \sqrt{M_s^*}\,K_s\right)\mathrm{d}s, \tag{120}$$

$$\mathbb{E}_{p_t^v}|\Phi_t^\theta|^2 \le t\int_0^t \left[2\varepsilon_{D,s}^2 + 4M_s^* e_s(S_s + B_s G_s)\right]\,\mathrm{d}s. \tag{121}$$

For the ideal head $D_s^\theta = m_s$, the residual vanishes and $\tilde{p}_t^{v,\theta} = \tilde{p}_t^v = p_t$. In this case,

$$\|\Phi_t\|_{L^2(p_t^v)} \le \int_0^t \sqrt{M_s^*}\,\eta_s\,\mathrm{d}s \le \int_0^t \sqrt{M_s^*}\,K_s\,\mathrm{d}s, \tag{122}$$

$$\mathbb{E}_{p_t^v}[\Phi_t^2] \le t\int_0^t M_s^* K_s^2\,\mathrm{d}s \le 2t\int_0^t M_s^* e_s(S_s + B_s G_s)\,\mathrm{d}s. \tag{123}$$

Proof. The exact decomposition along a learned trajectory is

$$\Phi_t^\theta(\phi_t(x_0)) = \log\frac{p_t}{p_t^v}(\phi_t(x_0)) + \int_0^t r_{D,s}(\phi_s(x_0))\,\mathrm{d}s \tag{124}$$

$$= \int_0^t \mathcal{E}_s^\theta(\phi_s(x_0))\,\mathrm{d}s. \tag{125}$$

The pushforward relation $\phi_{s\#}p_0 = p_s^v$ and Minkowski's integral inequality give

$$\|\Phi_t^\theta\|_{L^2(p_t^v)} \le \int_0^t \|\mathcal{E}_s^\theta\|_{L^2(p_s^v)}\,\mathrm{d}s. \tag{126}$$

By the triangle inequality and change of measure,

$$\|\mathcal{E}_s^\theta\|_{L^2(p_s^v)} \le \varepsilon_{D,s} + \sqrt{M_s^*}\,\eta_s \le \varepsilon_{D,s} + \sqrt{M_s^*}\,K_s, \tag{127}$$

$$\mathbb{E}_{p_s^v}|\mathcal{E}_s^\theta|^2 \le 2\varepsilon_{D,s}^2 + 4M_s^* e_s(S_s + B_s G_s). \tag{128}$$

Cauchy–Schwarz in time proves the mean-square bound. Setting $r_{D,s} = 0$ gives the ideal-head norm bound; applying Cauchy–Schwarz directly to that bound gives the stated ideal-head mean-square estimate. □

A.8.4 Observable Error of the Teacher Flow

For $h : \mathbb{R}^d \to \mathbb{R}^k$, the population importance-sampling estimand is

$$H_t := \mathbb{E}_{p_t} h, \qquad \tilde{H}_t := \mathbb{E}_{p_t^v}\left[\frac{p_t}{\tilde{p}_t^v} h\right] = \mathbb{E}_{p_t^v} h. \tag{129}$$

The weight is identically one by Equation (114). Accordingly, the following result bounds observable bias of the approximate teacher using a square-root-density argument.

Theorem 10 (Observable error of the teacher flow) *Under the assumptions of Lemma 1, suppose* $\int_0^t K_s \, \mathrm{d}s < \infty$. *Then*

$$\|\sqrt{p_t} - \sqrt{p_t^v}\|_{L^2(\mathrm{d}x)} \le \tfrac{1}{2} \int_0^t \eta_s \, \mathrm{d}s \le \tfrac{1}{2} \int_0^t K_s \, \mathrm{d}s. \tag{130}$$

If $\sigma_{h,t}^2 := \mathbb{E}_{p_t}\|h - H_t\|^2 < \infty$ *and the terminal ratio* $M_t^* < \infty$, *then*

$$\|H_t - \tilde{H}_t\| \le \tfrac{1}{2}\sigma_{h,t}(1 + \sqrt{M_t^*}) \int_0^t K_s \, \mathrm{d}s. \tag{131}$$

Alternatively, for bounded h, *no density-ratio bound is needed:*

$$\|H_t - \tilde{H}_t\| \le \|h\|_\infty \int_0^t K_s \, \mathrm{d}s. \tag{132}$$

All three conclusions apply at $t = 1$ *under the same endpoint regularity.*

Proof. Let $J_s(x_0) = |\det \nabla \phi_s(x_0)|$ and set $u_s(x_0) = \sqrt{p_s(\phi_s(x_0))J_s(x_0)}$. The continuity equation gives

$$\partial_s u_s = \tfrac{1}{2}\mathcal{E}_s(\phi_s(x_0))u_s, \qquad \sqrt{p_s^v(\phi_s(x_0))J_s(x_0)} = \sqrt{p_0(x_0)}. \tag{133}$$

By change of variables, $\|\partial_s u_s\|_{L^2(\mathrm{d}x_0)} = \eta_s/2$. Integrating from $u_0 = \sqrt{p_0}$ and applying Minkowski proves Equation (130). Since both densities integrate to one, subtracting H_t from h leaves the difference of expectations unchanged. Factor $p_t - p_t^v = (\sqrt{p_t} - \sqrt{p_t^v})(\sqrt{p_t} + \sqrt{p_t^v})$ and apply Cauchy–Schwarz. The inequality $(\sqrt{p_t} + \sqrt{p_t^v})^2 \le (1 + \sqrt{M_t^*})^2 p_t$ proves Equation (131). For bounded h, use instead $\|\sqrt{p_t} + \sqrt{p_t^v}\|_2 \le 2$ to obtain Equation (132). □

Density-ratio assumptions are substantive: even a nonzero, arbitrarily small translation of a Gaussian makes both supremum ratios infinite. The bounded-observable result above still applies.

A.8.5 Error of a Fitted Likelihood Head

For the vectorized implementation, the scalar instantaneous head is $D_t^\theta(x) = \mathbf{1}_d^\top D_\theta^{\mathrm{vec}}(x, t, t)$.

Training typically samples p_t, whereas $\varepsilon_{D,t}$ is measured under p_t^v. To connect them, let Y_t^{vec} be the vector CDM target with the same fixed teacher, and let $m_t^{\mathrm{vec}}(x) = \mathbb{E}[Y_t^{\mathrm{vec}} \mid x_t = x]$. Its components sum to m_t. Whenever the squared losses are finite,

$$R_t := \mathbb{E}_{p_t}\|D_\theta^{\mathrm{vec}}(x_t, t, t) - m_t^{\mathrm{vec}}(x_t)\|^2, \tag{134}$$

$$\varepsilon_{D,t}^2 \le d M_t^* R_t. \tag{135}$$

Here R_t is the excess vector CDM risk over its unrestricted population optimum, by the conditional-variance identity. The inequality follows from $|\mathbf{1}_d^\top a|^2 \le d\|a\|^2$ and density transfer. The factor d comes from summing the componentwise head errors. An empirical training loss alone does not bound R_t without a generalization argument.

Corollary 11 (Population importance-sampling error with a fitted head) *For* $\mathbb{E}_{p_t}\|h\|^2 < \infty$, *define*

$$\tilde{H}_t^\theta := \mathbb{E}_{p_t^v}\left[\frac{p_t}{\tilde{p}_t^{v,\theta}} h\right], \tag{136}$$

$$M_t := \|p_t/p_t^v\|_{L^\infty(p_t^v)}, \qquad \kappa_t^\theta := \|e^{\max(-\Phi_t^\theta, 0)}\|_{L^\infty(p_t)}. \tag{137}$$

Under the assumptions of Proposition 9, with $M_t, \kappa_t^\theta, A_t^\theta < \infty$,

$$\|H_t - \tilde{H}_t^\theta\| \le \sqrt{\mathbb{E}_{p_t}\|h\|^2}\, \kappa_t^\theta \sqrt{M_t}\, A_t^\theta. \tag{138}$$

This population bound also applies at $t = 1$; *it does not include finite-particle Monte Carlo error.*

Proof. Changing measure gives $H_t - \tilde{H}_t^\theta = \mathbb{E}_{p_t}[h(1 - e^{-\Phi_t^\theta})]$. Using $|1 - e^{-u}| \le |u| e^{\max(-u,0)}$, Cauchy–Schwarz, and density transfer yields

$$\|H_t - \tilde{H}_t^\theta\| \le \sqrt{\mathbb{E}_{p_t}\|h\|^2}\, \kappa_t^\theta \sqrt{M_t}\, \|\Phi_t^\theta\|_{L^2(p_t^v)}. \tag{139}$$

Apply Proposition 9. □

We next propagate assumed population learning rates; we do not derive these rates from the empirical training objective.

Assumption 5 (Finite-sample learning rates) *Denote the number of samples used to train the velocity field v_t as n_v, and denote the number of samples used to train the total-score estimator D_t^θ as n_D. Assume that uniformly over $s \in [0,t]$,*

$$\varepsilon_{D,s}(n_D) \le C_D n_D^{-\beta_D}, \tag{140}$$
$$e_{s,n_v} \le C_v n_v^{-\beta_v}, \tag{141}$$
$$B_{s,n_v} \le C_B, \tag{142}$$
$$S_{s,n_v} \le C_S, \tag{143}$$

where $C_D, C_v, C_B, C_S > 0, \beta_D, \beta_v > 0$ are constants. Further assume that G_s and M_s^* are uniformly bounded on $[0,t]$, i.e.

$$G_s \le C_G, \qquad M_s^* \le C_M, \qquad s \in [0,t]. \tag{144}$$

For the importance-sampling conclusion, also assume, uniformly in n_D and n_v,

$$\mathbb{E}_{p_t}\|h(x_t)\|^2 \le C_h^2, \qquad M_t \le C_{\text{ratio}}, \qquad \kappa_t^\theta \le C_\kappa. \tag{145}$$

All constants are independent of n_D and n_v. If the learning-rate bounds are probabilistic over the training samples, all conclusions below hold on the same event on which these uniform bounds hold.

Corollary 12 (Sample-size dependence) *Under the assumptions of Proposition 9 and Corollary 11 and the uniform learning-rate bounds in Assumption 5, the local deployed-estimator error satisfies*

$$\mathbb{E}_{p_s^v}\left[\left(\mathcal{E}_s^\theta(x_s)\right)^2\right] = O(n_D^{-2\beta_D}) + O(n_v^{-\beta_v}), \tag{146}$$

uniformly over $s \in [0,t]$. Consequently, the likelihood-estimation error satisfies

$$\mathbb{E}_{p_t^v}\left[\left(\Phi_t^\theta(x_t)\right)^2\right] = O(t^2 n_D^{-2\beta_D}) + O(t^2 n_v^{-\beta_v}), \tag{147}$$

and the resulting importance-sampling error satisfies

$$\|H_t - \tilde{H}_t^\theta\| = O(t n_D^{-\beta_D}) + O(t n_v^{-\beta_v/2}). \tag{148}$$

In particular, if both stages achieve root-sample rates, $\beta_D = \beta_v = 1/2$, then

$$\|H_t - \tilde{H}_t^\theta\| = O(t n_D^{-1/2}) + O(t n_v^{-1/4}). \tag{149}$$

At $t = 1$, this becomes

$$\|H_1 - \tilde{H}_1^\theta\| = O(n_D^{-1/2}) + O(n_v^{-1/4}). \tag{150}$$

Proof. Substituting Assumption 5 into Equation (128) gives, uniformly over $s \in [0,t]$,

$$\mathbb{E}_{p_s^v}|\mathcal{E}_s^\theta|^2 \le 2C_D^2 n_D^{-2\beta_D} + 4C_M C_v (C_S + C_B C_G) n_v^{-\beta_v}. \tag{151}$$

Integrating this bound in Proposition 9 gives the stated mean-square likelihood rate. Equivalently, $A_t^\theta = O(t n_D^{-\beta_D}) + O(t n_v^{-\beta_v/2})$; substituting this into Corollary 11 gives the importance-sampling rate. Set $\beta_D = \beta_v = 1/2$ for the final statements. □

B Expanded Discussion of Related Works

The following are three dominant approaches used in computational chemistry for training neural networks that support efficient sampling *and* likelihood evaluation. Once such neural networks are trained, they can be used as a *proposal* distribution, known in computational chemistry as a Boltzmann generator, for self-normalized importance sampling of a Boltzmann distribution.

Discrete and Continuous Normalizing Flows Boltzmann Generators (BGs) are tractable proposal distributions used in importance sampling, allowing both efficient sampling and likelihood evaluation. They were first applied to molecular systems by Noé et al. (2019), who parameterized them using neural networks in the form of Normalizing Flows (NFs). Subsequent work has evolved along two main directions. The first replaces NFs with their continuous counterpart, Continuous Normalizing Flows (CNFs Chen et al., 2019) that can be parameterized using more flexible model architectures. For instance, Köhler et al. (2020); Klein et al. (2023) develop equivariant CNFs tailored to molecular systems. However, this approach does not scale well: computing the log-likelihood requires integrating the divergence of the velocity field (as in Eq. 3), which becomes prohibitively expensive in high dimensions. The second direction instead focuses on improving the expressivity and sampling efficiency of standard NFs. Recent work, such as TarFlow (Zhai et al., 2025), introduces scalable transformer-based architectures, achieving state-of-the-art performance. Building on this line, Tan et al. (2026a,b); Schebek et al. (2025) pushed the boundaries of NFs for atomic systems. In particular, Tan et al. (2026a) also proposes an inference-time scaling strategy to further improve sampling efficiency.

Energy-based Diffusion Models. Training diffusion models with energy parameterization acts as a bridge to alleviate the computational overhead of solving the divergence integration in CNFs, where the learned energies define Boltzmann distributions on different noise levels. Recent works focus on improving the training of energy-based diffusion models beyond Denoising Score Matching. Aggarwal et al. (2025); OuYang et al. (2026a) incorporate Noise-Contrastive Estimation (Gutmann and Hyvärinen, 2012), where Aggarwal et al. (2025) treats the learned energy of clean data as an approximate normalized likelihood and OuYang et al. (2026a) leverages the learned intermediate energies for further inference-time control. He et al. (2026) improves the training by imposing a Bayes' rule constraint, with an application of the EBMs for free energy estimation. Plainer et al. (2026) regularizes the EBMs to enforce the Fokker-Planck Equation, and uses the learned EBMs as a force field. Yu et al. (2026) proposes a unified framework for learning EBMs. In contrast to the standard data-driven setting, OuYang et al. (2026b) proposes an energy matching objective for learning a Boltzmann neural sampler when data are unavailable, relying instead on the unnormalized target density.

Flow Maps with Approximate Invertibility. NFs enable exact likelihood evaluation through their invertibility, which also limits their flexibility. To enable the use of more flexible network architectures, Free-form Flows (Draxler et al., 2024) propose to softly enforce invertibility through an additional reconstruction loss. To extend this approach to recent few-step generative models, FALCON (Rehman et al., 2026) proposes to make flow maps invertible by additionally imposing reconstruction losses on the jump between two times $t < s$. FALCON enables fast yet accurate likelihood estimation with flow maps and shows its effectiveness and scalability on molecular systems. However, FALCON requires the exact log-determinant of the Jacobian of the flow map for likelihood evaluation; over T steps, this is computationally expensive, resulting in $\mathcal{O}(Td)$ function evaluations and an additional $\mathcal{O}(Td^3)$ cost.

C Experimental Details for Molecules

The molecular experiments are conducted on one `NVIDIA RTX PRO 6000` GPU.

C.1 Details of Training

Model Architecture. Following Rehman et al. (2026), we adopt the same Diffusion Transformer (DiT) backbone for both `SCALLOP` and F2D2, with additional time conditioning adapted for flow maps, across all benchmarks. We use the vectorized divergence matching objectives stated in Section 3.4 for both `SCALLOP` and F2D2. To obtain a parameter count similar to FALCON, we use 6 DiT blocks as shared parameters, and 3 additional DiT blocks for the log-density-change readout head. The details of the backbone configuration are in Table 3 below.

Hyperparameters for Training. We train all models with an exponential moving average (EMA) on the weights using a decay rate of 0.999. We use the AdamW optimizer with learning rate 10^{-4}, gradient-norm clip 10, and weight decay 10^{-4}. We employ a square-root annealing learning rate

Table 3: Overview of SCALLOP and F2D2 configurations across benchmarks.

System	Hid. Size	Shared Blocks	log-density Blocks	Heads	Cond. Dim.	Param. (M)
ALA-2	192	6	3	6	64	6.25
ALA-3	192	6	3	6	64	6.25
ALA-4	192	6	3	6	64	6.25
ALA-6	192	6	3	6	64	6.25

schedule with a warm-up phase covering 2.5% of the training iterations. All training runs use a batch size of 1000, with 4×10^6 updates in total.

Data Augmentation. To preserve SE(3) symmetries, we employ center-of-mass adjustment and standardize the data using the mean and standard deviation estimated from the training set. We further apply a random rotation to the data every time a sample is drawn, to impose a soft constraint on rotational equivariance.

C.2 Details of Inference

Hyperparameters. In all evaluations, we draw 2×10^5 samples from the model and employ SNIS for resampling. In SNIS, logit values are clipped at 0.002, as in standard baselines (Tan et al., 2026a,b; Rehman et al., 2026). We use batch size 10^4 for SCALLOP, F2D2, and SBG-IS during inference to maximize GPU utilization, for all benchmarks. However, FALCON only fits with significantly smaller batch sizes due to the need for exact Jacobian calculation. On a single NVIDIA RTX PRO 6000 GPU, FALCON only fits batch sizes (1000, 1000, 500, 100) for ALA-(2, 3, 4, 6), respectively. We employ a linear schedule for all generation in SCALLOP and F2D2.

Algorithms. We provide the pseudocode for inference of SCALLOP and F2D2 in Algorithm 1 below:

Algorithm 1 Inference for SCALLOP and F2D2

Input: base distribution p_0, trained likelihood flow map $f_\theta = [u_\theta; D_\theta]$, time schedule $\{t_i\}_{i=0}^N$
Output: Samples $x_N \sim p_1^\theta$ with associated estimated log-likelihoods ℓ_N
$x_0 \sim p_0$
$\ell_0 \leftarrow \log p_0(x_0)$
for $i = 1$ **to** N **do**
  $(t, s) \leftarrow (t_{i-1}, t_i)$
  $x_i \leftarrow x_{i-1} + (s - t)u_\theta(x_{i-1}, t, s)$
  if $D_\theta(x_{i-1}, t, s) \in \mathbb{R}$ **then**
    $\ell_i \leftarrow \ell_{i-1} + (s - t)D_\theta(x_{i-1}, t, s)$
  else if $D_\theta(x_{i-1}, t, s) \in \mathbb{R}^d$ **then**
    $\ell_i \leftarrow \ell_{i-1} + (s - t)D_\theta(x_{i-1}, t, s) \cdot \mathbf{1}_d$ ▷ for vectorized divergence
  end if
end for
return x_N, ℓ_N

C.3 Details of Alanine Systems

For all the alanine systems, including ALA-(2, 3, 4, 6), we follow the same setting as Tan et al. (2026a); Rehman et al. (2026) to split the data into training, validation, and test sets, with sizes 10^5, 2×10^4, and 10^4. Data are obtained from MD for 1 μs, with a timestep of 1 fs, at 300 K, 310 K, 300 K, 310 K for ALA-(2, 3, 4, 6) respectively, provided by the official repository of Tan et al. (2026a). The force fields for ALA-(2, 3, 4, 6) are Amber ff99SBildn, Amber 14, and Amber ff99SBildn, respectively.

Data provenance and licenses. We obtain the molecular trajectories from the Sequential Boltzmann Generators data release, whose dataset card specifies the MIT License. The release credits the ALA-2

(Ace-A-Nme) trajectories to Köhler et al. (2021), the ALA-4 (Ace-AAA-Nme) trajectories to Dibak et al. (2021), and the ALA-3 (AAA) and ALA-6 (AAAAAA) trajectories to Tan et al. (2026a). The core Transferable Samplers codebase is also MIT-licensed; third-party components retain the separate licenses identified in its NOTICE file and individual file headers.

C.4 Details of Metrics

Effective Sample Size. ESS (Kish, 1957) quantifies the resampling efficiency of SNIS. Given N generated samples $x_{1:N}$ with associated unnormalized importance weights $w_{1:N}$, the ESS is defined as:

$$\mathrm{ESS}(w_{1:N}) = \frac{(\sum_{n=1}^{N} w_n)^2}{N \sum_{n=1}^{N} w_n^2} \in [0, 1], \tag{152}$$

where a higher ESS generally reflects better SNIS performance. Therefore, ESS can be treated as a proxy for evaluating how accurate the estimated log densities of samples are, given that the model can generate high-quality samples.

2-Wasserstein Distance. The 2-Wasserstein distance ($\mathcal{W}_2$) is a measure of difference between two probability distributions p and q defined on $\mathbb{R}^m$, defined as follows

$$\mathcal{W}_2(p, q)^2 = \min_{\pi \in \Pi(p,q)} \int_{\mathbb{R}^m \times \mathbb{R}^m} c(x, y) \mathrm{d}\pi(x, y), \tag{153}$$

where Π is the set of admissible couplings and $c : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is the cost function.

- 2-Wasserstein Distance of energy ($\mathcal{E}$-$\mathcal{W}_2$) measures the energy histogram difference between generated samples and reference ones, which admits $m = 1$ and uses $c(x, y) = |x - y|^2$.
- 2-Wasserstein Distance of torus ($\mathbb{T}$-$\mathcal{W}_2$) measures the torsional-angle histogram difference between generated samples and reference ones. For an ALA-N system, there are $2(N-1)$ torsional angles in total. Let $\mathrm{Dihedral}(x) = (\phi_1, \psi_1, ..., \phi_{N-1}, \psi_{N-1}) \in [-\pi, \pi]^{2(N-1)}$ be the torsional angles of a sample x. To account for the fact that these torsional angles lie on the torus $[-\pi, \pi]^{2(N-1)}$ with periodic boundaries, we use:

$$c(x, y) = \sum_{i=1}^{2(N-1)} [(\mathrm{Dihedral}(x)_i - \mathrm{Dihedral}(y)_i + \pi) \mathrm{mod} 2\pi - \pi]^2 \tag{154}$$

  as the cost function.

D Additional Experimental Results for Molecules

Recall that we employ (4, 8, 8, 16) steps for ALA-(2, 3, 4, 6) systems, following Rehman et al. (2026).

Ground Truth Torsional Angles Histograms. We provide the Ramachandran plots, *i.e.* the histograms of torsional angles, of the training data, from ALA-2 to ALA-6, in Figure 3.

Resampled Torsional Angles Histograms. We provide the Ramachandran plots for the resampled data in Figure 4 (for `SCALLOP`) and Figure 5 (for F2D2).

Resampled Energy Histogram. We provide the energy histograms of the resampled data in Figure 6a (for `SCALLOP`) and Figure 6b (for F2D2).

E Experimental Details for Images

We use CelebA-64 Liu et al. (2015) as the image-generation benchmark and report both likelihood-based and sample-quality metrics. Unless otherwise specified, the model architecture and optimization hyperparameters follow the configuration summarized in Table 4.

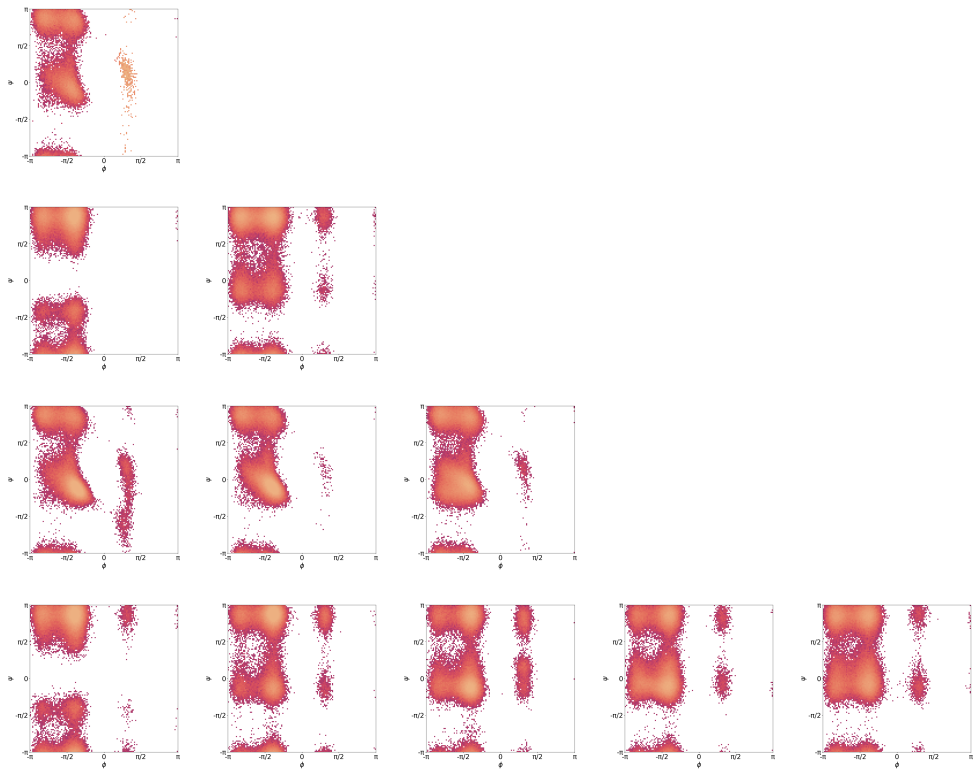


Figure 3: Ramachandran plots for ALA-N systems, from ALA-2 (TOP) to ALA-6 (BOTTOM). Torsional angle indices are ordered from LEFT to RIGHT.

Dataset terms of use. CelebA is provided for non-commercial research purposes only under the dataset authors' usage agreement, which also restricts redistribution of the dataset. We cite the original dataset paper above.

LSD Model Training. For the vanilla LSD baseline, we follow the experimental setup of Boffi et al. (2026). We adopt the batch-allocation strategy used in Ai et al. (2026): during the first 200k iterations, 75% of each batch is used for the flow-matching loss and the remaining 25% is used for the self-distillation loss; during the subsequent 150k iterations, the allocation is changed to 50% for flow matching and 50% for self-distillation. For the self-distillation objective, we sample two time steps independently and uniformly from $[0, 1]$, without imposing an ordering constraint between them. Equivalently, the full time-step pair is sampled as $(t, s) \sim \mathcal{U}([0, 1]^2)$, which allows the model to learn both forward and backward estimators.

SCALLOP Training. For `SCALLOP`, we follow the same image-generation training setup as Ai et al. (2026) and initialize the model from the pretrained LSD model, adding only the divergence-prediction head. We train the model for 60k iterations with an initial learning rate of 10^{-4}. At each iteration, 50% of the batch is used for the flow-matching loss, while the remaining 50% is used for the self-distillation loss. The vectorized divergence head is implemented on the decoder of the image UNet: instead of producing a single scalar divergence estimate, it applies a 1×1 convolution to predict a spatially resolved divergence field with the same shape as the noisy image state, namely (B, C, H, W).

FID Evaluation. We compute FID using 50k images generated by each model.

BPD Evaluation. We compute bits per dimension (BPD) on the full CelebA-64 test set, following the procedure of Lipman et al. (2023). For numerical integration, we discretize the reverse-time interval from $t = 1$ to $t = 0$ into equally spaced segments.

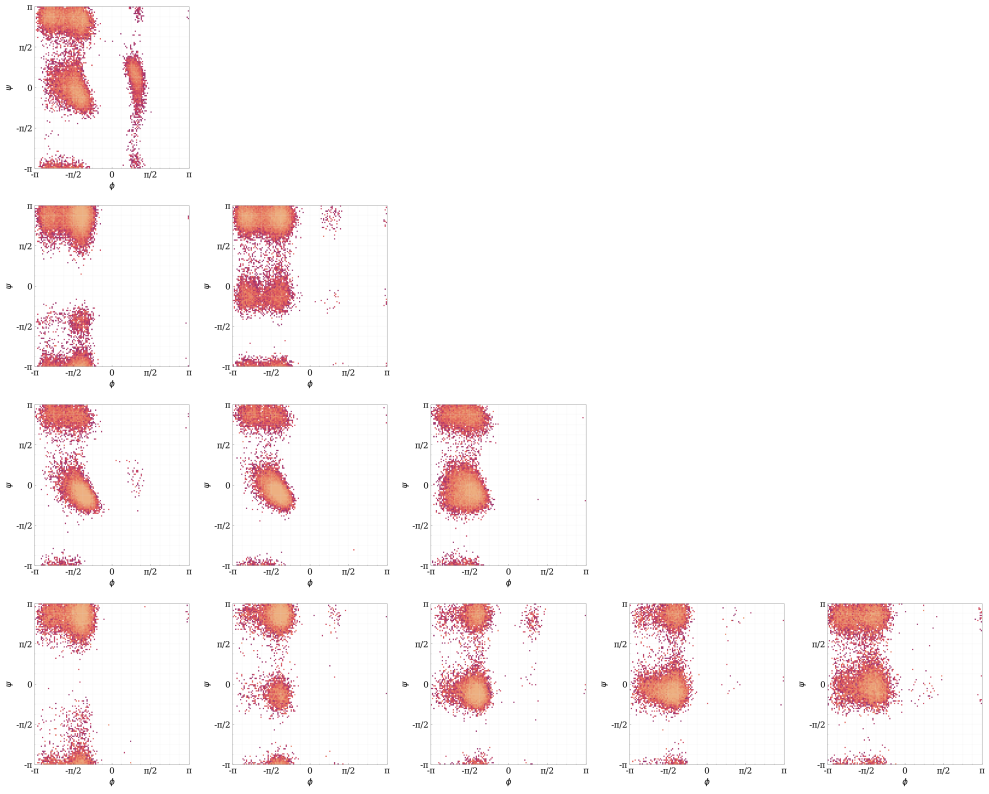


Figure 4: Ramachandran plots for ALA-N data generated by SCALLOP, from ALA-2 (TOP) to ALA-6 (BOTTOM). Torsional angle indices are ordered from LEFT to RIGHT.

F Additional Experimental Results for Images

We train SCALLOP on CelebA-64 and compare to F2D2.

Stability of Training. Figure 7 compares the training dynamics of F2D2 and SCALLOP over the first 60k training steps, including both the training loss and the gradient norm. The results show that SCALLOP has more stable training dynamics with lower variance than F2D2, which is consistent with the observation in molecular systems.

Image Generation Examples. We provide qualitative generation results of SCALLOP on unconditional CelebA-64 in Figures 8 to 11 , using 1, 2, 4, and 8 sampling steps respectively.

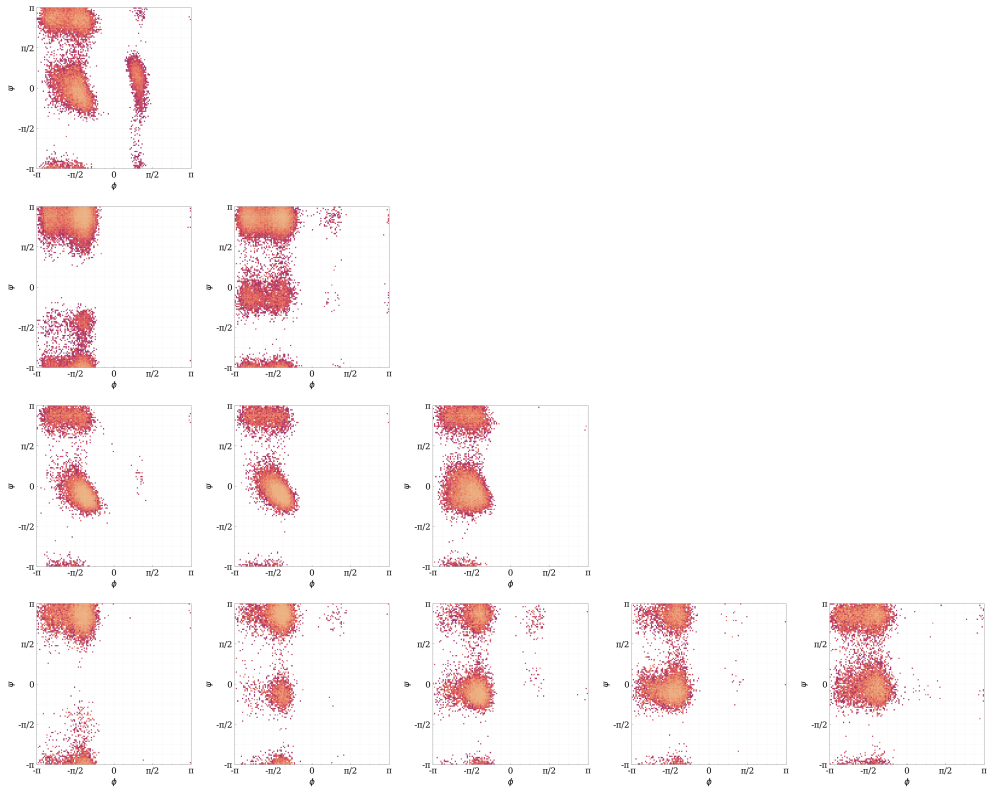
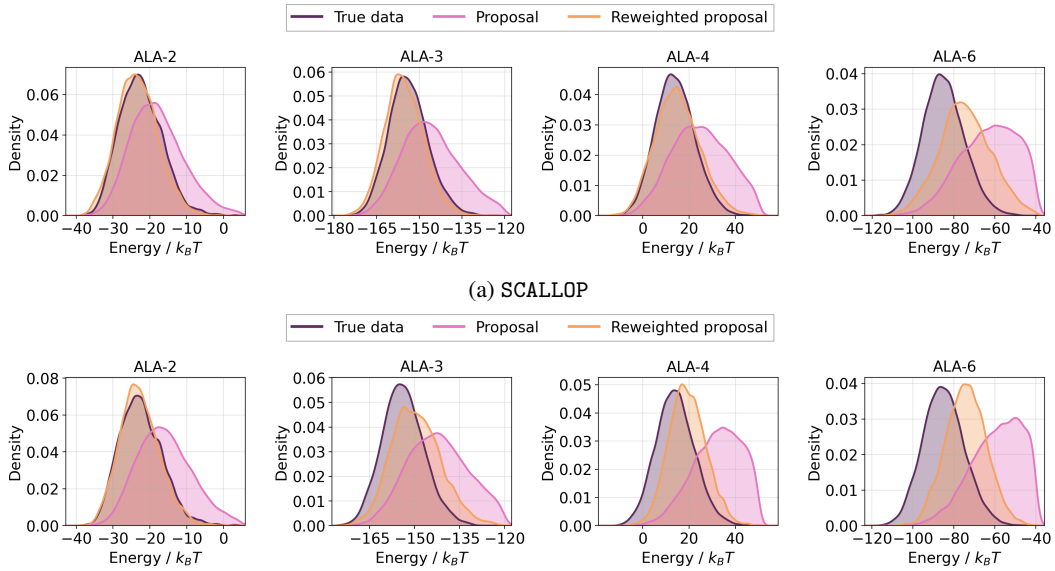


Figure 5: Ramachandran plots for ALA-N data generated by F2D2, from ALA-2 (TOP) to ALA-6 (BOTTOM). Torsional angle indices are ordered from LEFT to RIGHT.



(a) SCALLOP

(b) F2D2

Figure 6: Energy histograms of resampled data from SCALLOP (TOP) and F2D2 (BOTTOM), ranging from ALA-2 to ALA-6.

Table 4: Training hyperparameters on CelebA-64.

	CelebA-64
Noise embedding	Positional
Channels	128
Channel multipliers	1,2,3,4
Attention resolution	16,8
Residual blocks	3
Dropout	0.0
Batch size	256
GPUs	4 A6000s
Iterations	350k
Learning Rate	1e-2 (Sqrt decay at 35k)
Precision	bfloat16
Optimizer	RAdam
EMA rate	0.9999

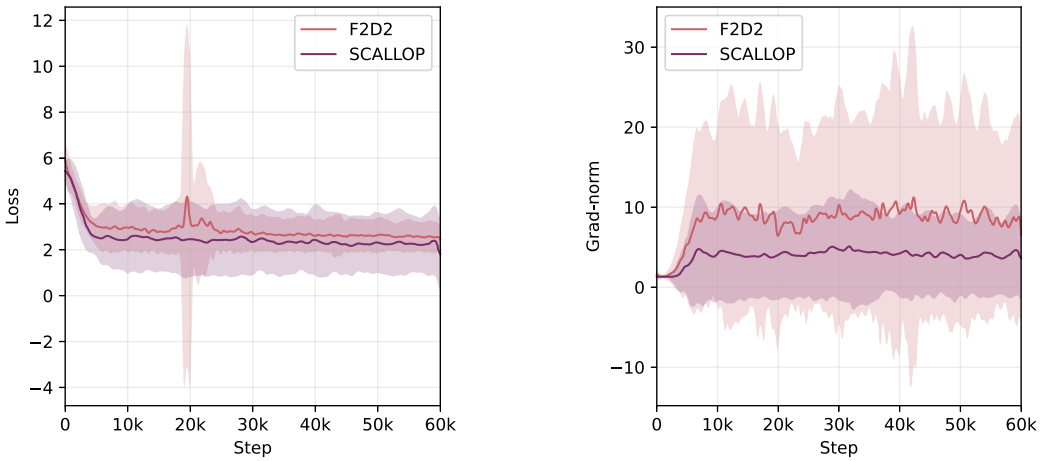


Figure 7: Training dynamics comparison between F2D2 and SCALLOP over 60k training steps. **Left:** Training loss. **Right:** Training gradient norm. Curves are smoothed for visualization, and shaded regions indicate local variability.

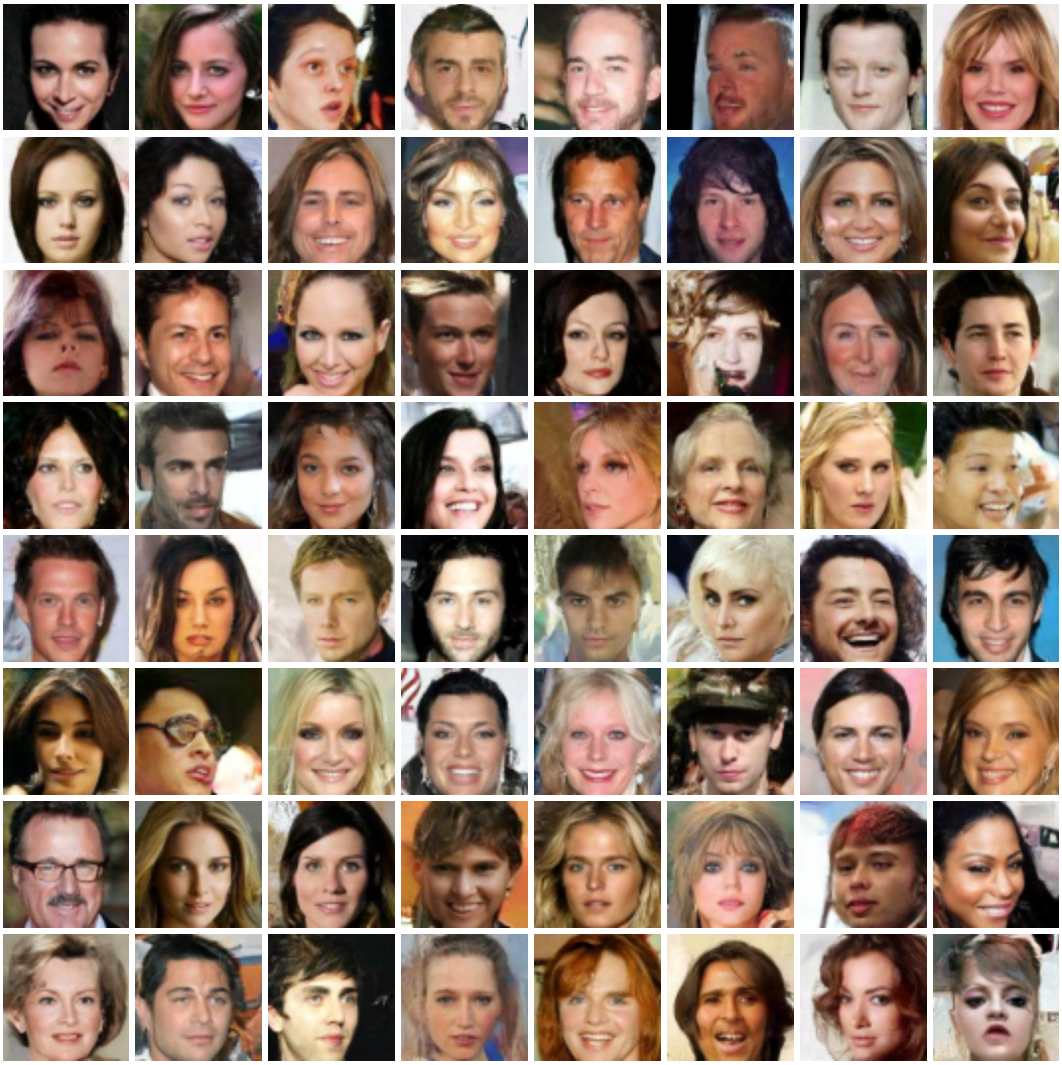


Figure 8: 1-step unconditional CelebA-64 generation with SCALLOP.

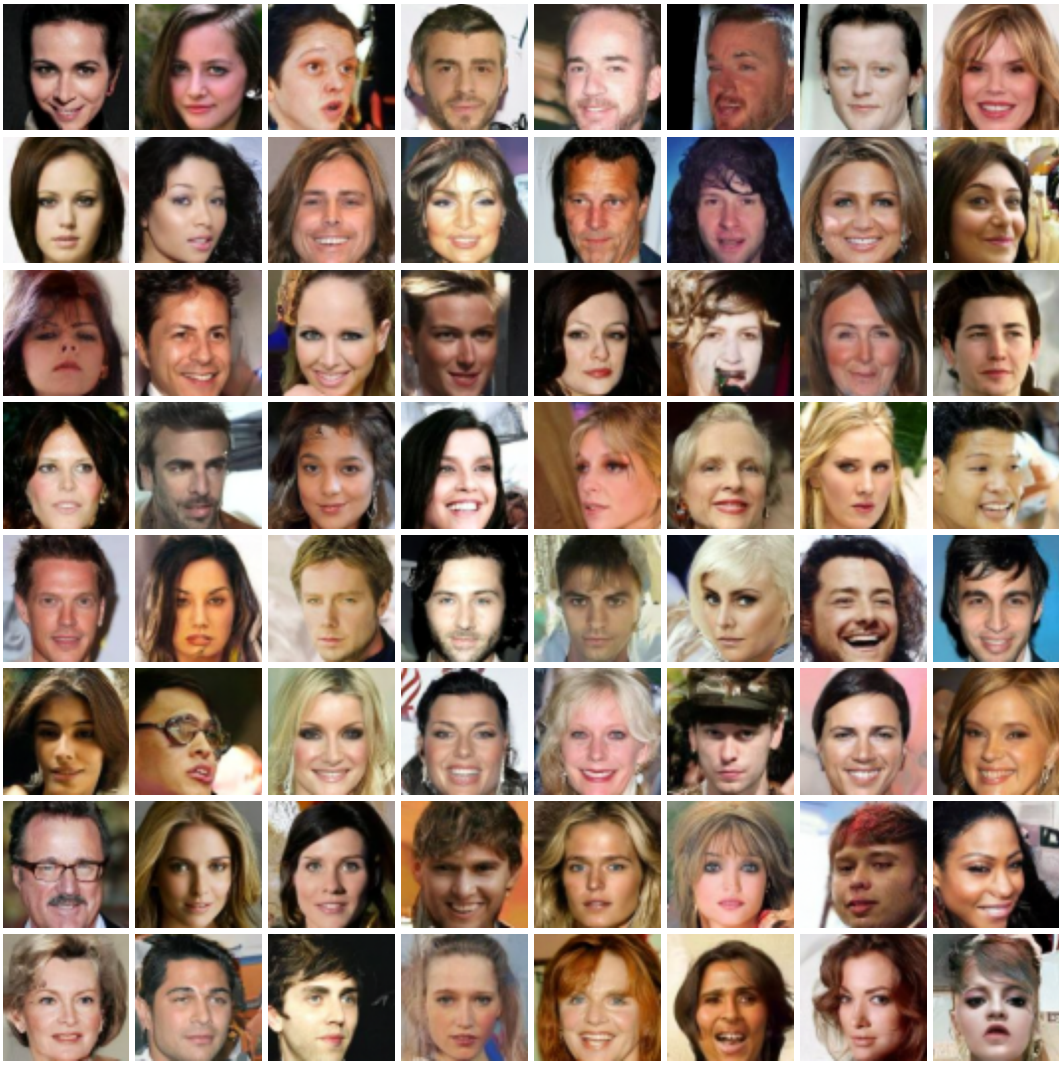


Figure 9: 2-step unconditional CelebA-64 generation with SCALLOP.

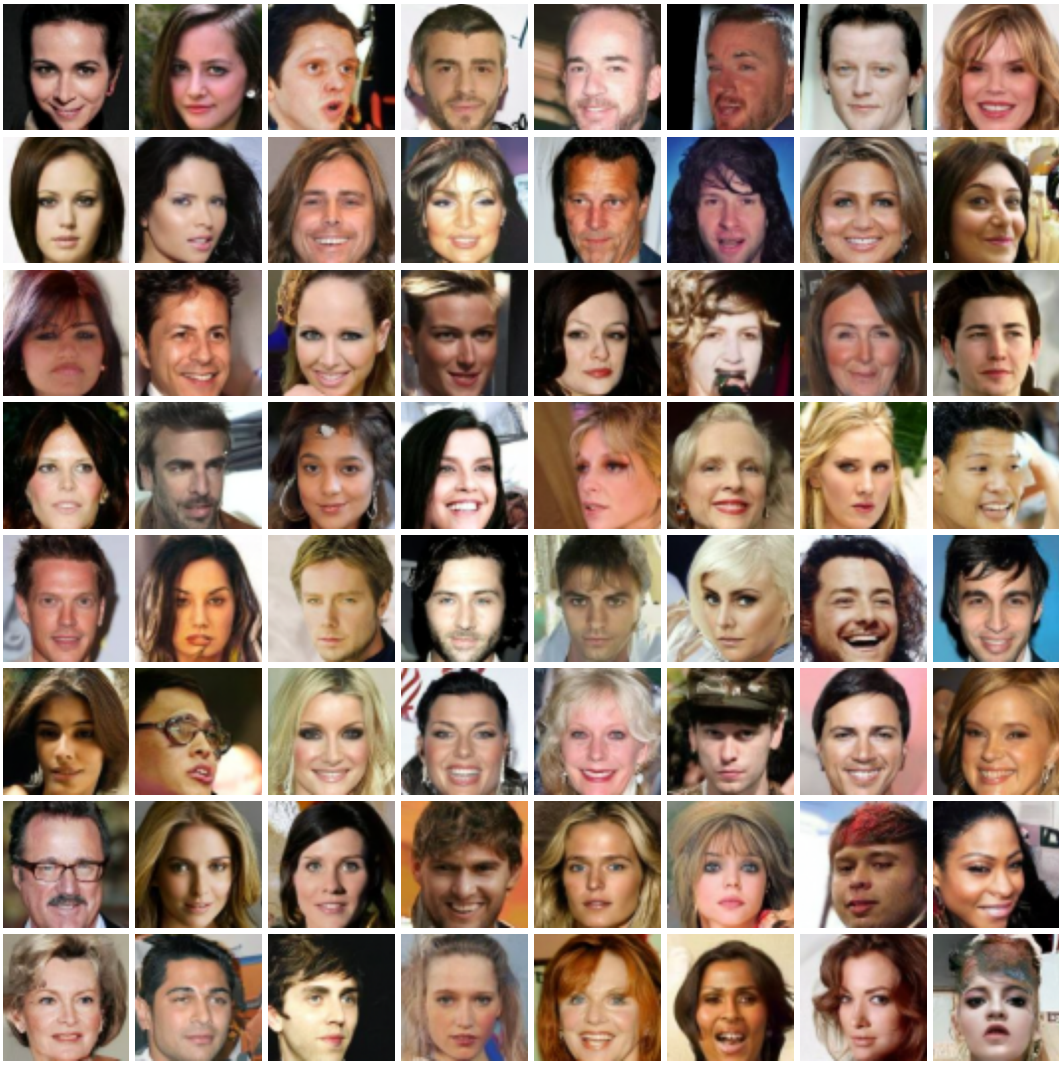


Figure 10: 4-step unconditional CelebA-64 generation with SCALLOP.

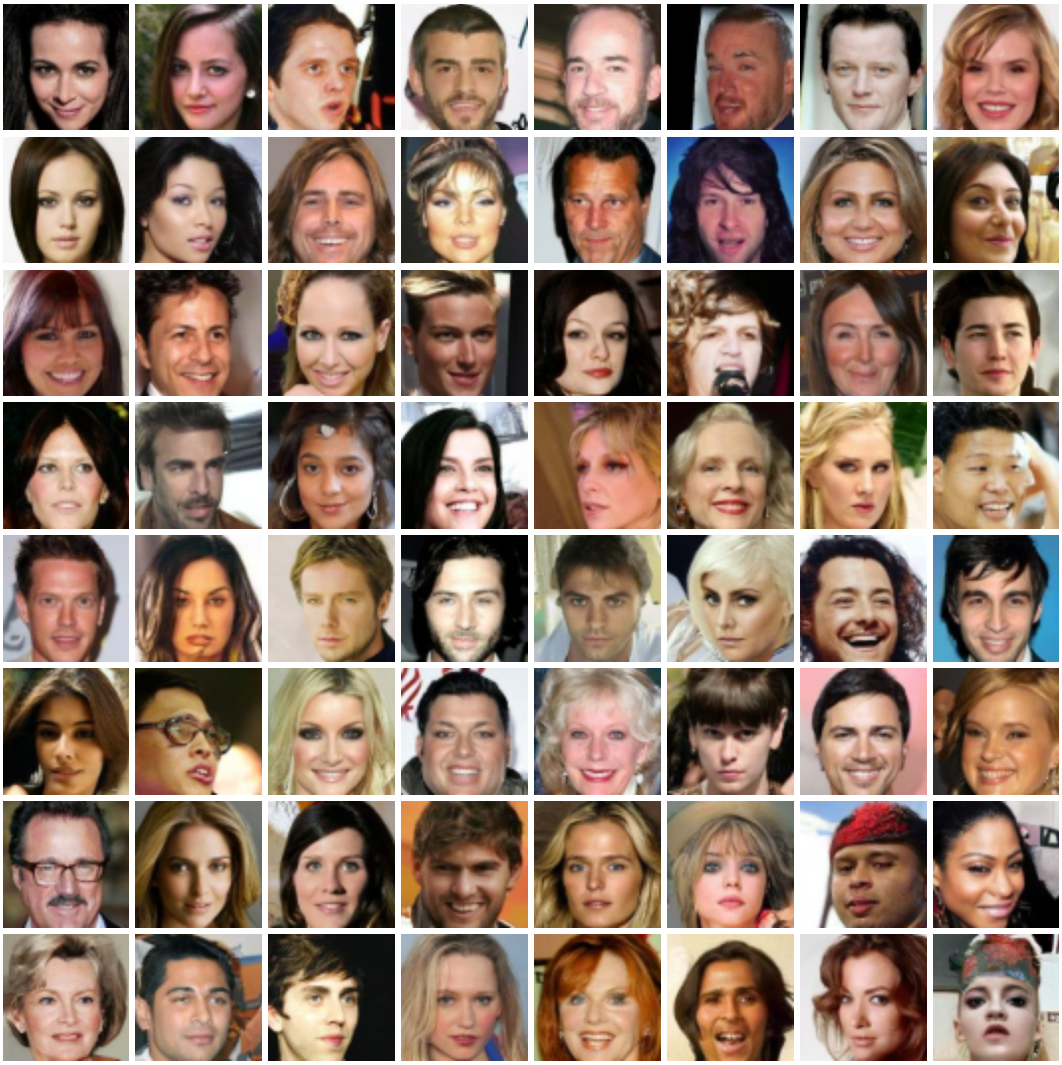


Figure 11: 8-step unconditional CelebA-64 generation with SCALLOP.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction reflect the contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our method.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We report our results and proofs correctly, to the best of our ability.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We disclose the information required to reproduce our experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our code is publicly available at https://github.com/tonyauyeung/scallop. Experimental settings and data sources are described in Appendices C and E.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We report sufficient training and test details to reproduce our experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We include standard deviations in our result tables, and shaded regions of uncertainty in our graphs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details of the computational resources used for the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is fundamental research and, to the best of our knowledge, does not present any foreseeable societal consequences.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We credit the original molecular trajectory sources and document the MIT license of the Sequential Boltzmann Generators data release, along with the Transferable Samplers code licensing, in Appendix C. CelebA is provided under non-commercial research terms, which we describe and link in Appendix E.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release the SCALLOP implementation at https://github.com/tonyauyeung/scallop under the MIT License. The repository includes environment setup, training and evaluation instructions, and experiment configurations.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This research does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This research does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used for polishing text, not for ideation.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.