Scalable Equilibrium Sampling with Sequential Boltzmann Generators

Charlie B. Tan [*1] **Avishek Joey Bose** [*1 2] **Chen Lin** [1] **Leon Klein** [3] **Michael M. Bronstein** [1 4] **Alexander Tong** [2 5]

Abstract

Scalable sampling of molecular states in thermodynamic equilibrium is a long-standing challenge in statistical physics. Boltzmann generators tackle this problem by pairing normalizing flows with importance sampling to obtain uncorrelated samples under the target distribution. In this paper, we extend the Boltzmann generator framework with two key contributions, denoting our framework SEQUENTIAL BOLTZMANN GENERATORS (SBG). The first is a highly efficient Transformer-based normalizing flow operating directly on all-atom Cartesian coordinates. In contrast to the equivariant continuous flows of prior methods, we leverage exactly invertible non-equivariant architectures which are highly efficient during both sample generation and likelihood evaluation. This efficiency unlocks more sophisticated inference strategies beyond standard importance sampling. In particular, we perform inference-time scaling of flow samples using a continuous-time variant of sequential Monte Carlo, in which flow samples are transported towards the target distribution with annealed Langevin dynamics. SBG achieves state-of-the-art performance w.r.t. all metrics on peptide systems, demonstrating the first equilibrium sampling in Cartesian coordinates of tri-, tetra- and hexa-peptides that were thus far intractable for prior Boltzmann generators.

1. Introduction

The sampling of molecular systems at the all-atom resolution is of central interest in understanding complex natural processes. These include important biophysical processes such as protein-folding (Noé et al., 2009; Lindorff-Larsen et al., 2011), protein-ligand binding (Buch et al., 2011), and formation of crystal structures (Parrinello & Rahman, 1980; Matsumoto et al., 2002), whose understanding can aid in problems that range from long-standing global health challenges, to efficient energy storage (Deringer, 2020).

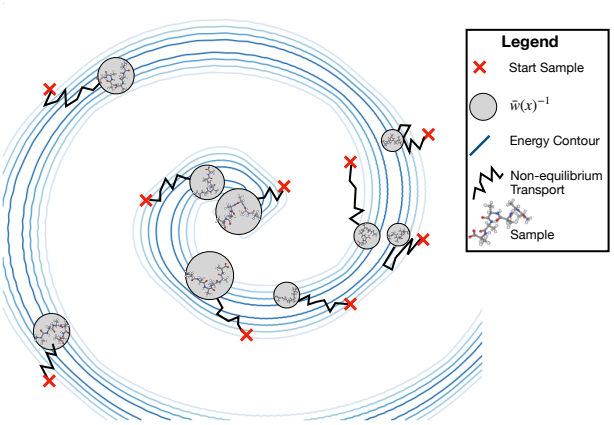


Figure 1. SBG uses annealed Langevin dynamics to transport proposal flow samples towards towards the target distribution.

The dominant paradigm for molecular sampling involves running Markov chain Monte Carlo (MCMC) or molecular dynamics (MD), whereby the equations of motion are integrated with finely discretized time steps. However, such molecular systems often exist in thermodynamic equilibrium by remaining for extended periods in metastable states. Such metastable states are captured in the minima of a complex energy landscape, itself defining the molecular system's equilibrium (Boltzmann) distribution at a given temperature. The high-energy barriers separating metastable states lead to infrequent state transitions (Wirnsberger et al., 2020), presenting an obstacle for effective sampling with simulation-based methods such as molecular dynamics or MCMC, requiring long simulation periods with small time steps on the order of femtoseconds $1\,\text{fs} = 10^{-15}\text{s}$.

Boltzmann generators (BG) (Noé et al., 2019) offer an alternative approach, in which powerful generative models, such as normalizing flows (Dinh et al., 2017; Rezende & Mohamed, 2015), are trained on existing (but assumed to be biased) datasets, and leveraged as a proposal for self-normalized importance sampling (SNIS), targeting the de-

[*]Equal contribution [1]University of Oxford [2]Mila – Québec AI Institute [3]Freie Universität Berlin [4]AITHYRA [5]Université de Montréal. Correspondence to: CBT <charlie.tan@exeter.ox.ac.uk>, AJB <joey.bose@mail.mcgill.ca>, AT <alexandertongdev@gmail.com>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

sired Boltzmann distribution. Boltzmann generators permit accelerated sampling through amortization as the uncorrelated proposal generation avoids the slow state transitions suffered by MD and MCMC. Despite their appeal, it remains challenging for existing BGs to model systems beyond the smallest peptides (2 amino acids) in Cartesian coordinates (Klein et al., 2023b; Midgley et al., 2023a). The principal drawback inhibiting scalability stems from the lack of expressive equivariant architectures that are also exactly invertible (Bose et al., 2021; Midgley et al., 2023a), or the present over-reliance on simple E(n)-GNN (Satorras et al., 2021) based equivariant vector fields in continuous-time normalizing flows (Chen et al., 2018). As a result, even the most performant BGs suffer from poor target distribution overlap, leading to low sampling efficiency during SNIS.

Present work. In this paper, we introduce SEQUENTIAL BOLTZMANN GENERATORS (SBG) a novel extension to the existing Boltzmann generator framework.[1] SBG makes progress on the scalability of Boltzmann generators in Cartesian coordinates along two complementary axes: (1) scalable pre-training of softly SE(3)-equivariant proposal normalizing flows in BGs; and (2) inference time scaling via continuous-time variants of annealed importance sampling (AIS) (Neal, 2001) and sequential Monte Carlo (SMC) (Doucet et al., 2001). The use of AIS or SMC over SNIS enables more effective sampling given suboptimal proposal-target overlap, enabling SBG to draw uncorrelated $\mu_{\text{target}}(x)$ samples for peptide system up to 6 residues.

Table 1. Method overview for samplers, given biased data samples.

Method	Use $\mathcal{E}(x)$	Exact likelihoods	Use data	Annealing
DEM (Akhound-Sadegh et al., 2024)	✓	✗	✗	✗
NETS (Albergo & Vanden-Eijnden, 2025)	✓	✓	✗	✓
BG (Noé et al., 2019)	✓	✓	✓	✗
SBG (Ours)	✓	✓	✓	✓

SBG scales up normalizing flows in BGs by following recent advances in atomistic generative modeling (Abramson et al., 2024). In particular, we remove the rigid SE(3)-equivariance as an explicit architectural inductive bias in favor of softly enforcing it through simpler and more efficient data augmentations. To further improve sampling we perform inference-time scaling by defining an interpolation between the proposal flow energy distribution (i.e., negative log density of samples) and the known target Boltzmann energy. Crucially, simulating samples at inference via annealed Langevin dynamics may be coupled to a corresponding time evolution of importance weights, converting naturally to continuous-time variants of the well-established annealed importance sampling (AIS) (Neal, 2001) and sequential Monte Carlo (SMC) (Doucet et al., 2001). As a result, SBG can readily improve over the simple one-step importance sampling methodology used in existing BGs.

[1]We open source our full codebase at https://github.com/charliebtan/transferable-samplers.

We summarize the different aspects of our proposed SBG in comparison to other learned samplers in Table 1.

We instantiate SBG using a best-in-class, general-purpose, *non-equivariant* normalizing flow, named TarFlow (Zhai et al., 2024). TarFlow is a modernized normalizing flow architecture employing a scalable transformer backbone to parameterize an exactly invertible transformation. We demonstrate that such exactly invertible architectures, via fast and accurate log-likelihood evaluation, benefit from inference-scaling. We emphasize this is in stark contrast to continuous normalizing flows that underpin prior SOTA Boltzmann generators which require both the costly simulation of the 2nd order divergence operator as well as differentiation of an ODE solver. Furthermore, we demonstrate that enforcing equivariance softly along enables us to stably scale proposal flows in SBG, far beyond prior BGs. On a theoretical front, we study a novel inference-time proposal energy adjustment to counteract the influence of training data centroid augmentation when resampling, as well as quantify the additional bias of common thresholding tricks employed to improve resampling numerical stability. Empirically, we observe SBG to achieve state-of-the-art results across metrics, far outperforming continuous BGs on all datasets. In particular, SBG is the first uncorrelated learned sampler to scale successfully in Cartesian coordinates to tripeptides, tetrapeptides, hexapeptides, and makes significant progress towards equilibrium sampling of decapeptides.

2. Background and Preliminaries

We are interested in drawing statistically independent samples from the target Boltzmann distribution μ_{target}, with partition function $\mathcal{Z}$, defined over $\mathbb{R}^{n\times 3}$:

$$\mu_{\text{target}}(x) \propto \exp\left(\frac{-\mathcal{E}(x)}{k_{\text{B}}T}\right), \mathcal{Z} = \int_{\mathbb{R}^d} \exp\left(\frac{-\mathcal{E}(x)}{k_{\text{B}}T}\right) dx.$$

The Boltzmann distribution is defined for a given system and includes the Boltzmann constant k_{B}, and a specified temperature T. Additionally, the potential energy of the system $\mathcal{E} : \mathbb{R}^{n\times 3} \to \mathbb{R}$ and its gradient $\nabla\mathcal{E}$ can be evaluated at any point $x \in \mathbb{R}^{n\times 3}$, but the exact density $\mu_{\text{target}}(x)$ is not available as the partition function $\mathcal{Z}$ evaluation is intractable for all but the simplest systems.

In this paper, unlike pure sampling-based settings, we are afforded access to a small biased dataset of N samples $\mathcal{D} = \{x^i\}_{i=1}^N$, provided as an empirical distribution $p_{\mathcal{D}}$. Consequently, it is possible to perform an initial learning phase that fits a generative model p_θ, with parameters θ, to $p_{\mathcal{D}}$—e.g. by minimizing the forward KL $\mathbb{D}_{\text{KL}}(p_{\mathcal{D}}||p_\theta)$—to act as a proposal distribution that can be corrected.

2.1. Normalizing Flows

A key desirable property needed for the correction of a trained generative model p_θ on a biased dataset $\mathcal{D}$ is the ability to extract an exact likelihood $p_\theta(x)$. Normalizing flows (Dinh et al., 2017; Rezende & Mohamed, 2015) represent exactly such a model class as they learn to transform an easy-to-sample base density to a desired target density using a parametrized diffeomorphism. More formally, given a sample from a (prior) base density $x_0 \sim p_0$ and a diffeomorphism $f_\theta : \mathbb{R}^{n\times 3} \to \mathbb{R}^{n\times 3}$ that maps the initial sample to $x_1 = f_\theta(x_0)$. We can obtain an expression for the log density of x_1 via the classical change of variables,

$$\log p_1(x_1) = \log p_0(x_0) - \log\det\left|\frac{\partial f_\theta(x_0)}{\partial x_0}\right|. \quad (1)$$

In Eq. 1 above the $\log\det|\cdot|$ term corresponds to the Jacobian determinant of f_θ evaluated at x_0. Optimizing Eq. 1 is the maximum likelihood objective for training normalizing flows and results in f_θ learning $p_1 \approx p_{\text{data}}$. There are multiple ways to construct the (flow) map f_θ. Perhaps the most popular approach is to consider the flow to be a composition of a finite number of elementary diffeomorphisms $f_\theta = f_M \circ f_{M-1} \cdots \circ f_1$, resulting in the change in log density to be: $\log p_1(x_1) = \log p_0(x_0) - \sum_{i=1}^{M} \log|\partial f_{i,\theta}(x_{i-1})/\partial x_{i-1}|$. We note that the construction of each $f_{i,\theta}, i \in [M]$ is motivated such that both the inverse $f_{i,\theta}^{-1}(x)$ and Jacobian $\partial f_{i,\theta}(x)/\partial x$ are computationally cheap to compute.

Continuous normalizing flows. In the limit of infinite elementary diffeomorphisms, a normalizing flow transforms into a continuous normalizing flow (CNF) (Chen et al., 2018). Formally, a *flow* is a one-parameter time-dependent diffeomorphism $\psi_t : [0,1] \times \mathbb{R}^{n\times 3} \to \mathbb{R}^{n\times 3}$ that is the solution to the following ordinary differential equation (ODE): $\frac{d}{dt}\psi_t(x) = u_t(\psi_t(x))$, with initial conditions $\psi_0(x_0) = x_0$, for a time-dependent vector field $u_t : [0,1] \times \mathbb{R}^{n\times 3} \to \mathbb{R}^{n\times 3}$. It is often desirable to construct the target flow by associating it to a designated *probability path* $p_t : [0,1] \times \mathbb{P}(\mathbb{R}^{n\times 3}) \to \mathbb{P}(\mathbb{R}^{n\times 3})$ which is a time-indexed interpolation in probability space between two distributions $p_0, p_1 \in \mathbb{P}(\mathbb{R}^{n\times 3})$. In such cases, the flow ψ_t is said to generate p_t if it pushes forward p_0 to p_1 by following u_t — $p_t = [\psi_t]_\#(p_0)$. As ψ_t is a valid flow and satisfies an ODE the change in log density can be computed using the instantaneous change of variables:

$$\log p(x_1) = \log p(x_0) - \int_0^1 \nabla \cdot u_t(x_t)dt, \quad (2)$$

where $x_t = \psi_t(x_0)$ and $\nabla\cdot$ is the divergence operator.

A CNF can then be viewed as a neural flow that seeks to learn a designated target flow ψ_t for all time $t \in [0,1]$. The most scalable way to train CNFs is to employ a flow-matching learning framework (Liu, 2022; Albergo & Vanden-Eijnden, 2023; Lipman et al., 2023; Tong et al., 2024). Specifically, flow-matching regresses a learnable vector field of a CNF $f_{t,\theta}(t,\cdot) : [0,1] \times \mathbb{R}^{n\times 3} \to \mathbb{R}^{n\times 3}$ to the target vector field $u_t(x_t)$ associated to the flow ψ_t. In practice, it is considerably easier to regress against a target *conditional* vector field $u_t(x_t|z)$—which generates the conditional probability path $p_t(x_t|z)$—as we do not have closed form access to the (marginal) vector field u_t which generates p_t. The conditional flow-matching (CFM) objective can then be stated as a simple simulation-free regression,

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(z),p_t(x_t|z)}\|f_{t,\theta}(t,x_t) - u_t(x_t|z)\|_2^2. \quad (3)$$

The conditioning distribution $q(z)$ can be chosen from any valid coupling, for instance, the independent coupling $q(z) = p(x_0)p(x_1)$. We highlight that Eq. 3 allows for greater flexibility in $f_{t,\theta}$ as there is no exact invertibility constraint. To generate samples and their corresponding log density according to the CNF we may solve the following flow ODE numerically with initial conditions $x_0 = \psi_0(x_0)$ and $c = \log p_0(x_0)$, which is the log density under the prior:

$$\frac{d}{dt}\begin{bmatrix}\psi_{t,\theta}(x_t)\\ \log p_t(x_t)\end{bmatrix} = \begin{bmatrix}f_{t,\theta}(t,x_t)\\ -\nabla\cdot f_{t,\theta}(t,x_t)\end{bmatrix}. \quad (4)$$

2.2. Boltzmann Generators

A Boltzmann generator (Noé et al., 2019) μ_θ pairs a normalizing flow as the proposal generative model p_θ, which is then corrected to obtain i.i.d. samples under μ_{target} using self-normalized importance sampling. More precisely, as normalizing flows are exact likelihood models, BG's first draw K independent samples $x^i \sim p_\theta(x), i \in [K]$ and compute the corresponding (unnormalized) importance weights for each sample $w(x^i) = \exp\left(\frac{-\mathcal{E}(x^i)}{k_{\text{BT}}}\right)/p_\theta(x^i)$. Leveraging the importance weights we can compute a Monte-Carlo approximation to any observable $\phi(x)$ of interest under μ_{target} using self-normalized importance sampling as follows:

$$\mathbb{E}_{\mu_{\text{target}}(x)}[\phi(x)] = \mathbb{E}_{p_\theta}[\phi(x)\bar{w}(x)] \approx \frac{\sum_{i=1}^K w(x^i)\phi(x^i)}{\sum_{i=1}^K w(x^i)}.$$

In addition, computing importance weights also enables resampling the pool of samples according to the collection of normalized importance weights $W = \{\bar{w}(x^i)\}_{i=1}^K$.

3. Sequential Boltzmann Generators

We now present SBG, which extends and improves over classical Boltzmann generators by including an annealing process to transport proposal samples towards the target distribution. We begin by identifying the key limitation in current BGs as SNIS with a suboptimal proposal. Indeed, while the SNIS estimator is consistent, its efficacy is highly

dependent on the overlap between proposal p_θ and target μ_{target}, where the optimal proposal is proportional to the minimizer of the variance of $\phi(x^i)\mu_{\text{target}}(x^i)$ (Owen, 2013). Unfortunately, since p_θ within a BG is trained on a biased dataset $\mathcal{D}$ the importance weights typically exhibit large variance, resulting in a small effective sample size (ESS).[2]

We address the need for more flexible proposals in §3.1 with modernized scalable training recipes for atomistic normalizing flows. In §3.2 we outline our novel application of non-equilibrium sampling with sequential Monte Carlo (Doucet et al., 2001). We term the overall process of combining a pre-trained Boltzmann generator with inference scaling through annealing SEQUENTIAL BOLTZMANN GENERATORS.

Symmetries of molecular systems. The energy function $\mathcal{E}(x)$ in a molecular system using classical force fields is invariant under global rotations and translation, which corresponds to the group $\mathrm{SE}(3) \cong \mathrm{SO}(3) \ltimes (\mathbb{R}^3, +)$. Unfortunately, $\mathrm{SE}(3)$ is a non-compact group which does not allow for defining a prior density $p_0(x_0)$ on $\mathbb{R}^{n\times 3}$. Equivariant generative models circumvent this issue by defining a mean-free prior which is a projection of a Gaussian prior $\mathcal{N}(0, I)$ onto the subspace $\mathbb{R}^{(n-1)\times 3}$ (Garcia Satorras et al., 2021). Thus pushing forward a mean free prior with an equivariant flow provably leads to an invariant proposal $p_1(x_1)$ (Köhler et al., 2020; Bose et al., 2021). We next build BGs departing from exactly equivariant maps by considering soft equivariance, unlocking scalable and efficient architectures.

3.1. Scaling Training of Boltzmann Generators

To improve proposal flows in SBG we favor scalable architectural choices that are more expressive than exactly equivariant ones. We motivate this choice by highlighting that many classes of normalizing flow models are known to be universal density approximators (Teshima et al., 2020; Lee et al., 2021). Thus, expressive enough non-equivariant flows *can learn to approximate any equivariant map*.

Soft equivariance. We instantiate SBG with a state-of-the-art TarFlow (Zhai et al., 2024) which is based on block-wise masked autoregressive flow (Papamakarios et al., 2017) based on a causal Vision Transformer (ViT) (Alexey, 2021) modified for molecular systems where patches are over the particle dimension. Since the data comes mean-free we further normalize the data to unity standard deviation. Combined, this allows us to scale both the depth and width of the models stably as there is no tension between a hard equivariance constraint and the invertibility of the network.

We include a series of strategies to improve training of non-equivariant flows by softly enforcing $\mathrm{SE}(3)$-equivariance. First, we softly enforce equivariance to global rotations through data augmentation by sampling random rotations $R \in \mathrm{SO}(3)$ and applying them to data samples $R \circ x_1 \sim p_1(x_1)$. Secondly, as the data is mean-free and has $(n-1)\times 3$ degrees of freedom, we lift the data dimensionality back to n by adding noise to the center of mass. This allows us to easily train with a non-translational equivariant prior distribution such as the standard normal $p_0 = \mathcal{N}(0, I)$. More precisely, a data sample is constructed $x = R\bar{x} + c$, where $\bar{x} \in \mathbb{R}^{(n-1)\times 3} \hookrightarrow \mathbb{R}^{n\times 3}$ is the mean-free data point embedded in $\mathbb{R}^{n\times 3}$, $R \in \mathrm{SO}(3)$, and $c \sim \mathcal{N}(0, \sigma^2)$. At inference, the impact of this center of mass noise is that we must account for $p(\|c\|)$, which follows a χ_3 distribution in three dimensions. Consequently, during reweighting we adjust the proposal energy to account for the impact of center of mass training augmentation as follows:

$$\log p^c_\theta(x) = \log p_\theta(x) + \frac{\|c\|^2}{2\sigma^2} - \log\left[\frac{\|c^2\|}{\sqrt{2}\sigma^3\Gamma\left(\frac{3}{2}\right)}\right], \quad (5)$$

where $\Gamma(\cdot)$ is the gamma function. We empirically analyze the impact of this adjustment in §F.3.

We next outline a proposition, and prove in §B.1, that demonstrates that reweighting using the adjusted proposal provably leads to better SNIS effective sample size (ESS).

Proposition 1. *Given an* $\mathrm{SE}(3)$*-invariant* $\mu_{target}(x)$*, consider the decomposition of a data point* $x \in \mathbb{R}^{n\times 3}$ *into its constituent mean-free component,* $\bar{x} \in \mathbb{R}^{(n-1)\times 3} \hookrightarrow \mathbb{R}^{n\times 3}$ *and center of mass* $c \in \mathbb{R}^3$*,* $x = \bar{x} + c$*, where* $c \sim \mathcal{N}(0, \sigma^2)$*. Now, assume both the proposal* $p_\theta(x)$ *and the adjusted proposal* $p^c_\theta(x)$ *factorize independently over the mean-free component and the center of mass. Then setting* $p^c_\theta(x) = p_\theta(\bar{x}) \cdot 1/\sigma\chi_3(\|c\|)$*, leads to the following inequality on the effective sample size in the limit of* $K \to \infty$*:*

$$ESS\left(\frac{\mu_{target}(x)}{p_\theta(x)}\right) < ESS\left(\frac{\mu_{target}(x)}{p^c_\theta(x)}\right). \quad (6)$$

3.2. Inference Time Scaling of Boltzmann Generators

Given a trained BG with proposal flow p_θ, the self-normalized importance sampling estimator suffers from a large variance of importance weights as the dimensionality and complexity of $\mu_{\text{target}}(x)$ grows in large molecular systems. We aim to address this bottleneck by proposing an inference time scaling algorithm that anneals samples $x^i \sim p_\theta(x)$ — and corresponding unnormalized importance weights $w(x^i)$ — in a continuous manner towards μ_{target}.

Improved sampling via annealing. We leverage a class of methods that fall under non-equilibrium sampling to improve the base proposal flow samples. One of the simplest instantiations of this idea is to use annealed Langevin dynamics with reweighting through a continuous-time variant

[2] ESS is defined as: $\text{ESS} = 1/\sum_i^K (\bar{w}(x^i))^2$.

of Annealed Importance Sampling (AIS) (Neal, 2001). Concretely, we consider the following SDE that drives proposal samples towards the target Boltzmann density:

$$dx_\tau = -\epsilon_\tau \nabla \mathcal{E}_\tau(x_\tau) d\tau + \sqrt{2\epsilon_\tau} dW_\tau, \tag{7}$$

where $\epsilon_\tau \geq 0$ is a time-dependent diffusion coefficient and W_τ is the standard Wiener process. We distinguish τ, from t used in the context of training p_θ, as the time variable that evolves initial proposal samples at $\tau = 0$ towards the target at $\tau = 1$. The energy interpolation $\mathcal{E}_t$ is a design choice, and we opt for a simple linear interpolant $\mathcal{E}_t = (1-\tau)\mathcal{E}_0 + \tau\mathcal{E}_1$, and set $\mathcal{E}_0(x) = -\log p_\theta(x)$. We highlight that unlike past work in pure sampling (Máté & Fleuret, 2023; Albergo & Vanden-Eijnden, 2025) which use the prior energy $\mathcal{E}_0(x) = -\log p_0(x)$, our design affords the significantly more informative proposal given by the pre-trained normalizing flow p_θ. As such, there is no need for *additional learning*, with the annealing process extending the inference capabilities of the Boltzmann generator $\mu_\theta(x)$.

To resample or compute observables with the transported samples, we use the well-known and celebrated *Jarzynski's equality*, that enables the calculation of equilibrium statistics from non-equilibrium processes. We recall the result, originally derived in Jarzynski (1997), and recently re-derived in continuous-time in the context of learned sampling algorithm by Vargas et al. (2024); Albergo & Vanden-Eijnden (2025), that describes the importance weight time evolution.

Proposition 2 (Albergo & Vanden-Eijnden (2025))**.** *Let (x_τ, w_τ) solve the coupled system of SDE / ODE*

$$\begin{aligned} dx_\tau &= -\epsilon_\tau \nabla \mathcal{E}_\tau(x_\tau) d\tau + \sqrt{2\epsilon_\tau} dW_\tau \\ d\log w_\tau &= -\partial_\tau \mathcal{E}_\tau(x_\tau) d\tau \quad \textit{with } x_0 \sim p_\theta, w_0 = 0 \end{aligned}$$

then for any test function $\phi : \mathbb{R}^d \to \mathbb{R}$ we have

$$\int_{\mathbb{R}^d} \phi(x) p_\tau(x) dx = \frac{\mathbb{E}[w_\tau \phi(x_\tau)]}{\mathbb{E}[w_\tau]} \tag{8}$$

and

$$\mathcal{Z}_\tau / \mathcal{Z}_1 = \mathbb{E}[e^{w_\tau}] \quad \textit{(Jarzynski's equality)} \tag{9}$$

The final samples $x_{\tau=1}$ are then reweighted with the importance weights $w_{\tau=1}$, themselves lower variance than SNIS in conventional BGs. It is crucial to highlight that the prior is not directly constituent of this annealing process, but instead the learned proposal $p_\theta(x_0)$ acts as the initial distribution. It is precisely this learned proposal density that $d\log w_\tau$ evolves during the annealing process. Alnealed importance sampling can be considered a special case of sequential Monte Carlo (SMC) (Doucet et al., 2001). In SMC, resampling can occur at arbitrary times τ, typically using an ESS threshold as in adaptive resampling. Intuitively by resampling during the annealing process SMC can avoid particle redundancy in which all but a few particles have negligible weight. We state the full SBG sampling algorithm with adaptive resampling in Algorithm 1; to recover the SBG AIS variant we simply set $\text{ESS}_{\text{threshold}} = -1.0$.

Algorithm 1 SBG Sampling

Require: # particles K, # annealed distributions N, Energy annealing schedule $\mathcal{E}_\tau(x_\tau)$

$x_0 \sim \mathcal{E}_0(x_0)$; $\quad \Delta \leftarrow 1/N$
for $i = 1$ to N **do**
  $x_{\tau+\Delta} \leftarrow x_\tau - \epsilon_\tau \nabla \mathcal{E}_\tau(x_\tau) d\tau + \sqrt{2\epsilon_\tau} dW_\tau$
  $\log w_{\tau+\Delta} \leftarrow \log w_\tau - \partial_\tau \mathcal{E}(x_\tau) d\tau$
  $\tau \leftarrow \tau + \Delta$
  if ESS $<$ $\text{ESS}_{\text{threshold}}$ **then**
    $x_\tau \leftarrow \text{RESAMPLE}(x_\tau, w_\tau)$
    $w_\tau \leftarrow 0$
  end if
end for

To simulate the Langevin SDE in Equation (7), and the corresponding importance weight evolution, we require the gradient of the energy interpolant:

$$\nabla \mathcal{E}_\tau(x_\tau) = (1-\tau)\nabla(-\log p_\theta(x_\tau)) + \tau \nabla \left(\frac{\mathcal{E}(x_\tau)}{k_B T} \right),$$

which requires efficient gradient computation through the log-likelihood estimation under the normalizing flow p_θ as given by Eq. 1. This presents the first point of distinction between finite flows and CNFs. The former class of flows trained using Eq. 1 gives fast exact likelihoods — especially for our scalable non-equivariant TarFlow model. In contrast, CNFs must simulate Eq. 4 and differentiate through an ODE solver to compute $\nabla \log p_\theta(x_\tau)$ for each step of the Langevin SDE in Eq. 7. As a result, a TarFlow proposal is considerably cheaper to simulate and reweight with AIS than a CNF. In §A we present an alternate interpolant that does not require the proposal distribution during sampling which is appealing when only samples are needed but at the cost of more expensive computation of log weights. These paths are of interest in the setting of Boltzmann emulators and other generative models, and are of independent interest, but are not considered further in the context of SBG.

For improved numerical stability during annealing, and to further reduce computational footprint, we propose a strategy that eliminates the forward evolution of the initial proposal that already obtain high energy. Specifically, we can simulate a large number of samples via Eq. 12 and threshold using an energy threshold $\gamma > 0$, and evaluate the log weights of promising samples. We justify our strategy by

first remarking a lower bound to the log partition function of μ_{target} using a Monte Carlo estimate,

$$\log \mathcal{Z} = \log \mathbb{E}_{x\sim p_\theta(x)}\left[\frac{\exp\left(\frac{-\mathcal{E}(x)}{k_B T}\right)}{p_\theta(x)}\right] \geq \mathbb{E}_{x\sim p_\theta(x)}\left[\frac{-\mathcal{E}(x)}{k_B T} - \log p_\theta(x)\right] = \log \hat{\mathcal{Z}}. \quad (10)$$

Plugging this estimate in the definition of the target Boltzmann distribution we get an upper bound,

$$\log \mu_{\text{target}}(x) \leq \log\left(\frac{-\mathcal{E}(x)}{k_B T}\right) - \log \hat{\mathcal{Z}}.$$

An upper bound on $\mu_{\text{target}}(x)$ allows us to threshold samples using the energy function, $\mathcal{E}(x) > \gamma$, of the target. Formally, this corresponds to truncating the target distribution $\hat{\mu}_{\text{target}}(x) := \mathbb{P}\left(\mu_{\text{target}}(x) \geq \frac{\gamma}{\log \hat{\mathcal{Z}}}\right)$ which places zero mass on high energy conformations. Correcting flow samples with respect to this truncated target introduces an additional bias into the self-normalized importance sampling estimate, which precisely corresponds to the difference in total variation distance between the two distributions $\text{TV}(\hat{\mu}_{\text{target}}, \mu_{\text{target}})$. We prove this result using an intermediate result in Lemma 1 included in §B.

Our next theoretical result provides a prescriptive strategy of setting an appropriate threshold γ as a function of the number of samples K and effective sample size under $\hat{\mu}_{\text{target}}(x)$.

Proposition 3. *Given an energy threshold $\mathcal{E}(x) > \gamma$, for $\gamma > 0$ large and the resulting truncated target distribution $\hat{\mu}_{target}(x) := \mathbb{P}\left(\mu_{target}(x) \geq \frac{\gamma}{\log \hat{\mathcal{Z}}}\right)$. Further, assume that the density of unnormalized importance weights w.r.t. to $\hat{\mu}_{target}$ is square integrable $(\hat{w}(x))^2 < \infty$. Given a tolerance $\rho = 1/ESS$ and bias of the original importance sampling estimator in total variation $b = TV(\mu_\theta, \mu_{target})$, then the γ-truncation threshold with K-samples for $TV(\mu_\theta, \hat{\mu}_{target})$ is:*

$$\gamma \geq \frac{1}{\lambda}\log\left(\frac{Kb}{12\rho\mathbb{E}[\exp(-\lambda X)]}\right) + \log \hat{\mathcal{Z}}. \quad (11)$$

The proof for Proposition 3 is located in §B.3. Proposition 3 allows us to appropriately set a energy threshold γ as a function of tolerance ρ that depends on ESS. In practice, this allows us to negotiate the amount of acceptable bias when dropping initial samples that obtain high-energy before any further AIS correction. Moreover, this gives a firmer theoretical foundation to existing practices of thresholding high importance weight samples (Midgley et al., 2023b;a).

Analogous to thresholding based on $\mathcal{E}(x)$, we can also threshold by the probability under the proposal flow with truncation $\hat{p}_\theta(x) := \mathbb{P}(p_\theta(x) \geq \delta)$, for small $\delta > 0$. Essentially, this thresholding filters low probability samples under the model prior to any importance sampling. The additional bias incurred by performing such thresholding is theoretically analyzed in Proposition 4 and presented in §B.4.

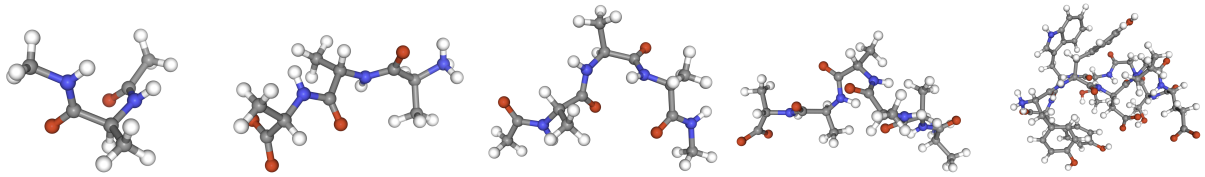
4. Experiments

We evaluate SBG on small peptides using classical forcefield energy functions, further experimental details are described in §E. SBG samples are generated by Algorithm 1, both with adaptive resampling (SMC) and without (AIS).

Datasets. We consider small peptides composed of up to 6 alanine residues, with some systems additionally incorporating an acetyl group and an N-methyl group. All datasets are generated from a single MD simulation in implicit solvent using a classical force field. For each system, the first 1 µs is used for training, the next 0.2 µs for validation, and the remainder serves as the test set. Therefore, some metastable states may not be represented in the training set. An exception is alanine dipeptide, for which we use the dataset from Klein & Noé (2024). In addition to the alanine systems, we also investigate the 138-atom peptide *chignolin*, consisting of 10 residues (GYDPETGTWG) and notable for it's formation of β-hairpin structure in water solvent Honda et al. (2004). We provide additional dataset details in §D.

Baselines. For baselines, we train prior state-of-the-art equivariant Boltzmann generators. Specifically, we train the exactly invertible and equivariant SE(3)-augmented coupling flow (Midgley et al., 2023a), and the equivariant continuous normalizing flow (ECNF) employed in Transferable Boltzmann Generators (Klein & Noé, 2024). We also include an improved variant, denoted ECNF++, as a stronger baselines; this uses a refined flow matching objective, larger network, and improved optimization hyperparameters, full details provided in §E.4 for full details. We note that both SE(3)-EACH and ECNF to be equivariant to E(3) and hence generate samples of both global chiralities, which we resolve by applying a flip transformation as in Klein & Noé (2024), for further details and related results see §F.

Metrics. We report effective sample size (ESS) along with Wasserstein-2 distances on the energy distribution $\mathcal{E}$-$\mathcal{W}_2$ and dihedral angle torus $\mathbb{T}$-$\mathcal{W}_2$. The energy distribution is highly sensitive to fine-grained details whereas the dihedral angles encode macrostructural information such as metastable state occupancy; full metric definitions are provided in §E. Additional results for Wasserstein-2 distance on time-lagged independent component analysis (TICA) projections TICA-$\mathcal{W}_2$ are provided in §F. We provide energy histograms in the main text whilst Ramachandran plots (Ramachandran et al., 1963) detailing the mode coverage via dihedral angle distributions are presented in §F.

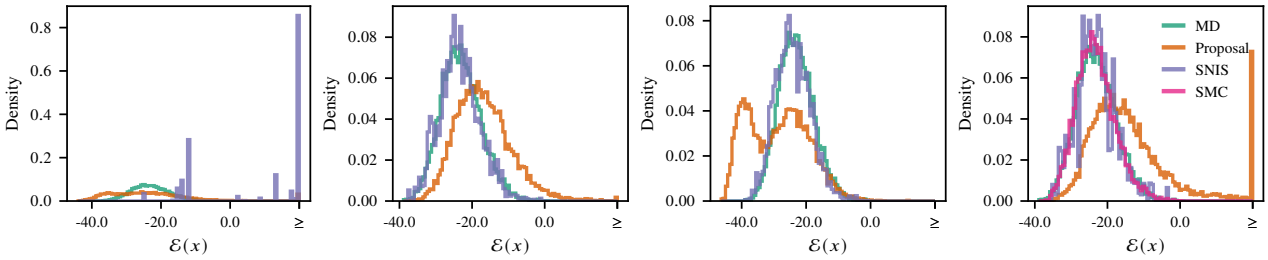


(a) Alanine dipeptide (b) Trialanine (c) Alanine tetrapeptide (d) Hexa-alanine (e) Chignolin

Figure 2. Samples generated by SBG on peptide systems ranging from 2 to 10 residues.

Table 2. Quantitative results on alanine dipeptide and trialanine. Baseline methods presented with SNIS.

	Alanine dipeptide			Trialanine		
Algorithm ↓	ESS ↑	$\mathcal{E}$-$\mathcal{W}_2$ ↓	$\mathbb{T}$-$\mathcal{W}_2$ ↓	ESS ↑	$\mathcal{E}$-$\mathcal{W}_2$ ↓	$\mathbb{T}$-$\mathcal{W}_2$ ↓
SE(3)-EACF	$< 10^{-3}$	108.202	2.867	—	—	—
ECNF	0.119	0.419	0.311	—	—	—
ECNF ++ (Ours)	**0.275 ± 0.010**	0.914 ± 0.122	**0.189 ± 0.019**	0.003 ± 0.002	2.206 ± 0.813	0.962 ± 0.253
SBG AIS (Ours)	0.030 ± 0.012	0.630 ± 0.249	0.418 ± 0.090	**0.052 ± 0.013**	0.797 ± 0.094	**0.450 ± 0.043**
SBG SMC (Ours)	—	**0.412 ± 0.125**	0.430 ± 0.100	—	**0.590 ± 0.267**	0.455 ± 0.076



(a) SE(3)-EACF (b) ECNF (c) ECNF++ (d) SBG

Figure 3. Energy histograms for baseline methods and SBG on alanine dipeptide dataset.

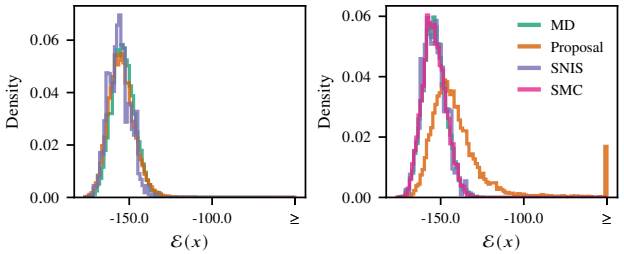


Figure 4. Energy distribution histograms for baseline ECNF++ **(left)** and SBG **(right)** on trialanine dataset.

4.1. Results

We evaluate SBG and our baseline methods with quantitative metrics summarized in Table 2 and Table 3. Where ± is present three models are independently trained and sampled; unless otherwise stated 10^4 particles are sampled. We provide examples of SBG generated samples in Figure 2.

Alanine dipeptide. SE(3)-EACF was originally trained on an alanine dipeptide dataset at 800 K; we retrain on our more challenging 300 K data using the original codebase of Midgley et al. (2023a). Despite the proposal distribution having good overlap with the MD data, we find the SNIS reweighted performance to be substantially degraded at this lower temperature when using the same 0.2% weight clipping threshold as the original work; see §F for analysis of more aggressive clipping thresholds. The ESS and $\mathbb{T}$-$\mathcal{W}_2$ of ECNF++ outperform both SBG variants by a large margin, although the inverse is true for the $\mathcal{E}$-$\mathcal{W}_2$. Furthermore, the original ECNF model trained by Klein & Noé (2024) achieves superior $\mathcal{E}$-$\mathcal{W}_2$ to our proposed ECNF++ but inferior ESS and $\mathbb{T}$-$\mathcal{W}_2$. These results are further substantiated by the energy distribution histograms in Figure 3.

Trialanine. Despite achieving acceptable performance on alanine dipeptide, ECNF was unable to scale to trialanine and was omitted. The computational cost of SE(3)-EACF (c.f. Table 5) precluded it's consideration. The SBG variants are significantly stronger in all metrics compared to ECNF++, with SBG AIS achieving higher ESS and both AIS and SMC outperforming on $\mathcal{E}$-$\mathcal{W}_2$ and $\mathbb{T}$-$\mathcal{W}_2$. However the performance of ECNF++ is acceptable, and constitutes the first CNF-based Boltzmann generator on a tripeptide system in Cartesian coordinates. There is no notable distinction in performance between SBG variants.

Table 3. Quantitative results on alanine tetrapeptide and hexa-alanine. ECNF++ presented with SNIS.

Datasets →	Alanine tetrapeptide			Hexa-alanine		
Algorithm ↓	ESS ↑	$\mathcal{E}$-$\mathcal{W}_2$ ↓	$\mathbb{T}$-$\mathcal{W}_2$ ↓	ESS ↑	$\mathcal{E}$-$\mathcal{W}_2$ ↓	$\mathbb{T}$-$\mathcal{W}_2$ ↓
ENCF++ (Ours)	0.016 ± 0.001	5.638 ± 0.483	1.002 ± 0.061	0.006 ± 0.001	10.668 ± 0.285	1.902 ± 0.055
SBG AIS (Ours)	$\mathbf{0.046 \pm 0.014}$	$\mathbf{0.883 \pm 0.213}$	$\mathbf{0.866 \pm 0.076}$	$\mathbf{0.034 \pm 0.015}$	$\mathbf{1.021 \pm 0.239}$	$\mathbf{1.431 \pm 0.085}$
SBG SMC (Ours)	—	1.027 ± 0.465	0.888 ± 0.114	—	1.189 ± 0.357	1.444 ± 0.140

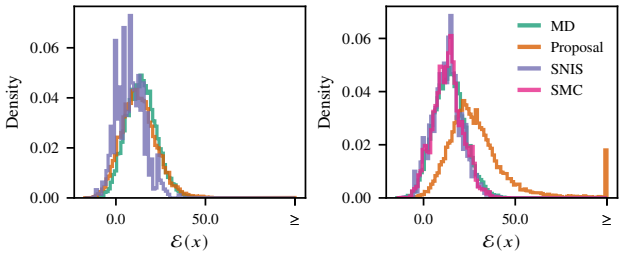


Figure 5. Energy distribution histograms for baseline ECNF++ **(left)** and SBG **(right)** on alanine tetrapeptide dataset.

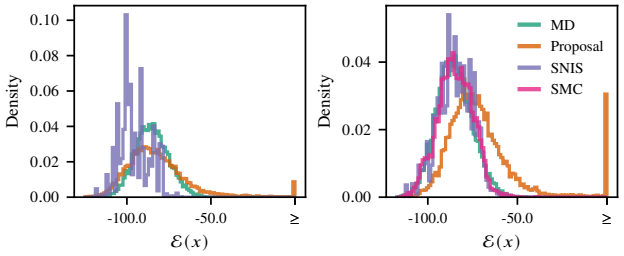


Figure 6. Energy distribution histograms for baseline ECNF++ **(left)** and SBG **(right)** on hexa-alanine dataset.

Alanine tetrapeptide and hexa-alanine. At this scale the ECNF++ baseline diverges from the target distribution, reflected particularly in $\mathcal{E}$-$\mathcal{W}_2$. In contrast, SBG is readily scalable up to hexapeptides, achieving greatly reduced $\mathcal{E}$-$\mathcal{W}_2$ on both datasets. As reweighted samples under SBG show extremely high overlap with the ground truth $\mu_{\text{target}}(x)$, we argue that SBG successfully solves these molecular systems in comparison to prior BGs. This conclusion is supported by the energy histograms in Figure 5 and Figure 6, in which the SNIS reweighted SNIS does not approximate the MD data well, in contrast to the good alignment of SBG. Notably, the proposals for ECNF++ have good overlap with the target density, indicating the error to be introduced by the likelihood estimation itself.

Inference scaling. To illustrate the scalability of SBG in relation to other methods we plot in Figure 7 the GPU hours required by each method to sample 10^4 points. We observe exponential scaling of inference time for ECNF++ as the size of the system grows, whilst SBG is less sensitive to system size and over an order of magnitude faster on the hexapeptide system. We additionally plot the $\mathbb{T}$-$\mathcal{W}_2$ on

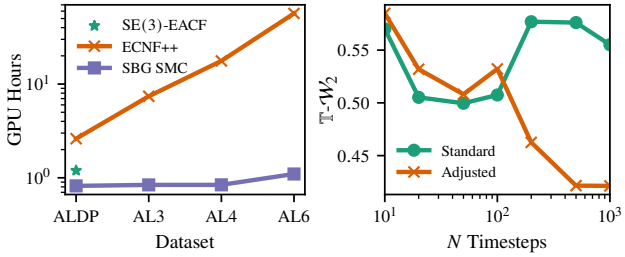


Figure 7. **Left:** GPU hours (NVIDIA L40S) for sampling and reweighting 10^4 points. **Right:** $\mathbb{T}$-$\mathcal{W}_2$ on trialanine as a function of Langevin timestep discretization for both standard $p_\theta(x)$ and center of mass adjusted proposal energy functions $p^c_\theta(x)$.

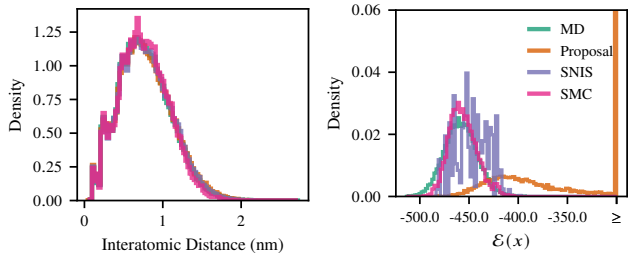


Figure 8. SBG interatomic distance histogram **(left)** and energy distribution histogram **(right)** for decapeptide chignolin (GYDPETGTWG) . SNIS $\mathcal{E}$-$\mathcal{W}_2 = 12.046$, SMC $\mathcal{E}$-$\mathcal{W}_2 = 3.571$.

trialanine as a function of Langevin timestep granularity for SBG SMC both with and without the center of mass proposal energy adjustment, as stated Equation (5). When the center of mass adjusted energy is employed we observe a strong inverse relationship between time discretization steps and $\mathbb{T}$-$\mathcal{W}_2$, however without this adjustment there is no clear relationship. This evidences both the efficacy of the center of mass adjustment at improving reweighting as well as the potential of SBG for inference-time scaling — a capability not present in the standard Boltzmann generator.

4.2. Scaling to Decapeptide

We now apply SBG to the decapeptide chignolin. As no other method can scale to this system we report energy histograms and distance plots for SBG SMC only in Figure 8. We observe success of SBG at matching the interatomic distance distribution. We additionally observe a strong overlap of SMC sample energy distribution despite the notably poor

proposal overlap, providing further demonstration of the viability of the SBG approach to molecular system sampling. Our application of SBG to chignolin represents a significant step forwards in the scalability of BGs, where prior methods struggled on even alanine tetrapeptide, as observable in results for ECNF++ $\mathcal{E}$-$\mathcal{W}_2$ presented in Table 3.

5. Related Work

Boltzmann generators (BGs) (Noé et al., 2019) have been applied to both free energy estimation (Wirnsberger et al., 2020; Rizzi et al., 2023; Schebek et al., 2024) and molecular sampling. Initially, BGs relied on system-specific representations, such as internal coordinates, to achieve relevant sampling efficiencies (Noé et al., 2019; Köhler et al., 2021; Midgley et al., 2023b; Köhler et al., 2023; Dibak et al., 2022). However, these representations are generally not transferable across different systems, leading to the development of BGs operating in Cartesian coordinates (Klein et al., 2023b; Midgley et al., 2023a; Klein & Noé, 2024). While this improves transferability, they are currently limited in scalability, struggling to extend beyond dipeptides. Scaling to larger systems typically requires sacrificing exact sampling from the target distribution (Jing et al., 2022; Abdin & Kim, 2023; Jing et al., 2024a; Lewis et al., 2024). An alternative to direct sampling from $\mu_{\text{target}}(x)$ is to generate samples iteratively by learning large steps in time (Schreiner et al., 2023; Fu et al., 2023; Klein et al., 2023a; Diez et al., 2025; Jing et al., 2024b; Daigavane et al., 2024) to accelerate methods such as molecular dynamics via coarse-graining.

Amortized sampling. The field of sampling has seen renewed interest with the rise of generative models. In particular, the use of diffusion-based samplers has seen rapid application with a plethora of approaches exploiting the favorable theoretical properties of mode-mixing of diffusion models (Berner et al., 2024; Vargas et al., 2023; Richter et al., 2024; Zhang & Chen, 2022; Vargas et al., 2024). While initial approaches focused on simulation-based dynamics, including both overdamped and underdamped Langevin (Blessing et al., 2025; Chen et al., 2025), it is expected that simulation-free methods that also exploit diffusion properties (Akhound-Sadegh et al., 2024; Huang et al., 2021; De Bortoli et al., 2024) are an attractive opportunity to tackle larger-scale systems due to their scalability. Finally, flow-based models have also been employed for sampling with classical flows augmenting MCMC (Arbel et al., 2021; Gabrié et al., 2021; Matthews et al., 2022; Midgley et al., 2023b; Hagemann et al., 2023), and through CNFs that construct ODE bridges, such as linear interpolants between the prior and target (Máté & Fleuret, 2023), and more general bridges that rely on satisfying the mass transport equations (Tian et al., 2024; Fan et al., 2024).

6. Conclusion

In this paper, we introduce SBG an extension to the Boltzmann generator framework that scales inference through the use of annealing processes. Unlike past BGs, in SBG, we scale training using a non-equivariant transformer-based TarFlow architecture with soft equivariance penalties to 6 peptides. In terms of limitations, using non-equilibrium sampling as presented in SBG does not enjoy easy application to CNFs due to expensive simulation, which limits the use of modern flow matching methods in a SBG context. Considering hybrid approaches that mix CNFs through distillation to an invertible architecture or consistency-based objectives is thus a natural direction for future work. Finally, considering other classes of scalable generative models such as autoregressive ones which also permit exact likelihoods is also a ripe direction for future work.

Acknowledgements

The authors thank Damien Ferbach, Tara Akhound-Sadegh, Lars Holdijk, Kacper Kapuśniak, Kirill Neklyudov, Michael Albergo, and Majdi Hassan for insightful conversations and feedback. In addition, the authors thank Paul Skaluba for constructive comments on Proposition 1 of an older draft.

The authors acknowledge funding from UNIQUE, CIFAR, NSERC, Intel, and Samsung. The research was enabled in part by computational resources provided by the Digital Research Alliance of Canada (https://alliancecan.ca), Mila (https://mila.quebec), and NVIDIA. AJB is partially supported by an NSERC Post-doc fellowship. This research is partially supported by the EPSRC Turing AI World-Leading Research Fellowship No. EP/X040062/1 and EPSRC AI Hub No. EP/Y028872/1.

Impact Statement

This work studies sampling from Boltzmann densities, a problem of general interest in machine learning and AI4Science that arises both in pure statistical modeling and within applications. We highlight the training Boltzmann generators on molecular tasks are in turn applicable to drug and material discovery. While we do not foresee immediate negative impacts of our advances in this area, we encourage due caution whilst scaling to prevent their potential misuse.

References

Abdin, O. and Kim, P. M. Pepflow: direct conformational sampling from peptide energy landscapes through hypernetwork-conditioned diffusion. *bioRxiv*, pp. 2023–06, 2023. (Cited on page 9)

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pp. 1–3, 2024. (Cited on page 2)

Agapiou, S., Papaspiliopoulos, O., Sanz-Alonso, D., and Stuart, A. M. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, pp. 405–431, 2017. (Cited on page 16)

Akhound-Sadegh, T., Rector-Brooks, J., Bose, J., Mittal, S., Lemos, P., Liu, C.-H., Sendera, M., Ravanbakhsh, S., Gidel, G., Bengio, Y., Malkin, N., and Tong, A. Iterated denoising energy matching for sampling from boltzmann densities. In *International Conference on Machine Learning (ICML)*, 2024. (Cited on pages 2 and 9)

Albergo, M. S. and Vanden-Eijnden, E. Building normalizing flows with stochastic interpolants. *International Conference on Learning Representations (ICLR)*, 2023. (Cited on page 3)

Albergo, M. S. and Vanden-Eijnden, E. Nets: A non-equilibrium transport sampler. In *International Conference on Machine Learning (ICML)*, 2025. (Cited on pages 2, 5, and 14)

Alexey, D. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. (Cited on page 4)

Arbel, M., Matthews, A., and Doucet, A. Annealed flow transport monte carlo. In *International Conference on Machine Learning*, pp. 318–330. PMLR, 2021. (Cited on page 9)

Berner, J., Richter, L., and Ullrich, K. An optimal control perspective on diffusion-based generative modeling. *Transactions on Machine Learning Research (TMLR)*, 2024. (Cited on page 9)

Blessing, D., Berner, J., Richter, L., and Neumann, G. Underdamped diffusion bridges with applications to sampling. In *International Conference on Learning Representations (ICLR)*, 2025. (Cited on page 9)

Bose, A. J., Brubaker, M., and Kobyzev, I. Equivariant finite normalizing flows. *arXiv*, 2021. (Cited on pages 2 and 4)

Buch, I., Giorgino, T., and De Fabritiis, G. Complete reconstruction of an enzyme-inhibitor binding process by molecular dynamics simulations. *Proceedings of the National Academy of Sciences*, 108(25):10184–10189, 2011. (Cited on page 1)

Chen, J., Richter, L., Berner, J., Blessing, D., Neumann, G., and Anandkumar, A. Sequential controlled langevin diffusions. In *International Conference on Learning Representations (ICLR)*, 2025. (Cited on page 9)

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. *Neural Information Processing Systems (NIPS)*, 2018. (Cited on pages 2 and 3)

Daigavane, A., Vani, B. P., Saremi, S., Kleinhenz, J., and Rackers, J. Jamun: Transferable molecular conformational ensemble generation with walk-jump sampling. *arXiv*, 2024. (Cited on page 9)

De Bortoli, V., Hutchinson, M., Wirnsberger, P., and Doucet, A. Target score matching. *arXiv*, 2024. (Cited on page 9)

Deringer, V. L. Modelling and understanding battery materials with machine-learning-driven atomistic simulations. *Journal of Physics: Energy*, 2(4):041003, oct 2020. doi: 10.1088/2515-7655/abb011. (Cited on page 1)

Dibak, M., Klein, L., Krämer, A., and Noé, F. Temperature steerable flows and Boltzmann generators. *Phys. Rev. Res.*, 4:L042005, Oct 2022. doi: 10.1103/PhysRevResearch.4.L042005. (Cited on pages 9 and 21)

Diez, J. V., Schreiner, M., Engkvist, O., and Olsson, S. Boltzmann priors for implicit transfer operators. *International Conference on Learning Representations (ICLR)*, 2025. (Cited on page 9)

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using Real NVP. *International Conference on Learning Representations (ICLR)*, 2017. (Cited on pages 1 and 3)

Domingo-Enrich, C., Drozdzal, M., Karrer, B., and Chen, R. T. Q. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Representation Learning (ICLR)*, 2025. (Cited on pages 18 and 19)

Doucet, A., De Freitas, N., Gordon, N. J., et al. *Sequential Monte Carlo methods in practice*, volume 1. 2001. (Cited on pages 2, 4, and 5)

Eastman, P., Swails, J., Chodera, J. D., McGibbon, R. T., Zhao, Y., Beauchamp, K. A., Wang, L.-P., Simmonett, A. C., Harrigan, M. P., Stern, C. D., et al. Openmm 7: Rapid development of high performance algorithms for

molecular dynamics. *PLoS computational biology*, 13(7): e1005659, 2017. (Cited on page 21)

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Lacey, K., Goodwin, A., Marek, Y., and Rombach, R. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024. (Cited on page 25)

Fan, M., Zhou, R., Tian, C., and Qian, X. Path-guided particle-based sampling. In *International Conference on Machine Learning (ICML)*, 2024. (Cited on page 9)

Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N. T., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., Tavenard, R., Tong, A., and Vayer, T. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. (Cited on page 23)

Fu, X., Xie, T., Rebello, N. J., Olsen, B., and Jaakkola, T. S. Simulate time-integrated coarse-grained molecular dynamics with multi-scale graph networks. *Transactions on Machine Learning Research*, 2023. (Cited on page 9)

Gabrié, M., Rotskoff, G. M., and Vanden-Eijnden, E. Efficient Bayesian sampling using normalizing flows to assist Markov chain Monte Carlo methods. *arXiv*, 2021. (Cited on page 9)

Garcia Satorras, V., Hoogeboom, E., Fuchs, F., Posner, I., and Welling, M. E(n) equivariant normalizing flows. *Neural Information Processing Systems (NeurIPS)*, 2021. (Cited on page 4)

Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., and Duvenaud, D. FFJORD: free-form continuous dynamics for scalable reversible generative models. In *International Conference on Representation Learning (ICLR)*, 2019. (Cited on page 25)

Hagemann, P. L., Hertrich, J., and Steidl, G. *Generalized normalizing flows via Markov chains*. 2023. (Cited on page 9)

Honda, S., Yamasaki, K., Sawada, Y., and Morii, H. 10 residue folded peptide designed by segment statistics. *Structure*, 12(8):1507–1518, 2004. (Cited on page 6)

Huang, J., Jiao, Y., Kang, L., Liao, X., Liu, J., and Liu, Y. Schrödinger-Föllmer sampler: sampling without ergodicity. *arXiv*, 2021. (Cited on page 9)

Hutchinson, M. 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. (Cited on page 25)

Jarzynski, C. Nonequilibrium equality for free energy differences. *Phys. Rev. Lett.*, 78:2690–2693, Apr 1997. doi: 10.1103/PhysRevLett.78.2690. (Cited on page 5)

Jing, B., Corso, G., Chang, J., Barzilay, R., and Jaakkola, T. Torsional diffusion for molecular conformer generation. *Advances in Neural Information Processing Systems*, 35: 24240–24253, 2022. (Cited on page 9)

Jing, B., Berger, B., and Jaakkola, T. Alphafold meets flow matching for generating protein ensembles. In *International Conference on Machine Learning (ICML)*, 2024a. (Cited on page 9)

Jing, B., Stärk, H., Jaakkola, T., and Berger, B. Generative modeling of molecular dynamics trajectories. In *Neural Information Processing Systems (NeurIPS)*, 2024b. (Cited on page 9)

Karczewski, R., Heinonen, M., and Garg, V. Diffusion models as cartoonists! the curious case of high density regions. In *International Conference on Learning Representations (ICLR)*, 2024. (Cited on pages 14, 19, and 20)

Klein, L. and Noé, F. Transferable boltzmann generators. In *Advances in Neural Information Processing Systems*, 2024. (Cited on pages 6, 7, 9, 21, 22, 25, 26, 27, and 29)

Klein, L., Foong, A. Y., Fjelde, T. E., Mlodozeniec, B., Brockschmidt, M., Nowozin, S., Noé, F., and Tomioka, R. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics. *Neural Information Processing Systems (NeurIPS)*, 2023a. (Cited on pages 9 and 21)

Klein, L., Krämer, A., and Noé, F. Equivariant flow matching. *Neural Information Processing Systems (NeurIPS)*, 2023b. (Cited on pages 2 and 9)

Köhler, J., Klein, L., and Noé, F. Equivariant flows: exact likelihood generative learning for symmetric densities. *International Conference on Machine Learning (ICML)*, 2020. (Cited on page 4)

Köhler, J., Krämer, A., and Noé, F. Smooth normalizing flows. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 2796–2809, 2021. (Cited on page 9)

Köhler, J., Invernizzi, M., De Haan, P., and Noé, F. Rigid body flows for sampling molecular crystal structures. *International Conference on Machine Learning (ICML)*, 2023. (Cited on page 9)

Lee, H., Pabbaraju, C., Sevekari, A. P., and Risteski, A. Universal approximation using well-conditioned normalizing flows. *Advances in Neural Information Processing Systems*, 34:12700–12711, 2021. (Cited on page 4)

Lewis, S., Hempel, T., Jiménez Luna, J., Gastegger, M., Xie, Y., Foong, A. Y., García Satorras, V., Abdin, O., Veeling, B. S., Zaporozhets, I., et al. Scalable emulation of protein equilibrium ensembles with generative deep learning. *bioRxiv*, pp. 2024–12, 2024. (Cited on page 9)

Lindorff-Larsen, K., Piana, S., Dror, R. O., and Shaw, D. E. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011. (Cited on page 1)

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *International Conference on Learning Representations (ICLR)*, 2023. (Cited on page 3)

Liu, Q. Rectified flow: A marginal preserving approach to optimal transport. *arXiv*, 2022. (Cited on page 3)

Liu, X., Zhang, X., Ma, J., Peng, J., and Liu, Q. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation, 2024. (Cited on page 25)

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2017. (Cited on page 25)

Máté, B. and Fleuret, F. Learning interpolations between boltzmann densities. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. (Cited on pages 5 and 9)

Matsumoto, M., Saito, S., and Ohmine, I. Molecular dynamics simulation of the ice nucleation and growth process leading to water freezing. *Nature*, 416(6879):409–413, 2002. (Cited on page 1)

Matthews, A., Arbel, M., Rezende, D. J., and Doucet, A. Continual repeated annealed flow transport monte carlo. *International Conference on Machine Learning (ICML)*, 2022. (Cited on page 9)

Midgley, L. I., Stimper, V., Antorán, J., Mathieu, E., Schölkopf, B., and Hernández-Lobato, J. M. SE(3) equivariant augmented coupling flows. *Neural Information Processing Systems (NeurIPS)*, 2023a. (Cited on pages 2, 6, 7, 9, and 24)

Midgley, L. I., Stimper, V., Simm, G. N., Schölkopf, B., and Hernández-Lobato, J. M. Flow annealed importance sampling bootstrap. *International Conference on Learning Representations (ICLR)*, 2023b. (Cited on pages 6 and 9)

Neal, R. M. Annealed importance sampling. *Statistics and computing*, 11:125–139, 2001. (Cited on pages 2 and 5)

Noé, F., Schütte, C., Vanden-Eijnden, E., Reich, L., and Weikl, T. R. Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations. *Proceedings of the National Academy of Sciences*, 106 (45):19011–19016, 2009. (Cited on page 1)

Noé, F., Olsson, S., Köhler, J., and Wu, H. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019. (Cited on pages 1, 2, 3, and 9)

Owen, A. B. *Monte Carlo theory, methods and examples*. 2013. (Cited on page 4)

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017. (Cited on page 4)

Parrinello, M. and Rahman, A. Crystal structure and pair potentials: A molecular-dynamics study. *Physical review letters*, 45(14):1196, 1980. (Cited on page 1)

Ramachandran, G. N., Ramakrishnan, C., and Sasisekharan, V. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, pp. 95–99, 1963. (Cited on page 6)

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. *International Conference on Machine Learning (ICML)*, 2015. (Cited on pages 1 and 3)

Richter, L., Berner, J., and Liu, G.-H. Improved sampling via learned diffusions. *International Conference on Learning Representations (ICLR)*, 2024. (Cited on page 9)

Rizzi, A., Carloni, P., and Parrinello, M. Free energies at qm accuracy from force fields via multimap targeted estimation. *Proceedings of the National Academy of Sciences*, 2023. (Cited on page 9)

Satorras, V. G., Hoogeboom, E., and Welling, M. E (n) equivariant graph neural networks. *International Conference on Machine Learning (ICML)*, 2021. (Cited on page 2)

Schebek, M., Invernizzi, M., Noé, F., and Rogal, J. Efficient mapping of phase diagrams with conditional boltzmann generators. *Machine Learning: Science and Technology*, 2024. (Cited on page 9)

Schreiner, M., Winther, O., and Olsson, S. Implicit transfer operator learning: Multiple time-resolution models for molecular dynamics. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. (Cited on page 9)

Skreta, M., Atanackovic, L., Bose, A. J., Tong, A., and Neklyudov, K. The superposition of diffusion models using the itô density estimator. In *International Conference on Learning Representations (ICLR)*, 2025. (Cited on pages 14 and 19)

Teshima, T., Ishikawa, I., Tojo, K., Oono, K., Ikeda, M., and Sugiyama, M. Coupling-based invertible neural networks are universal diffeomorphism approximators. *Advances in Neural Information Processing Systems*, 33:3362–3373, 2020. (Cited on page 4)

Tian, Y., Panda, N., and Lin, Y. T. Liouville flow importance sampler. In *International Conference on Machine Learning (ICML)*, 2024. (Cited on page 9)

Tong, A., Fatras, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., and Bengio, Y. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. (Cited on pages 3 and 25)

Vargas, F., Grathwohl, W., and Doucet, A. Denoising diffusion samplers. *International Conference on Learning Representations (ICLR)*, 2023. (Cited on page 9)

Vargas, F., Padhy, S., Blessing, D., and Nüsken, N. Transport meets variational inference: Controlled Monte Carlo diffusions. *International Conference on Learning Representations (ICLR)*, 2024. (Cited on pages 5 and 9)

Wirnsberger, P., Ballard, A. J., Papamakarios, G., Abercrombie, S., Racanière, S., Pritzel, A., Jimenez Rezende, D., and Blundell, C. Targeted free energy estimation via learned mappings. *J. Chem. Phys.*, 2020. (Cited on pages 1 and 9)

Zhai, S., Zhang, R., Nakkiran, P., Berthelot, D., Gu, J., Zheng, H., Chen, T., Bautista, M. A., Jaitly, N., and Susskind, J. Normalizing flows are capable generative models. *arXiv*, 2024. (Cited on pages 2, 4, and 26)

Zhang, Q. and Chen, Y. Path integral sampler: a stochastic control approach for sampling. *International Conference on Learning Representations (ICLR)*, 2022. (Cited on page 9)

A. Alternate Paths

A.1. Proposal Free Langevin Dynamics

We can also modify the Langevin SDE in Eq. 7 to include an additional drift term $\nu_\tau(x_\tau) \in \mathbb{R}^d$ as follows:

$$dx_\tau = -\epsilon_\tau \nabla \mathcal{E}_t(x_\tau) d\tau + \nu_\tau(x_\tau) d\tau + \sqrt{2\epsilon_\tau} dW_\tau.$$

Under perfect drift $\nu_\tau(\tau)$ the log weights do not change and there is no need for correction. For imperfect drift the corresponding coupled ODE time-evolution of log-weights $d \log w_\tau$ needed to apply AIS was derived in NETS (Albergo & Vanden-Eijnden, 2025, Proposition 3):

$$dw_\tau = \nabla \cdot \nu_\tau(x_\tau) d\tau - \nabla \mathcal{E}_\tau(x_\tau) \cdot \nu_\tau(x_\tau) d\tau - \partial_\tau \mathcal{E}_\tau(x_\tau) d\tau.$$

In contrast to learning a drift as done in NETS (Albergo & Vanden-Eijnden, 2025) we now illustrate that a judicious choice of $\nu_\tau(x_\tau)$ eliminates the need to compute the gradient of log-likelihood under the proposal. For instance, we can choose $\nu_\tau(x_\tau) = \epsilon_\tau \nabla \mathcal{E}_\tau(x_\tau) - \epsilon_\tau \nabla \left(\frac{\mathcal{E}(x_\tau)}{k_B T} \right)$, which by straightforward calculation gives the following SDE:

$$\begin{aligned} dx_\tau &= -\epsilon_\tau \nabla \mathcal{E}_t(x_\tau) d\tau + \nu_\tau(x_\tau) d\tau + \sqrt{2\epsilon_\tau} dW_\tau \\ &= -\epsilon_\tau \nabla \left(\frac{\mathcal{E}(x_\tau)}{k_B T} \right) d\tau + \sqrt{2\epsilon_\tau} dW_\tau. \end{aligned} \tag{12}$$

This new SDE greatly simplifies the simulation of samples x_τ as it is independent of the proposal energy $\nabla \mathcal{E}_0(x_\tau) = -\nabla \log p_\theta(x_\tau)$. However, the log weights ODE still requires the computation of the gradient of the proposal energy. The form of Eq. 12 suggests the possibility of massively parallel simulation schemes under a regular normalizing flow and a CNF. However, due to simulatio the log weights remains expensive for CNFs due to the need to compute the divergence operator in Eq. 4. Furthermore, while recent advances in divergence-free density estimation via the Itô density estimator (Skreta et al., 2025; Karczewski et al., 2024) might appear attractive we show that the log density under this estimator is necessarily biased and may limit the fidelity of self-normalized importance sampling incurs non-negotiable added bias. For ease of presentation, we present this theoretical investigation in §C.2 and characterize the added bias in Proposition 5. In totality, this limits the application of continuous BG's to only the conventional IS setting, unlike finite flows like TarFlow which can benefit from non-equilibrium transport and AIS.

B. Proofs

B.1. Proof of Proposition 1

Proposition 1. *Given an* $\mathrm{SE}(3)$*-invariant* $\mu_{target}(x)$*, consider the decomposition of a data point* $x \in \mathbb{R}^{n\times 3}$ *into its constituent mean-free component,* $\bar{x} \in \mathbb{R}^{(n-1)\times 3} \hookrightarrow \mathbb{R}^{n\times 3}$ *and center of mass* $c \in \mathbb{R}^3$*,* $x = \bar{x} + c$*, where* $c \sim \mathcal{N}(0, \sigma^2)$*. Now, assume both the proposal* $p_\theta(x)$ *and the adjusted proposal* $p^c_\theta(x)$ *factorize independently over the mean-free component and the center of mass. Then setting* $p^c_\theta(x) = p_\theta(\bar{x}) \cdot 1/\sigma\chi_3(\|c\|)$*, leads to the following inequality on the effective sample size in the limit of* $K \to \infty$*:*

$$\mathit{ESS}\left(\frac{\mu_{\mathit{target}}(x)}{p_\theta(x)} \right) < \mathit{ESS}\left(\frac{\mu_{\mathit{target}}(x)}{p^c_\theta(x)} \right). \tag{6}$$

Proof. Recall the definition of effective sample size using Kish's formula, both $p_\theta(x)$ and the adjusted proposal $p^c_\theta(x)$:

$$\begin{aligned} \mathrm{ESS}\left(\frac{\mu_{\mathrm{target}(x)}}{p_\theta(x)} \right) &= \frac{1}{\sum_i^K (\bar{w}(x^i))^2} = \frac{\left(\sum_i^K w(x^i)\right)^2}{\sum_i^K w(x^i)^2} = \frac{\left(\sum_i^K \mu_{\mathrm{target}}(x^i)/p_\theta(x^i)\right)^2}{\sum_i^K w(x^i)^2} \\ \mathrm{ESS}\left(\frac{\mu_{\mathrm{target}(x)}}{p^c_\theta(x)} \right) &= \frac{1}{\sum_i^K (\bar{w}^c(x^i))^2} = \frac{\left(\sum_i^K w^c(x^i)\right)^2}{\sum_i^K w^c(x^i)^2} = \frac{\left(\sum_i^K \mu_{\mathrm{target}}(x^i)/p^c_\theta(x^i)\right)^2}{\sum_i^K \left(\mu_{\mathrm{target}}(x^i)/p^c_\theta(x^i)\right)^2}. \end{aligned}$$

We can rewrite the weights for the regular proposal's ESS calculation as follows,

$$w(x^i) = \frac{\mu_{\text{target}}(\bar{x}^i + c)}{p_\theta(\bar{x}^i + c)} = \frac{\mu_{\text{target}}(\bar{x}^i)}{p_\theta(\bar{x}^i)p(c)}$$
$$w(x^i) = w(\bar{x}^i) \cdot w(c)$$

where we exploited the translation invariance of μ_{target} to remove the center of mass c and also the independence between the mean $\bar{x}$ and c in the proposal. Since $c \sim \mathcal{N}(0, \sigma^2)$ we can write $p(c) = p(\|c\|, \theta, \phi)$ in spherical coordinates to follow a scaled Chi distribution $\|c\| \sim \sigma\chi_3(\|c\|)$ with angular components that follow independent uniform distributions $(\theta, \phi) \sim \mathcal{U}(\theta)\mathcal{U}(\phi)$. Fixing canonical angular components (θ, ϕ) we have $p(c) = \sigma\chi_3(\|c\|)$ and $w(c) = 1/\sigma\chi_3(\|c\|)$.

For the adjusted proposal we set $p(c) = 1/\sigma\chi_3$ which gives the following weights:

$$w^c(x^i) = \frac{\mu_{\text{target}}(\bar{x}^i + c)}{p^c_\theta(\bar{x}^i + c)} = \frac{\mu_{\text{target}}(\bar{x}^i)}{p_\theta(\bar{x}^i)p(c)} = \frac{\mu_{\text{target}}(\bar{x}^i)}{p_\theta(\bar{x}^i)} = w(\bar{x}^i). \tag{13}$$

We now seek to prove that:

$$\frac{\left(\sum_{i=1}^K w(\bar{x}^i)w(c)\right)^2}{\sum_{i=1}^K w(\bar{x}^i)^2 w(c)^2} < \frac{\left(\sum_{i=1}^K w(\bar{x}^i)\right)^2}{\sum_{i=1}^K w(\bar{x}^i)^2}. \tag{14}$$

Now, denote $X^i = w(\bar{x}^i)$ and $Y^i = w(c)$ random variables over the sample index i. By construction, X^i and Y^i are independent for each i. Furthermore, all (X^i, Y^i) pairs are i.i.d. for $i \in [K]$.

We may now formalize equation 14 as proving

$$\frac{\left(\mathbb{E}[X\,Y]\right)^2}{\mathbb{E}[(X\,Y)^2]} < \frac{\left(\mathbb{E}[X]\right)^2}{\mathbb{E}[X^2]}. \tag{15}$$

Because X and Y are independent for each sample we know:

$$\mathbb{E}[X\,Y] = \mathbb{E}[X]\,\mathbb{E}[Y], \qquad \mathbb{E}[(X\,Y)^2] = \mathbb{E}[X^2]\,\mathbb{E}[Y^2]. \tag{16}$$

Hence

$$\frac{\left(\mathbb{E}[X\,Y]\right)^2}{\mathbb{E}[(X\,Y)^2]} = \frac{\left(\mathbb{E}[X]\right)^2}{\mathbb{E}[X^2]} \times \frac{\left(\mathbb{E}[Y]\right)^2}{\mathbb{E}[Y^2]}. \tag{17}$$

If $\mathrm{Var}(Y) > 0$, then $\mathbb{E}[Y^2] > (\mathbb{E}[Y])^2$. Consequently,

$$0 < \frac{\left(\mathbb{E}[Y]\right)^2}{\mathbb{E}[Y^2]} < 1. \tag{18}$$

Thus, at the level of population expectations,

$$\frac{\left(\mathbb{E}[X\,Y]\right)^2}{\mathbb{E}[(X\,Y)^2]} = \frac{\left(\mathbb{E}[X]\right)^2}{\mathbb{E}[X^2]} \times \underbrace{\frac{\left(\mathbb{E}[Y]\right)^2}{\mathbb{E}[Y^2]}}_{<1} < \frac{\left(\mathbb{E}[X]\right)^2}{\mathbb{E}[X^2]}\ . \tag{19}$$

Therefore, applying the adjustment is strictly better by ESS than unadjusted.

□

B.2. Proof of Lemma 1

We first prove a useful lemma that computes the total variation distance between the original distribution of the normalizing flow p_θ and the truncated distribution $\hat{p}_\theta$ before proving the propositions.

Lemma 1. *Let p_θ be a generative and denote $\hat{p}_\theta(x)$ the δ-truncated distribution such that $\hat{p}_\theta(x) := \mathbb{P}(p_\theta(x) \geq \delta)$, for a small $\delta > 0$. Define the constant $\beta = \mathbb{P}(p_\theta(x) < \delta)$ as the event where the truncation occurs. Then the total variation distance between the generative model and its truncated distribution is TV$(p_\theta, \hat{p}_\theta) = \beta$.*

Proof. We begin by first characterizing the total variation distance between flow after correction with importance sampling $p(x)$ with truncated distribution $\hat{p}(x)$. Recall that the truncated distribution is defined as follows:

$$\hat{p}(x) := \mathbb{P}(p(x) \geq \delta) = \frac{p(x)\mathbb{I}\{p(x) \geq \delta\}}{\int \mathbb{I}\{p(x) \geq \delta\}p(x)dx}, \tag{20}$$

where $\mathbb{I}$ is the indicator function. Denote the events $\alpha = \mathbb{P}(X \geq \delta)$ and $\beta = \mathbb{P}(X < \delta)$ for the random variance $X \sim p(x)$. Clearly, $\alpha + \beta = 1$ and $\alpha = \int \mathbb{I}\{\mu(x) \geq \delta\}p(x)dx$. Now consider the total variation distance between these two distributions:

$$\text{TV}(p, \hat{p}) = \sup_{\phi \in \Phi} \left|\mathbb{E}_{x\sim p(x)}[\phi(x)] - \mathbb{E}_{\hat{x}\sim \hat{p}(x)}[\phi(\hat{x})]\right| = \frac{1}{2}\int |p(x) - \hat{p}(x)|dx. \tag{21}$$

where $\Phi = \{\phi : \|\phi\|_\infty \leq 1\}$. Next we break up the event space into two regions R_1 and R_2 which correspond to the events $p(x) < \delta$ and $p(x) \geq \delta$ respectively. Now consider the total variation distance in the region R_1 whereby construction $\hat{p}(x) = 0$,

$$\frac{1}{2}\int_{R_1} |p(x) - \hat{p}(x)|dx = \frac{1}{2}\int_{R_1} p(x)dx = \frac{\beta}{2}. \tag{22}$$

A similar computation on R_2 gives,

$$\frac{1}{2}\int_{R_2} |p_\theta(x) - \hat{p}_\theta(x)|dx = \frac{1}{2}\int_{R_2} \left|p(x) - \frac{p(x)}{\alpha}\right| dx = \frac{1}{2}\int_{R_2} p(x)\left|1 - \frac{1}{\alpha}\right| dx = \frac{\alpha(\frac{1}{\alpha} - 1)}{2} = \frac{\beta}{2}, \tag{23}$$

where we exploited the fact that $\hat{p_\theta}(x) = \frac{p_\theta(x)}{\alpha}$ in the first equality and that $\alpha = \int_{R_2} p_\theta(x)dx$ in the second equality. Combining these results we get the full total variation distance:

$$\text{TV}(p, \hat{p}) = \frac{1}{2}\int |p(x) - \hat{p}(x)|dx = \frac{1}{2}\int_{R_1} |p(x) - \hat{p}(x)|dx + \frac{1}{2}\int_{R_2} |p(x) - \hat{p}(x)|dx = \beta. \tag{24}$$

Thus the TV$(p, \hat{p}) = \beta$ and 0 in the trivial case where $\alpha = 1$ and the truncated distribution are the same. □

B.3. Proof of Proposition 3

Proposition 3. *Given an energy threshold $\mathcal{E}(x) > \gamma$, for $\gamma > 0$ large and the resulting truncated target distribution $\hat{\mu}_{target}(x) := \mathbb{P}\left(\mu_{target}(x) \geq \frac{\gamma}{\log \hat{\mathcal{Z}}}\right)$. Further, assume that the density of unnormalized importance weights w.r.t. to $\hat{\mu}_{target}$ is square integrable $(\hat{w}(x))^2 < \infty$. Given a tolerance $\rho = 1/ESS$ and bias of the original importance sampling estimator in total variation $b =$ TV$(\mu_\theta, \mu_{target})$, then the γ-truncation threshold with K-samples for TV$(\mu_\theta, \hat{\mu}_{target})$ is:*

$$\gamma \geq \frac{1}{\lambda}\log\left(\frac{Kb}{12\rho\mathbb{E}[\exp(-\lambda X)]}\right) + \log\hat{\mathcal{Z}}. \tag{11}$$

Proof. We start by recalling a well-known result stating the bias of self-normalized importance sampling found in Agapiou et al. (2017, Theorem 2.1) using K samples from the proposal $\mu(x)$:

$$\sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\mu_\theta^K(\phi) - \mu_{\text{target}}(\phi)\right]\right| \leq \frac{12\rho}{K}, \quad \rho \approx \frac{K}{\text{ESS}} = \frac{K\sum_j^K w(x^j)^2}{\left(\sum_i^K w(x^i)\right)^2} \tag{25}$$

where the terms $\mu_\theta^K(\phi) = \sum_i^K \bar{w}(x^i)\phi(x^i)$ is the self-normalized importance estimator of μ_{target} with samples drawn according to $x^i \sim p_\theta(x)$ and $\|\phi(x)\| \leq 1$ is a bounded test function.

By truncating using an energy threshold $\mathcal{E}(x) < \gamma$, for a large $\gamma > 0$, we truncate the support of $\mu_{\text{target}}(x)$ by cutting off low probability regions that constitute high-energy configurations. More precisely, we have $\hat{\mu}_{\text{target}} := \mathbb{P}\left(\mu_{\text{target}}(x) \geq \frac{\gamma}{\log \hat{\mathcal{Z}}}\right)$, where $\log \hat{\mathcal{Z}}$ is as defined in Eq. 10. Note that $\hat{\mu}_{\text{target}}(x)$ is absolutely continuous w.r.t. to μ_{target} as the support is contained up to modulo measure zero sets. The importance sampling error incurred by using $\hat{\mu}_{\text{target}}$ can be bounded as follows:

$$\sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\mu_\theta^K(\phi) - \mu_{\text{target}}(\phi)\right]\right| \leq \sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\mu_\theta^K(\phi) - \hat{\mu}_{\text{target}}(\phi)\right]\right| + \sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\hat{\mu}_{\text{target}}(\phi) - \mu_{\text{target}}(\phi)\right]\right| \tag{26}$$

$$\leq \frac{12\hat{\rho}}{K} + \beta_1 \tag{27}$$

$$\leq \frac{12\rho}{K} + \beta_1. \tag{28}$$

The first inequality follows from the triangle inequality. Here we note that $\hat{\rho}$ is the ESS which corresponds to using importance weights computed with respect to the truncated target $\hat{\mu}_{\text{target}}$ rather than μ_{target}. The constant $\beta_1 = \text{TV}(\hat{\mu}_{\text{target}}, \mu_{\text{target}})$ and follows from an application of Lemma 1. Further, note that $\rho \geq \hat{\rho}$ since ESS must increase—and thereby $\hat{\rho}$ decreases—as the distributional overlap between the two distributions decreases. Now observe, $\beta_1 = \mathbb{P}\left(X < \frac{\gamma}{\log \hat{\mathcal{Z}}}\right)$, where samples follow the law $X \sim \hat{\mu}_{\text{target}}(x)$. Then a direct application of Chernoff's inequality gives us $\mathbb{P}\left(X < \frac{\gamma}{\log \hat{\mathcal{Z}}}\right) = \beta_1 \leq \exp\left(\frac{\lambda\gamma}{\log \hat{\mathcal{Z}}}\right) \mathbb{E}[\exp(-\lambda X)]$. Thus the additional bias incurred is,

$$\sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\hat{\mu}_\theta^K(\phi) - \mu_{\text{target}}(\phi)\right]\right| \leq \frac{12\rho}{K} + \beta_1 \leq \frac{12\rho}{K} + \exp\left(\frac{\lambda\gamma}{\log \hat{\mathcal{Z}}}\right) \mathbb{E}[\exp(-\lambda X)]. \tag{29}$$

Where the term $\mathbb{E}[\exp(-\lambda X)]$ is the moment generating function. Setting $b := \text{TV}(\mu_\theta^K, \mu_{\text{target}})$, then we have

$$\gamma \geq \frac{1}{\lambda} \log\left(\frac{Kb}{12\rho\mathbb{E}[\exp(-\lambda X)]}\right) + \log \hat{\mathcal{Z}}. \tag{30}$$

□

B.4. Proof of Proposition 4

Proposition 4. *Assume that the density of the model p_θ after importance sampling μ_θ is absolutely continuous with respect to the target μ_{target}. Further, assume that the density of unnormalized importance weights is square integrable $(w(x))^2 < \infty$. Given a tolerance $\rho = 1/ESS$ of the original importance sampling estimator under μ_θ and bias of the importance sampling estimator in total variation $b = TV(\mu_\theta, \mu_{target})$, then the δ-truncation for the truncated distribution $\hat{p}_\theta(x) := \mathbb{P}(p_\theta(x) \geq \delta)$ threshold with K-samples is:*

$$\delta \geq \frac{1}{\lambda} \log\left(\frac{Kb}{12\rho\mathbb{E}[\exp(-\lambda X)]}\right). \tag{31}$$

Proof. We aim to bound the total variation distance $\text{TV}(\hat{\mu}_\theta^K, \mu_{\text{target}})$ of using the truncated distribution $\mathbb{P}(p_\theta(x) > \delta)$ by again recalling the bias of self-normalized importance sampling using K samples from $\mu_\theta(x)$:

$$\sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\mu_\theta^K(\phi) - \mu_{\text{target}}(\phi)\right]\right| \leq \frac{12\rho}{K}, \quad \rho \approx \frac{K}{\text{ESS}} = \frac{K\sum_j^K w(x^j)^2}{\left(\sum_i^K w(x^i)\right)^2} \tag{32}$$

where the terms $\mu_\theta^K(\phi) = \sum_i^K \bar{w}(x^i)\phi(x^i)$ is the self-normalized importance estimator of μ_{target} with samples drawn according to $x^i \sim p_\theta(x)$ and $\|\phi(x)\| \leq 1$ is a bounded test function. We next characterize the error introduced by using the truncated distribution $\hat{p}_\theta$ for importance sampling in place of p_θ by first defining the truncated K-sample self-normalized

importance estimator $\hat{\mu}_\theta^K(\phi) = \sum_j^K \bar{w}(x^j)\phi(x^j)$, where $x^j \sim \hat{p}_\theta(x)$. Specifically, we bound the total variation distance:

$$\begin{aligned}
\mathrm{TV}(\mu_\theta, \hat{\mu}_\theta) &= \sup_{\|\phi\|_\infty \leq 1} \left| \mathbb{E}\left[\mu_\theta^K(\phi) - \hat{\mu}_\theta^K(\phi)\right] \right| && (33)\\
&= \sup_{\|\phi\|_\infty \leq 1} \left| \mathbb{E}_{x^i \sim p_\theta}\left[\sum_{i=1}^K \bar{w}(x^i)\phi(x^i)\right] - \mathbb{E}_{x^j \sim \hat{p}_\theta}\left[\sum_{j=1}^K \bar{w}(x^j)\phi(x^j)\right] \right| && (34)\\
&= \frac{1}{2}\left(\mathbb{E}_{x^i \sim p_\theta}\left[\sum_{i=1}^K \bar{w}(x^i)\right] - \mathbb{E}_{x^j \sim \hat{p}_\theta}\left[\sum_{j=1}^K \bar{w}(x^j)\right] \right) && (35)
\end{aligned}$$

Here in the second equality, we used the fact that the test function is bounded $||\phi|||_\infty \leq 1$ Next, we apply Lemma 1 and leverage the fact that the self-normalized weights are also bounded and achieve a bound on the total variation distance,

$$\begin{aligned}
\mathrm{TV}(\mu, \hat{\mu}) &= \frac{1}{2}\left(\mathbb{E}_{x^i \sim p_\theta}\left[\sum_{i=1}^K \bar{w}(x^i)\right] - \mathbb{E}_{x^j \sim \hat{p}_\theta}\left[\sum_{j=1}^K \bar{w}(x^j)\right] \right) && (36)\\
&= \beta_2, && (37)
\end{aligned}$$

where β_2 is the probability mass $\mathbb{P}(X < \delta)$ when $X \sim p_\theta(x)$. Like previously, the overall error can be bounded using the triangle inequality

$$\begin{aligned}
\sup_{\|\phi\|_\infty \leq 1} \left| \mathbb{E}\left[\mu_\theta^K(\phi) - \mu_{\text{target}}(\phi)\right] \right| &\leq \sup_{\|\phi\|_\infty \leq 1} \left| \mathbb{E}\left[\hat{\mu}_\theta^K(\phi) - \mu_{\text{target}}(\phi)\right] \right| + \sup_{\|\phi\|_\infty \leq 1} \left| \mathbb{E}\left[\mu_\theta^K(\phi) - \hat{\mu}_\theta^K(\phi)\right] \right| && (38)\\
&\leq \frac{12\hat{\rho}}{K} + \beta_2 && (39)\\
&\leq \frac{12\rho}{K} + \beta_2. && (40)
\end{aligned}$$

Where the last inequality follows from the same logic as in Proposition 3 where ESS goes up after truncation and therefore $\rho > \hat{\rho}$. A direct application of Chernoff's inequality gives us $\mathbb{P}(X < \delta) = \beta_2 \leq \exp(\lambda\delta)\mathbb{E}[\exp(-\lambda X)]$ where we used the moment generating function of $p_\theta(x)$. Thus the additional bias incurred is,

$$\sup_{\|\phi\|_\infty \leq 1} \left| \mathbb{E}\left[\mu_\theta^K(\phi) - \mu_{\text{target}}(\phi)\right] \right| \leq \frac{12\rho}{K} + \beta_2 \leq \frac{12\rho}{K} + \exp(\lambda\delta)\mathbb{E}[\exp(-\lambda X)]. \tag{41}$$

Setting $b := \mathrm{TV}(\mu_\theta, \mu_{\text{target}})$ as the bias, then we have

$$\delta \geq \frac{1}{\lambda}\log\left(\frac{Kb}{12\rho\mathbb{E}[\exp(-\lambda X)]}\right). \tag{42}$$

□

C. Itô Filtering

C.1. Flow Matching SDE

As shown in Domingo-Enrich et al. (2025) we can write Flow Matching with Gaussian conditional paths and Diffusion models under a unified SDE framework given a reference flow:

$$x_t = \beta_t x_0 + \alpha_t x_1, \tag{43}$$

where $(\alpha_t)_{t\in[0,1]}, (\beta_t)_{t\in[0,1]}$ are functions such that $\alpha_0 = \beta_1 = 0$ and $\alpha_1 = \beta_0 = 1$. In the specific case of flow matching with linear interpolants that we consider we have:

$$x_t = (1-t)x_0 + tx_1. \tag{44}$$

The unified SDE for both flow matching and continuous-time diffusion models as introduced in Domingo-Enrich et al. (2025) is then:

$$dx_t = \kappa_t x + \left(\frac{\sigma_t^2}{2} + \eta_t\right)\mathfrak{s}(x_t, t) + \sigma_t dW_t, \quad \kappa_t = \frac{\dot{\alpha}_t}{\alpha_t}, \eta_t = \beta_t\left(\frac{\dot{\alpha}_t}{\alpha_t}\beta_t - \dot{\beta}_t\right) \tag{45}$$

where $\mathfrak{s}(x_t, t)$ is the score function estimated by the diffusion model. Thus the flow matching SDE is:

$$dx_t = \left(2f_{t,\theta}(t, x_t) - \frac{x_t}{t}\right)dt + \sigma_t dW_t, \quad \sigma_t = \sqrt{(2(1-t)t)} \tag{46}$$

In fact, the Stein score can be estimated from the output of a velocity field and vice-versa:

$$\nabla \log p_t(x_t) = \frac{t f_{t,\theta}(t, x_t) - x_t}{1-t}, \quad f_{t,\theta}(t, x_t) = \frac{x_t + (1-t)\nabla \log p_t(x_t)}{t} \tag{47}$$

Rewriting Eq. 46 in terms of the score function we get,

$$dx_t = \frac{x_t}{t} + \sigma_t^2 \nabla \log p_t(x_t) + \sigma_t dW_t. \tag{48}$$

C.2. Itô Filtering

Proposition 5. *Assume that the density of the model p_θ after importance sampling μ_θ is absolutely continuous with respect to the target μ_{target}. Further, assume that the density of unnormalized importance weights is square integrable $(w(x))^2 < \infty$. Let $r(x_0)$ be the Itô density estimator for $\log p_0(x_0)$ of the flow matching SDE:*

$$dx_t = \frac{x_t}{t} + \sigma_t^2 \nabla \mathfrak{s}_\theta(t, x_t) + \sigma_t dW_t, \quad \sigma_t = \sqrt{(2(1-t)t)}. \tag{49}$$

Given $\rho = 1/ESS$, and $\zeta > 0$ which is the weight clipping threshold. Then the additional bias of using the Itô density estimator for importance sampling $\hat{\mu}_{r,\theta}$ with clipping is:

$$\sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\mu_{r,\theta}^K(\phi) - \mu_{target}(\phi)\right]\right| \leq \frac{12\rho}{K} + \beta_3 + \beta_4, \tag{50}$$

where $\beta_3 = TV(\mu_{r,\theta}, \mu_\theta)$ and $\beta_4 = TV(\mu_{r,\theta}, \hat{\mu}_{r,\theta})$.

We now recall Itô's lemma which states that for a stochastic process,

$$dx_t = f_t(t, x_t) + g_t dW_t, \tag{51}$$

and a smooth function $h : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$ the variation of h as a function of the stochastic SDE can be approximated using a Taylor approximation:

$$dh(t, x_t) = \left(\frac{\partial}{\partial t}h(t, x_t) + \frac{\partial}{\partial x}h(t, x_t)^T f_t(t, x_t) + \frac{1}{2}\sigma_t^2 \Delta_x h(t, x_t)\right)dt + \sigma_t \frac{\partial}{\partial x}h(t, x_t)dW_t. \tag{52}$$

where Δ_x is the Laplacian. We will use Itô's Lemma with $h(t, x_t) := \log p_t(x_t)$ to obtain the Itô density estimator (Skreta et al., 2025; Karczewski et al., 2024) but for flow models

$$d\log p_t(x_t) = \left(\frac{\partial}{\partial t}\log p_t(x_t) + \frac{\partial}{\partial x}\log p_t(x_t)^T f(t, x_t) + \frac{1}{2}\sigma_t^2 \Delta_x \log p_t(x_t)\right)dt + \sigma_t \frac{\partial}{\partial x}\log p_t(x_t)dW_t, \tag{53}$$

To solve for the change in density over time we can start from the log version of the Fokker-Plank equation:

$$\frac{\partial}{\partial t}\log p_t(x) = -\nabla \cdot (f(t, x)) + \frac{1}{2}\sigma_t^2 \Delta_x \log p_t(x) - \nabla_x \log p_t(x)^T \left(f(t, x) - \frac{1}{2}\sigma_t^2 \nabla_x \log p_t(x)\right) \tag{54}$$

in the general case we end with:

$$d\log p_t(x_t) = \left(-\nabla \cdot \left(f(t, x_t) - \sigma_t^2 \nabla_x \log p_t(x_t)\right) + \frac{1}{2}\sigma_t^2 \|\nabla_x \log p_t(x_t)\|^2\right)dt + \sigma_t \nabla_x \log p_t(x_t)^T dW_t. \tag{55}$$

We now apply this to the flow-matching SDE Eq. 48 written in terms of the score function. In particular, we have,

$$\begin{aligned} d\log p_t(x_t) &= \left(-\nabla\cdot\left(\sigma_t^2\nabla_x \log p_t(x_t) + \frac{x_t}{t} - \sigma_t^2\nabla_x \log p_t(x_t)\right) + \frac{1}{2}\sigma_t^2\|\nabla_x \log p_t(x_t)\|^2\right)dt \\ &\quad + \sigma_t\nabla_x \log p_t(x_t)^T dW_t \\ d\log p_t(x_t) &= \left(-d/t + \frac{1}{2}\sigma_t^2\|\nabla_x \log p_t(x_t)\|^2\right)dt + \sigma_t\nabla_x \log p_t(x_t)^T dW_t. \end{aligned} \tag{56}$$

The above equation makes an implicit assumption that we have access to the actual ground truth score function of $\nabla \log_t(x_t)$ rather than the estimated one $\mathfrak{s}_\theta$, expressed via the vector field as in Eq. 47. When working with imperfect score estimates we have the following SDE:

$$dx_t = \frac{x_t}{t} + \sigma_t^2\nabla\mathfrak{s}_\theta(t, x_t) + \sigma_t dW_t. \tag{57}$$

The score estimation error causes a discrepancy in $\log p_t(x_t)$ estimates whose error is captured in the theorem from Karczewski et al. (2024)[Theorem 3]:

$$\log r_0(x_0) = \log p_0(x_0) + Y \tag{58}$$

where $\log r_0$ is the bias of the log density starting at time $t = 0$ of the auxiliary process that does not track x_t correctly due to the estimation error of the score. Also, Y is a random variable such that that bias of r_0 is given by:

$$\mathbb{E}[Y] = \underbrace{\frac{1}{2}\mathbb{E}_{t\sim\mathcal{U}(0,1), x_t\sim p_t(x_t)}\left[\sigma_t^2||\mathfrak{s}_\theta(t, x_t) - \nabla\log p_t(x_t)||^2\right]}_{\geq 0} \tag{59}$$

Thus the Itô density estimator forms an upper bound to the true log density, i.e. $r_0(x_0) \geq \log p_0(x_0)$. This allows us to form an upper bound on the normalized log weights as an expectation,

$$\begin{aligned} \mathbb{E}_{x_0\sim p_\theta(x_0)}[\log \bar{w}(x_0)] &= \mathbb{E}_{x_0\sim p_\theta(x_0)}\left[-\frac{\mathcal{E}(x_0)}{k_B T} - \log p_0(x_0) - C\right] \\ &\leq \mathbb{E}_{x_0\sim p_\theta(x_0)}\left[-\frac{\mathcal{E}(x_0)}{k_B T} - r_0(x_0)\right], \end{aligned}$$

where C is a constant. We define $\log \bar{w}_r(x_0) := -\frac{\mathcal{E}(x_0)}{k_B T} - r_0(x_0)$ as the new normalized importance weights, module constants. We can now compute the additional bias of self-normalized importance sampling estimator $\mu_{r,\theta}^K$

$$\begin{aligned} \mathrm{TV}(\mu_{r,\theta}, \mu_\theta) &= \sup_{\|\phi\|_\infty\leq 1}\left|\mathbb{E}\left[\mu_{r,\theta}^K(\phi) - \mu_\theta^K(\phi)\right]\right| && (60)\\ &= \sup_{\|\phi\|_\infty\leq 1}\left|\mathbb{E}_{x^i\sim p_\theta}\left[\sum_{i=1}^K \bar{w}_r(x^i)\phi(x^i)\right] - \mathbb{E}_{x^j\sim p_\theta}\left[\sum_{j=1}^K \bar{w}(x^j)\phi(x^j)\right]\right| && (61)\\ &= \frac{1}{2}\left(\mathbb{E}_{x^i\sim p_\theta}\left[\sum_{i=1}^K \bar{w}_r(x^i)\right] - \mathbb{E}_{x^j\sim p_\theta}\left[\sum_{j=1}^K \bar{w}(x^j)\right]\right) && (62)\\ &= \frac{1}{2}\left(\mathbb{E}_{x^i\sim p_\theta}\left[\sum_{i=1}^K \exp\left(\frac{1}{2}\mathbb{E}_{t\sim\mathcal{U}(0,1), x_t\sim p_t(x_t)}\left[\sigma_t^2||\mathfrak{s}_\theta(t, x_t) - \nabla\log p_t(x_t)||^2\right]\right)\right]\right) && (63)\\ &:= \beta_3 && (64) \end{aligned}$$

The total bias is then

$$\sup_{\|\phi\|_\infty \leq 1} \left| \mathbb{E}\left[\mu^K_{r,\theta}(\phi) - \mu_{\text{target}}(\phi)\right] \right| \leq \frac{12\rho}{K} + \beta_3. \tag{65}$$

Finally, when clipping weights with $\zeta > 0$ we induce a truncated distribution $\hat{\mu}_{r,\theta}$, i.e. $\hat{r}_0 := \mathbb{P}(r_0 x_0 > \zeta)$. Using Lemma 1 this creates another constant factor that contributes $\text{TV}(\mu_{r,\theta}, \hat{\mu}_{r,\theta}) = \beta_4$ to the overall bias:

$$\sup_{\|\phi\|_\infty \leq 1} \left| \mathbb{E}\left[\mu^K_{r,\theta}(\phi) - \mu_{\text{target}}(\phi)\right] \right| \leq \frac{12\rho}{K} + \beta_3 + \beta_4. \tag{66}$$

D. Datasets

For all datasets besides alanine dipeptide we use a training set of 10^5 contiguous samples (1 µs simulation time) from a single MCMC chain, a validation set of the next $2 \cdot 10^4$ contiguous samples (0.2 µs simulation time), and a test set of 10^4 uniformly strided subsamples from the remaining trajectory. Since these are highly multimodal energy functions, this leaves us with biased training data relative to the Boltzmann distribution; we split trajectories this way to test the model in a challenging and realistic setting. We describe the datasets below and present the simulation parameters in Table 4. Ramachandran plots for the training and test data (before subsampling) are provided in Figures 9 to 12.

Table 4. Overview of molecular dynamics simulation parameters.

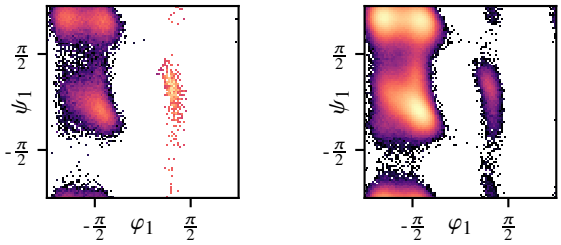
Peptide	Force field	Temperature	Time step
Alanine dipeptide	Amber ff99SBildn	300K	1fs
Trialanine	Amber 14	310K	1fs
Alanine tetrapeptide	Amber ff99SBildn	300K	1fs
Hexaalanine	Amber 14	310K	1fs
Chignolin	Amber 14	310K	1fs

Alanine dipeptide. For this dataset we use the data and data split from Klein & Noé (2024). Here the training set is purposely biased with an overrepresentation of an underrepresented mode, i.e. the positive φ state. This bias makes it easier to reweight to the target Boltzmann distribution. Alanine Dipeptide consist of one Alanine amino acids, an acetyl group, and an N-methyl group.

Trialanine and hexa-alanine. For the peptides composed of multiple alanine amino acids, we generate MD trajectories using the *OpenMM* library (Eastman et al., 2017). All simulations are conducted in implicit solvent, with the simulation parameters detailed in Table 4. These systems do not include any additional capping groups, such as those present in alanine dipeptide and alanine tetrapeptide, as they are generated in the same manner as described in Klein et al. (2023a). There are two peptide bonds in trialanine and five in hexa-alanine, resulting in two and five Ramachandran plots respectively.

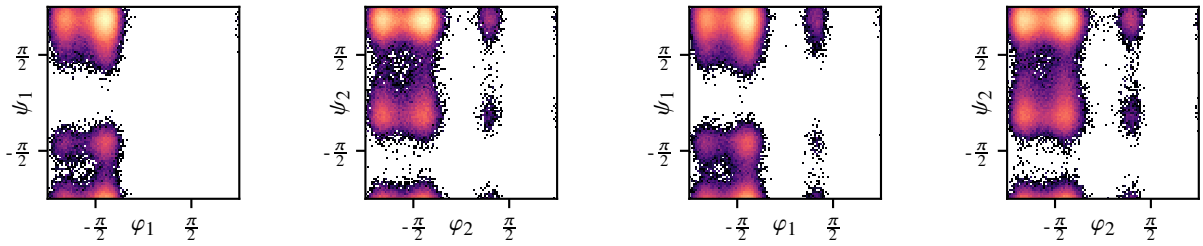
Alanine Tetrapeptide (AD4). For this dataset we use the same system setup as in Dibak et al. (2022), but treat all bonds as flexible. The original dataset kept all hydrogen bonds fixed, as the Boltzmann Generator was operating in internal coordinates. The MD simulation to generate the dataset is then performed as described above. Alanine Tetrapeptide consist of three Alanine amino acids, an acetyl group, and an N-methyl group. Therefore, there are four Ramachandran plots.

Chignolin. In addition to the small peptide systems, we also investigate the small protein chignolin, consisting of ten amino acids (`GYDPETGTWG`). We simulate this system using the same configuration as trialanine, defined in Table 4 for 4.5 µs.



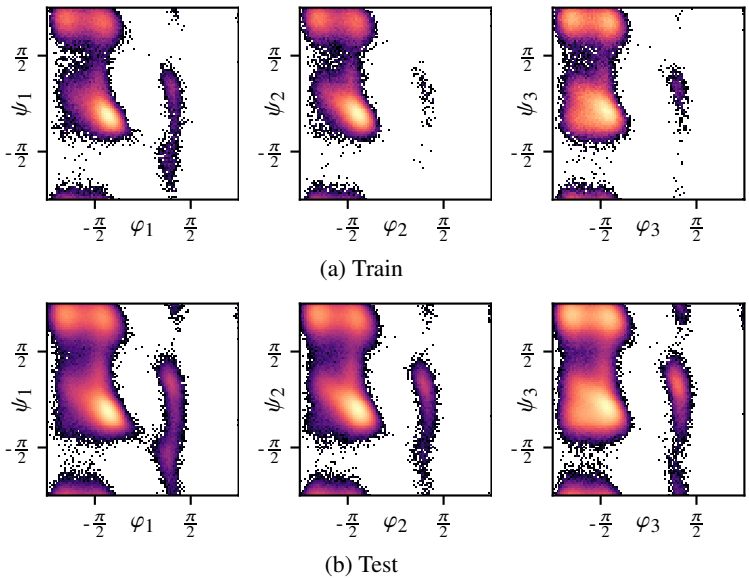
(a) Train (b) Test

Figure 9. Alanine dipeptide Ramachandran plots for train and test data subsets. The lower probability state is oversampled in the training data as in Klein & Noé (2024).



(a) Train (b) Test

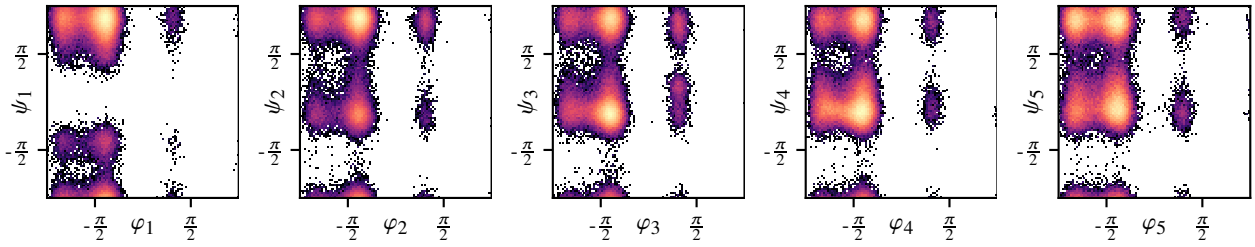
Figure 10. Trialanine Ramachandran plots for train and test data. We observe a missing mode in the first Ramachandran plot of the training subset.



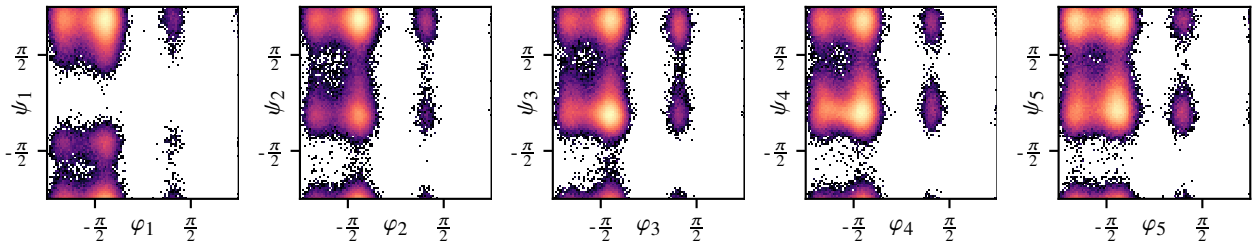
(a) Train

(b) Test

Figure 11. Alanine tetrapeptide Ramachandran plots for train and test data subsets. We obverse an underrepresented mode in the second Ramachandran plot of the training subset.



(a) Train



(b) Test

Figure 12. Hexa-alanine Ramachandran plots for train and test data subsets. We observe good mode coverage of the training data.

E. Experimental Details

E.1. Metrics

For computational efficiency we subsample 10^4 reference samples from the evaluation trajectory to serve as ground truth. Similarly in cases where a method produces more than 10^4 samples, a random subset of size 10^4 is selected for comparison. We quantify distributional similarity using empirical Wasserstein-2 distances between generated samples and the reference data. Given two empirical distributions, $\mu = \frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}$ and $\nu = \frac{1}{m}\sum_{j=1}^{m}\delta_{y_j}$, the Wasserstein-2 distance is computed as

$$W_2(\mu,\nu) = \min_{\pi\in\Pi(\mu,\nu)} \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m}\pi_{ij}\, c(x_i,y_j)^2}, \tag{67}$$

where $\Pi(\mu,\nu)$ denotes the set of admissible transport plans and $c(x,y)^2$ is a defined cost function. Optimal couplings are computed using the `POT` library (Flamary et al., 2021).

Energy cost. To assess energy distribution similarity, we define the following cost

$$c_E(x,y)^2 = |E(x) - E(y)|^2 . \tag{68}$$

Dihedral torus cost. We evaluate macrostructural similarity in the space of backbone dihedral angles (ϕ,ψ), which encode conformational information. For a molecule with L residues, we define the angle vector:

$$\text{Dihedrals}(x) = (\phi_1,\psi_1,\ldots,\phi_{L-1},\psi_{L-1}). \tag{69}$$

Due to angle periodicity, we define the cost on the resulting torus as:

$$c_{\mathbb{T}}(x,y)^2 = \sum_{i=1}^{2L}\left[(\text{Dihedrals}(x)_i - \text{Dihedrals}(y)_i + \pi) \bmod 2\pi - \pi\right]^2 , \tag{70}$$

capturing angular deviations while respecting circular geometry.

TICA projection cost Time-lagged Independent Component Analysis (TICA) performs dimensionality reduction for identification of slow dynamical modes. From the mean-centered time series $\tilde{x}_t$, we compute:

$$\hat{C}_{00} = \frac{1}{T-\tau}\sum_{t=1}^{T-\tau}\tilde{x}_t\tilde{x}_t^\top, \quad \hat{C}_{0\tau} = \frac{1}{T-\tau}\sum_{t=1}^{T-\tau}\tilde{x}_t\tilde{x}_{t+\tau}^\top, \tag{71}$$

and solve the generalized eigenvalue problem:

$$C_{0\tau}w = \lambda C_{00}w. \tag{72}$$

The top two eigenvectors w_1 and w_2 define projections capturing the most autocorrelated directions. Using these, we define the TICA cost:

$$c_{\text{TICA}}(x,y)^2 = \sum_{j=1}^{2}\left[w_j^\top x - w_j^\top y\right]^2. \tag{73}$$

Note that the TICA basis is computed from the full evaluation trajectory (un-subsampled), while the comparison subset is restricted to the 10^4 subset. TICA analysis is performed only on heavy atom coordinates.

E.2. Timings

Sampling time calculations. For the sampling inference times in Figure 7, we compute all times on a single NVIDIA L40S GPU, using the maximum power of two batch size possible.

Training time. For training times we compute all times on a single A100 80GB GPU except for $\mathrm{SE}(3)$-EACF which is trained on a single H100. We report the total time in hours until convergence for all methods in the table below.

Table 5. Training time (hours) for all methods.

Model	ALDP	AL3	AL4	AL6	Chignolin
$\mathrm{SE}(3)$-EACF	160	—	—	—	—
ECNF++	9.72	12.5	17.17	76.94	—
SBG	16.83	24.67	41.67	57.5	427.33

E.3. $\mathrm{SE}(3)$-EACF Implementation Details

Equivariant augmented coupling flow (EACF) (Midgley et al., 2023a). We adopt the original model configuration from (Midgley et al., 2023a) for our EACF baseline on ALDP. Specifically, we choose the more stable spherical-projection EACF with a 20-layer configuration. Each layer has two ShiftCoM layers and two core-transformation blocks. The EGNN used in the core transformation block consists of 3 message-passing layers with 128 hidden states. Stability enhancement tricks like stable MLP and dynamic weight clipping on each layer's output are fully applied. The model is trained for 50 epochs with a bathh size of 20 using Adam optimizer and peak learning rate of 1×10^{-4}. We use the default 20 samples for likelihood estimation.

EACF as a Boltzmann generator. EACF leverages augmented dimensions, and therefore to estimate the likelihood of a sample x under an EACF model, we need to use an estimate based on samples from the augmented dimension a. Specifically, for a Gaussian distributed augmented variable a, we can estimate the marginal density of an observation as

$$q(x) = \mathbb{E}_{a\sim\pi(\cdot|x)}\left[\frac{q(x,a)}{\pi(a|x)}\right], \tag{74}$$

however, this is only a consistent estimator of the likelihood and for finite sample sizes has variance. This makes EACF unsuitable for our application of large-scale Boltzmann generators, as in this setting we need to compute exact likelihoods. Variance in likelihood estimation would lead to bias in the final distribution under self-normalized importance sampling or a SBG strategy. We therefore do not consider EACF as a viable option for large scale Boltzmann distribution sampling.

E.4. ECNF Implementation Details

E.4.1. Network and Training

Algorithm 2 ECNF flow matching training

Input: Prior q_0, Empirical samples from data q_1, bandwidth σ, batchsize b, initial network v_θ.
while Training **do**
  $\boldsymbol{x}_0 \sim q_0(\boldsymbol{x}_0)$; $\boldsymbol{x}_1 \sim q_1(\boldsymbol{x}_1)$ {Sample batches of size b *i.i.d.* from the dataset}
  $t \sim \mathcal{U}(0, 1)$
  $\mu_t \leftarrow t\boldsymbol{x}_1 + (1-t)\boldsymbol{x}_0$
  $x \sim \mathcal{N}(\mu_t, \sigma^2 I)$
  $\mathcal{L}(\theta) \leftarrow \|v_\theta(t, x) - (\boldsymbol{x}_1 - \boldsymbol{x}_0)\|^2$
  $\theta \leftarrow \text{Update}(\theta, \nabla_\theta \mathcal{L}(\theta))$
end while
Return v_θ

Equivariant continuous normalizing flow (ECNF) (Klein & Noé, 2024). We use the supplied pretrained model from Klein & Noé (2024) for our ECNF baseline on alanine dipeptide. Therefore all training parameters are equivalent to, and specified in, that work. We use the specification for the model "TBG + Full" in that work.

ECNF++. We note four improvements to the ECNF, which together substantially improve scalability.

1. **Flow matching loss.** In Klein & Noé (2024) a flow matching algorithm with smoothing is employed which provides extra stability during training. This is depicted in Alg. 2, however this smooths out the optimal target distribution (Tong et al., 2024, Proposition 3.3). ECNF uses $\sigma = 0.01$ where we use $\sigma = 0$ for ECNF++. We find that $\sigma > 0$ in this case causes poor molecular structures to be generated as the bond lengths are not able to be controlled precisely enough. We note that $\sigma = 0$ is used in most recent large scale flow matching models (Liu et al., 2024; Esser et al., 2024).
2. **Architecture size.** Empirically, we find the ECNF to be underparameterized. We perform a grid search over layer width and depth, finding a width of 256 and depth of 5 block to be a good balance between performance and speed on alanine dipeptide. We employ the same parameters for larger molecular systems.
3. **Improved optimizer and LR scheduler.** We find using an AdamW (Loshchilov & Hutter, 2017) with moderate weight decay of 10^{-4} improves performance and stability. Prior work has found weight decay helps to keep the Lipschitz constant of the flow low and avoids stiff dynamics which enables accurate ODE solving during inference. We also use a smoothly varying cosine schedule with warm-up (over 5% of iteration budget) which enables a larger maximum learning rate and faster training than the two step schedule used previously. Both the start and end learning rates are 500 times lower than the defined maximum.
4. **Exponential moving average.** We use an exponential moving average (EMA) on the weights with decay 0.999. This is standard practice in flow models, which improves performance.

These four elements together greatly improve the ECNF training, enabling larger systems to be successfully modeled, and provide a strong foundation for future Boltzmann generator training on molecular systems using equivariant continuous normalizing flows. Qualitatively, we find ECNFs quite stable to train and robust to training parameters relative to invertible architectures. However, it is very slow to compute the exact likelihoods necessary for importance sampling.

Other parameters. For inference we use a Dormand-Prince 45 (dopri5) adaptive step size solver with absolute tolerance 10^{-4} and relative tolerance 10^{-4}.

Likelihood evaluation. Evaluating the likelihood of a CNF model requires calculating the integral of the divergence, as in Eq. 2. While there exist fast unbiased approximations of the likelihood using Hutchinson's trace estimator (Hutchinson, 1990; Grathwohl et al., 2019), these are unfortunately unsuitable for Boltzmann generator applications where variance in the likelihood estimator leads to biased weights under self-normalized importance sampling. We therefore calculate the Jacobian using automatic differentiation which is both memory and time intensive. For example, on hexa-alanine, the maximum batch size that can fit on an 80GB A100 GPU is 8. This batch takes around 2 minutes for 84 integration steps. We also use

an improved vectorized Jacobian trace implementation for all CNF which reduces memory by roughly half and time by roughly 3x over the previous implementation (Klein & Noé, 2024).

On using a CNF proposal with SBG. In principle it is possible to drop in replace our NF architecture with a CNF in SBG. However, there are several drawbacks to such an approach, most notably in efficiency. As previously discussed, CNFs are extremely computationally inefficient to sample a likelihood from. We find on the order of 100 SBG steps are necessary for best performance. This would make CNFs at least two orders of magnitude slower to sample from, when they are already at the edge of tractability for the current importance sampling estimates. We leave it to future work to consider faster CNF likelihoods and note that our SBG algorithm could be applied there readily.

E.5. SBG Implementation Details

Architecture. We scale the TarFlow architecture for increasingly challenging datasets. As advised by Zhai et al. (2024) we scale the layers per block alongside the number of blocks. The layers / blocks / channels and resulting number of parameters are presented in Table 6. We note the larger number of parameters in the TarFlow relative to the ECNF++ despite the faster inference walltime, due to the lack of simulation and higher computational efficiency of the architecture.

Table 6. TarFlow configurations across different datasets.

Dataset	Layers per Block	Number Blocks	Channels	Number Parameters (M)
ALDP	4	4	256	13
AL3	6	6	256	29
AL4	6	6	384	64
AL6	6	6	384	64
Chignolin	8	8	384	114

Training configuration. The training hyperparameters used closely follow those of Zhai et al. (2024), although we deviate in using a larger value of weight decay as instability was observed during training. Namely we use a learning rate of 1×10^{-4}, weight decay of 4×10^{-4}, Adam β_1, β_2 of $(0.90, 0.95)$. We additionally employ the same cosine decay learning rate schedule with warmup (start and end learning rate 500 times lower than maximum value) and exponential moving average decay (0.999) used in ECNF++. Training is performed for 1000 epochs. Center of mass augmentation is applied with a standard deviation of $\frac{1}{\sqrt{n}}$, for n the number of particles, to match that of the prior for a given system. As non-monotonic improvement was observed on validation metrics, we use early stopping on the SNIS $\mathcal{E}$-$\mathcal{W}_2$ against the validation dataset.

Sampling hyperparameters. Whilst the TarFlow is capable of generating low-energy peptide states, it is also prone to generating samples of extremely high target energy. For standard importance sampling this presents no issue as these samples will be assigned negligible importance weight. However, for the SBG these high energy samples were prone to numerical instability during Langevin dynamics. To mitigate this issue we truncate the proposal distribution prior p_θ based on an energy cutoff, noting that the bias introduced by this operation is bounded in Proposition 3. Similarly to EACF, we additionally remove the samples corresponding to the largest 0.2% of importance weights, in the case of SBG this is performed once, prior to Langevin dynamics. See §F.3 for an ablation on the effect of weight clipping in SBG. For alanine systems we use 100 Langevin time steps, with $\text{ESS}_{\text{threshold}} = 0.5$ and $\epsilon = 1 \times 10^{-5}$ up to trialanine, and $\epsilon = 1 \times 10^{-6}$ thereafter. For chignolin we use 500 time steps, with $\text{ESS}_{\text{threshold}} = 0.5$ and $\epsilon = 1 \times 10^{-5}$. For ablations of these hyperparameters see §F.3.

Table 7. Overview of training configurations

Training Parameter	ECNF	ECNF++	TarFlow
Learning Rate	5×10^{-4}	5×10^{-4}	1×10^{-4}
Weight Decay	0.0	1×10^{-2}	4×10^{-4}
β_1, β_2	0.9, 0.999	0.9, 0.999	0.9, 0.95
EMA Decay	0.000	0.999	0.999
Width	64	256	Varies
N blocks	5	5	Varies
Parameters	152 K	2.317 M	Varies

F. Additional Results

F.1. Chirality

The ECNF architecture is E(3) equivariant, hence is equivariant to reflections, and will generate samples of both possible global chiralities. As the energy functions are themselves invariant to reflections this is not resolved by importance sampling. Having the correct global chirality is necessary to match the test dataset dihedral angle distributions where only one global chirality is present in the data. The incorrect chirality can show up as a symmetric mode on Ramachandran plots. To resolve this issue we follow Klein & Noé (2024) in detecting incorrect chirality conformations, and reflecting them. However, unlike Klein & Noé (2024) points with unresolvable symmetry (e.g mixed chirality conformations) *are not* dropped. The results for $\mathbb{T}$-$\mathcal{W}_2$ before and after fixed-chirality samples are presented in Table 8. We observe a reduction in metric value (improved performance) on all configurations, which we attribute to evaluation noise. We further note that non-equivariant methods such as SBG do not suffer this effect and hence do not require any symmetry post-processing.

Table 8. $\mathbb{T}$-$\mathcal{W}_2$ results for unprocessed and fixed-chirality samples from ECNF and ECNF++

Datasets →	Tripeptide (AL3)		Tetrapeptide (AL4)		Hexapeptide (AL6)	
Algorithm ↓	Unprocessed	Fixed	Unprocessed	Fixed	Unprocessed	Fixed
ECNF++ (Ours)	1.967 ± 0.062	1.177 ± 0.145	2.414 ± 0.000	2.082 ± 0.005	5.405 ± 0.069	4.315 ± 0.018

F.2. Ramachandran Plots

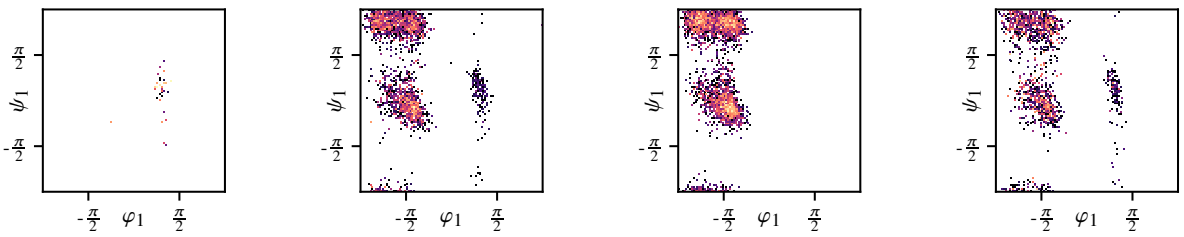
In this appendix we include the Ramachandran plots for each model on each peptide system. Please note that the ground truth training and test Ramachandran plots are presented in §D.

Alanine dipeptide. In Figure 13 we present the alanine dipetide Ramachandran plots. Both ECNF and SBG cover all relevant modes. We find that that ECNF++ models the distribution well, but drops the positive φ mode. This is notable as this mode is oversampled in the training data (see Figure 9).

Trialanine. In Figure 14 we can see the Ramachandran plot for resampled points for the trialanine dataset. Comparing this to the train and test data in Figure 10 we see that the AL3 training data is missing the φ_1 positive mode which is reflected in all of the models.

Alanine tetrapeptide. In Figure 15 we present Ramachandran plots for resampled points on the alanine tetrapeptide dataset. Comparing this to the train and test distributions in Figure 11 we observe that both ECNF++ and SBG capture the dihedral angle distribution with comparable success.

Hexa-alanine. In Figure 16 we present the Ramchandran plots for samples from ECNF++ and SBG. We find that SBG succeeds to capture the low density positive φ modes, albeit with a tight concentration of points as opposed to a broad range of low density. We additionally observe the negative Ψ_1 mode to be well captured by the SBG.



(a) EACF (b) ECNF (c) ECNF++ (d) SBG

Figure 13. Alanine dipeptide Ramachandran plots for baseline methods (SNIS) and SBG (SMC). 10^4 points sampled per method.

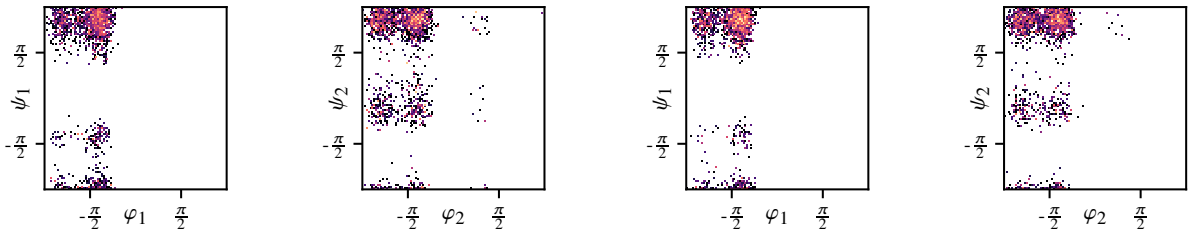


Figure 14. Trialanine Ramachandran plots for ECNF++ (SNIS) and SBG (SMC). 10^4 points sampled per method.

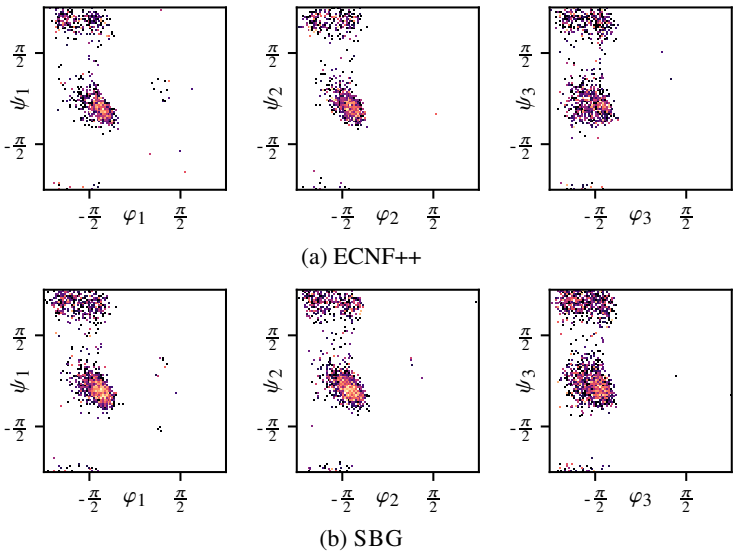


Figure 15. Alanine tetrapeptide Ramachandran plots for ECNF++ (SNIS) and SBG (SMC). 10^4 points sampled per method.

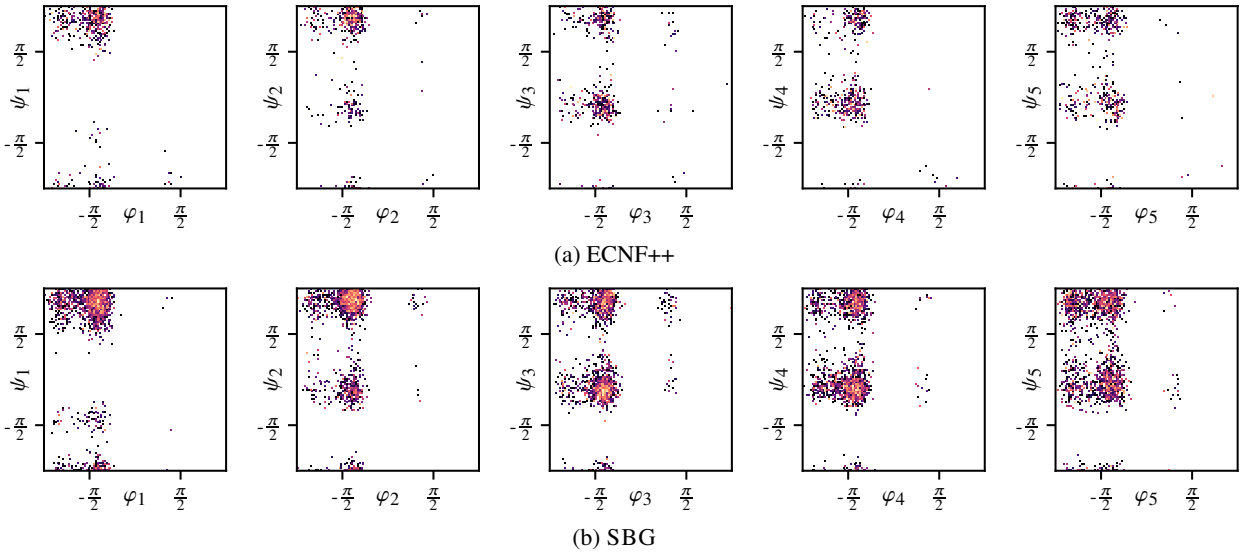


Figure 16. Hexa-alanine Ramachandran plots for ECNF++ (SNIS) and SBG (SMC). 10^4 points sampled per method.

F.3. Ablation Studies

Ablation of SNIS and SMC. In Tables 9 and 10 we compare the quantitative performance of ECNF++ and SBG as proposals, with SNIS, and in the case of SBG, with SMC. This table also presents the TICA-$\mathcal{W}_2$ results. We almost uniformly observe a significant reduction in $\mathcal{E}$-$\mathcal{W}_2$ using SNIS or SMC over the raw proposals, with only ECNF++ SNIS alanine tetrapeptide failing to achieve this. On alanine dipeptide we also observe a large decrease in macrostructure metrics $\mathbb{T}$-$\mathcal{W}_2$ and TICA-$\mathcal{W}_2$, which is expected as the training data and subsequently the proposal distribution was intentionally biased to provide improved coverage (Klein & Noé, 2024). On all other datasets (and both models) we generally see no change or an increase in these metrics after reweighting with either SNIS or SMC, suggesting there is some tradeoff between matching the energy distribution whilst maintaining good macrostructure metrics. We lastly note that the use of SMC over SNIS does not uniformly improve performance on $\mathcal{E}$-$\mathcal{W}_2$ for SBG, with only alanine dipeptide and trialanine exhibiting this trend. However, we believe this may be an artifact of the good proposal overlap, and draw attention to the significant reduction in $\mathcal{E}$-$\mathcal{W}_2$ achieved on chignolin by SMC over SNIS in Figure 8.

Table 9. Comparison of proposal, SNIS, and SMC for SBG and ECNF++ on alanine dipeptide and trialanine.

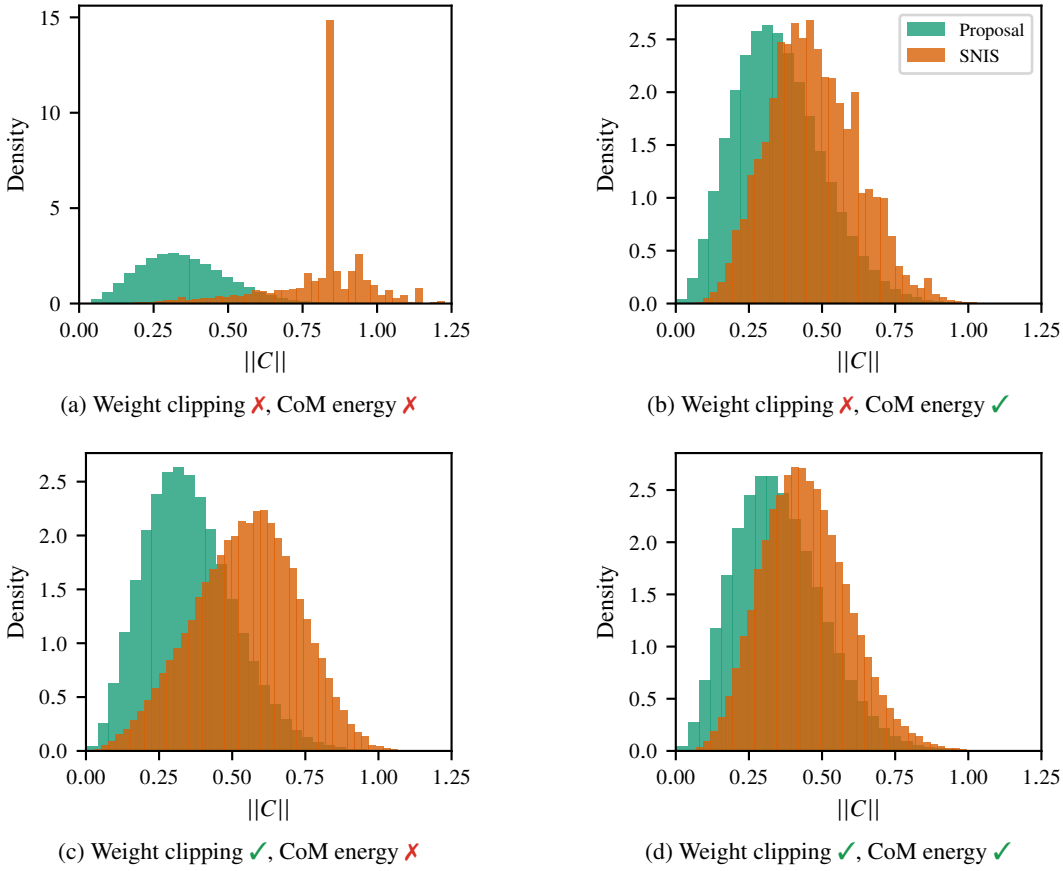
Datasets →	Alanine dipeptide			Trialanine		
Algorithm ↓	$\mathcal{E}$-$\mathcal{W}_2$	$\mathbb{T}$-$\mathcal{W}_2$	TICA-$\mathcal{W}_2$	$\mathcal{E}$-$\mathcal{W}_2$	$\mathbb{T}$-$\mathcal{W}_2$	TICA-$\mathcal{W}_2$
ECNF++ Proposal	6.675 ± 0.297	1.776 ± 0.018	3.920 ± 0.025	5.424 ± 1.595	0.277 ± 0.004	0.435 ± 0.009
ECNF++ SNIS	0.914 ± 0.122	0.189 ± 0.019	0.402 ± 0.002	2.206 ± 0.813	0.962 ± 0.253	0.597 ± 0.023
SBG Proposal	$\geq 10^4$	1.695 ± 0.015	3.862 ± 0.038	$\geq 10^8$	0.338 ± 0.036	0.449 ± 0.028
SBG SNIS	0.873 ± 0.338	0.439 ± 0.129	0.942 ± 0.268	0.758 ± 0.506	0.502 ± 0.016	0.518 ± 0.032
SBG AIS	0.960 ± 0.617	0.430 ± 0.034	0.806 ± 0.166	0.754 ± 0.230	0.495 ± 0.033	0.476 ± 0.048
SBG SMC	0.741 ± 0.189	0.431 ± 0.141	0.915 ± 0.316	0.598 ± 0.084	0.503 ± 0.029	0.501 ± 0.031

Table 10. Comparison of proposal, SNIS, and SMC for SBG and ECNF++ on alanine tetrapeptide and hexa-alanine.

Datasets →	Alanine tetrapeptide			Hexa-alanine		
Algorithm ↓	$\mathcal{E}$-$\mathcal{W}_2$	$\mathbb{T}$-$\mathcal{W}_2$	TICA-$\mathcal{W}_2$	$\mathcal{E}$-$\mathcal{W}_2$	$\mathbb{T}$-$\mathcal{W}_2$	TICA-$\mathcal{W}_2$
ENCF++ Proposal	2.983 ± 1.266	0.576 ± 0.002	0.737 ± 0.013	$\geq 10^4$	1.136 ± 0.030	0.688 ± 0.066
ECNF++ SNIS	5.638 ± 0.483	1.002 ± 0.061	0.832 ± 0.021	10.668 ± 0.285	1.902 ± 0.055	0.632 ± 0.087
SBG Proposal	$\geq 10^6$	0.624 ± 0.023	0.791 ± 0.050	$\geq 10^{12}$	1.079 ± 0.153	0.299 ± 0.039
SBG SNIS	1.068 ± 0.495	0.969 ± 0.067	0.879 ± 0.047	1.036 ± 0.534	1.473 ± 0.114	0.452 ± 0.245
SBG AIS	1.070 ± 0.272	0.923 ± 0.100	0.920 ± 0.028	1.131 ± 0.384	1.510 ± 0.113	0.492 ± 0.240
SBG SMC	1.007 ± 0.382	1.039 ± 0.069	0.904 ± 0.054	1.155 ± 0.635	1.517 ± 0.118	0.530 ± 0.198

Center of mass adjusted energy. As discussed in §3.1, the SBG proposal distribution is not mean-free due to the CoM data augmentation applied to the training data, with a centroid norm distribution $||C|| \sim \sigma\chi_3$. This can introduce adverse behavior, as the target energy is invariant to $||C||$, and thus the importance weights will depend on $||C||$ of a sample x. Concretely, a low (target) energy sample generated far from the origin (with large $||C||$) will have low likelihood under p_θ but high likelihood under p leading to a very large importance weight.

To provide a visual intuition for this effect, we plot in Figure 17 the centroid norm distribution of the SBG proposal samples before and after SNIS. Here the empirical distribution is generated using 2×10^7 samples, to approximate the asymptotic behavior. In Figure 17a we observe that, even with this extremely large number of samples, without weight clipping or center of mass adjusted energy Equation (5) the $||C||$ distribution is greatly influenced by the reweighting. In this case resampling shifts most density to high $||C||$ samples, where there was very little density prior to reweighting and hence little sample diversity, and a large peak manifests resulting from a single sample with very large importance weight. In contrast, in Figure 17b we see a much smaller change in $||C||$ distribution after reweighting, with no large peak for any given sample, after applying the CoM adjustment. In this case there is no overweighting of high $||C||$ regions with limited sample diversity. Adding weight clipping to the standard proposal energy function (Figure 17c) greatly reduces the change in distribution from reweighting, although the mean remains notably shifted towards higher $||C||$ samples. Applying weight clipping with the CoM adjusted proposal energy function (Figure 17d) has little effect on the $||C||$ distribution beyond smoothing. We



(a) Weight clipping ✗, CoM energy ✗

(b) Weight clipping ✗, CoM energy ✓

(c) Weight clipping ✓, CoM energy ✗

(d) Weight clipping ✓, CoM energy ✓

Figure 17. Centroid norm $||C||$ histograms for 2×10^7 proposal samples and reweighted proposal samples, with / without both of weight clipping (0.2%) and center of mass adjusted energy.

emphasize that these plots are presented for 2×10^7 samples hence clipping may have a larger still effect for both proposal energy functions for smaller sample sets.

In Figures 18 and 19 we ablate the utility of performing the center of mass energy adjustment to the proposal energy. Specifically, we ablate the center of mass adjustment as a function of number of samples used during inference and also as a function of a number of inference time steps. Each of these ablations is performed on trialanine. Considering Figure 18, there is little distinction between variants on $\mathcal{E}$-$\mathcal{W}_2$ with respect to N samples, although standard energy without clipping can be identified as the least performant, and all methods improve with increased N. On $\mathbb{T}$-$\mathcal{W}_2$ the standard energy without clipping is again evidently the worst performing, with a clear benefit to applying the CoM adjustment where clipping is not used. The best performing variants do employ clipping, with a slight but clear benefit to using the center of mass adjustment. Considering Figure 19, we observe again that the center of mass adjustment improves performance greatly where clipping is not employed, and notably still without clipping on $\mathbb{T}$-$\mathcal{W}_2$.

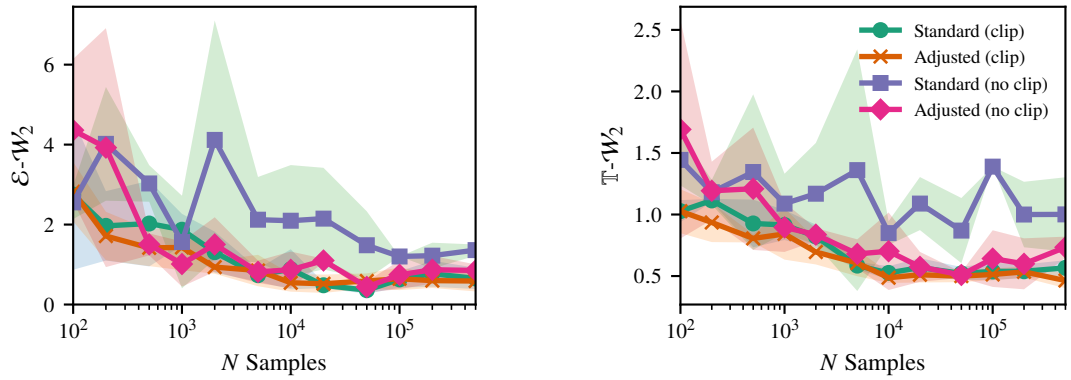


Figure 18. Trialanine SBG SMC $\mathcal{E}$-$\mathcal{W}_2$ and $\mathbb{T}$-$\mathcal{W}_2$ performance with standard and center of mass adjusted energy, with / without weight clipping (0.2%) at a variety of sampling set sizes. 100 Langevin timesteps used.

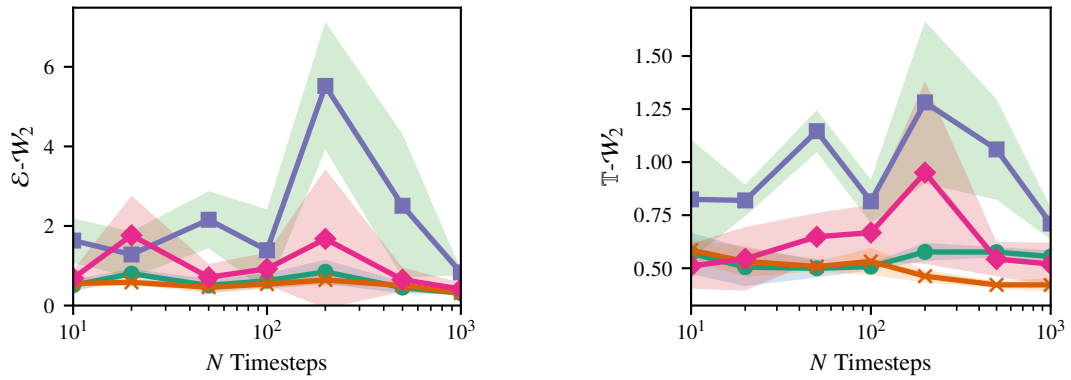
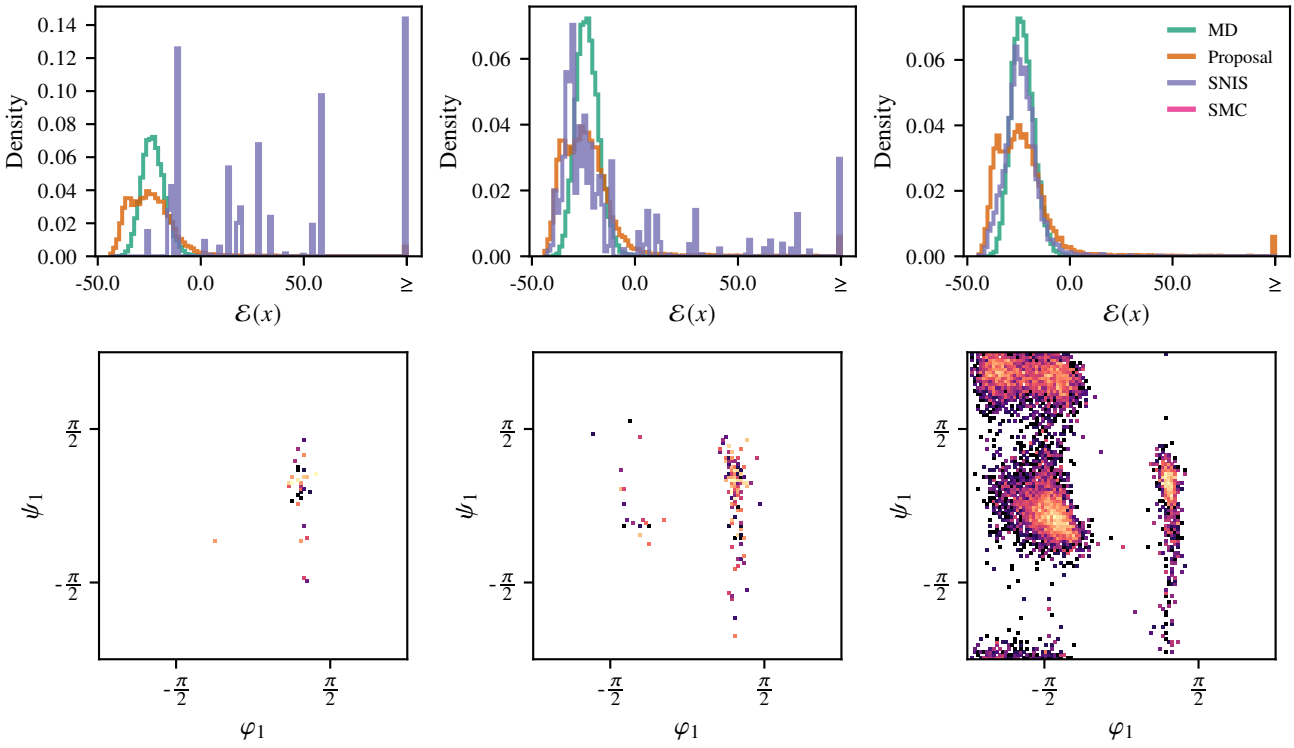


Figure 19. Trialanine SBG SMC $\mathcal{E}$-$\mathcal{W}_2$ and $\mathbb{T}$-$\mathcal{W}_2$ performance with standard and center of mass adjusted energy, with / without weight clipping (0.2%) at a varierty of Langevin time discretizations. 10^4 samples generated.

Ablation on EACF importance weight clipping. We lastly report additional EACF results on alanine dipeptide. In our main results we use the same 0.2% clipping threshold on the importance weights as other models for fair comparison. Nevertheless, in the resampling process, we observe a significant degradation in sample diversity, as evidenced by the energy histogram in Figure 3 and Ramachandran plots Figure 9. In Figure 20 we plot energy histograms and Ramachandran plots for a variety of different clipping thresholds. We observe that EACF generates highly unreliable importance weights, particularly visible in the energy histograms where there are extreme spikes and poor alignment with the true data distribution. This leads to poor resampling quality, as demonstrated in the corresponding Ramachandran plots where the resampled points fail to capture the true data distribution. While increasing the clipping threshold to 10% shows some improvement, the fundamental issue of inaccurate importance weight estimation by EACF persists across different clipping ratios.



(a) Clip 0.2% (b) Clip 2% (c) Clip 10%

Figure 20. Energy histogram and Ramachandran Plots of SE(3)-EACF under different weight clipping ratio $[0.2\%, 2\%, 10\%]$.