SCALABLE EQUILIBRIUM SAMPLING WITH SEQUENTIAL BOLTZMANN GENERATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Scalable sampling of molecular states in thermodynamic equilibrium is a long-standing challenge in statistical physics. Boltzmann generators tackle this problem by pairing powerful normalizing flows (NF) with importance sampling to obtain statistically independent samples under the target distribution. In this paper, we extend the Boltzmann generator framework and introduce SEQUENTIAL BOLTZMANN GENERATORS (SBG) with two key improvements. The first is a highly efficient non-equivariant Transformer-based normalizing flow operating directly on all-atom Cartesian coordinates. The second is inference time scaling of flow samples with non-equilibrium transport towards the target distribution and reweighting through a novel application of Sequential Monte Carlo (SMC). SBG is more computationally efficient as no explicit equivariance constraint is encoded in the NF, but is instead softly introduced via data augmentations. Further, SBG improves sample quality by applying SMC along the transport path. SBG achieves state-of-the-art performance w.r.t. all metrics on molecular systems, demonstrating the first equilibrium sampling in Cartesian coordinates of tri, tetra, and hexapeptides that were so far intractable for prior Boltzmann generators.

1 INTRODUCTION

The simulation 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).

The dominant paradigm for molecular simulation 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 systems often exist in thermodynamic equilibrium by remaining for long time horizons in metastable states before rapidly transitioning to another metastable state. Such states are captured in the minima of a complex energy landscape, associated with the molecular system's equilibrium (Boltzmann) distribution at a given temperature. Unfortunately, drawing uncorrelated samples from such metastable states via traditional MD or MCMC methods is prohibitively computationally expensive, requiring long simulation steps with small updates on the order of femtoseconds 1fs $= 10^{-15}$s, as transitions are rare events due to the presence of high-energy barriers between well-separated states (Wirnsberger et al., 2020).

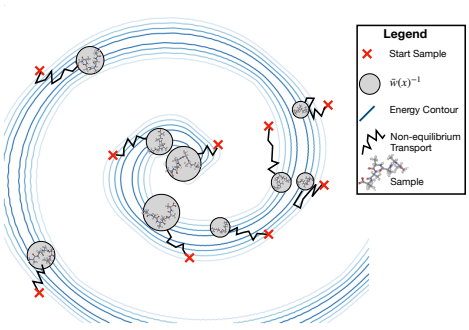


Figure 1: SBG uses inference time non-equilibrium transport to move initial proposal samples from a normalizing flow.

An alternative approach is to enhance sampling efficiency by leveraging powerful generative models such as normalizing flows (Dinh et al., 2017; Rezende & Mohamed, 2015) trained on existing biased datasets, to produce approximate samples which can then be reweighted via importance sampling to follow the desired Boltzmann distribution. Such models, called *Boltzmann generators* (BG) (Noé

et al., 2019), allow faster sampling through amortization as generation is significantly cheaper computationally than running MD or MCMC. Despite their appeal, it remains challenging for existing BGs to generate uncorrelated samples in their native Cartesian coordinates from the energy modes of the Boltzmann distribution for larger molecular systems at the scale of small peptides (2 amino acids) (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 $\mathrm{E}(n)$-GNN's (Satorras et al., 2021) based equivariant vector fields used in the design of continuous-time normalizing flows (Chen et al., 2018). As a result, even the most performant BGs suffer from low overlap with the target Boltzmann distribution, leading to poor sampling efficiency during importance sampling.

Present work. In this paper, we introduce SEQUENTIAL BOLTZMANN GENERATORS (SBG) a novel extension to the existing Boltzmann generator framework. SBG makes progress on the scalability of Boltzmann generators in Cartesian coordinates along two complementary axes: (1) scalable pre-training of softly $\mathrm{SE}(3)$-equivariant proposal normalizing flows in BGs; and (2) inference time scaling of proposal flow samples and their importance weights under fast non-equilibrium processes, e.g. such as Langevin dynamics. The final result yields higher quality generated samples that *require dramatically less correction* through reweighting and thus allowing for the computation of important observable quantities such as free energy differences between metastable states of $\mu_{\text{target}}(x)$.

Our proposed approach SBG scales up proposal normalizing flows in BG's by following recent advances in atomistic generative modeling, e.g. AlphaFold 3 (Abramson et al., 2024). In particular, we opt to remove the rigid $\mathrm{SE}(3)$-equivariance as an explicit architectural inductive bias in favor of softly enforcing it through simpler and more efficient data augmentation. To further improve samples and their importance weights—a crucial step in the real-world application of BGs—we perform inference scaling by designing a target-informed non-equilibrium process. More precisely, we define 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 the transport dynamics can be coupled to an equivalent time evolution of importance weights—*without the need to compute the original numerically unfavorable importance weights*—themselves, converting naturally to the well-established technique of Sequential Monte Carlo (SMC) in continuous time. As a result, SBG can easily improve over the simple one-step importance sampling methodology used in existing BGs. We summarize the different aspects of our proposed SBG in comparison to other learned samplers and Boltzmann generators in Table 1.

Table 1: Overview of the properties of models for sampling from target distributions with (possibly biased data).

Method	Exact Likelihoods	Use Data	Use $\mathcal{E}(x)$	Transport
NETS (Albergo & Vanden-Eijnden, 2024)	✗	✗	✓	✓
DEM (Akhound-Sadegh et al., 2024)	✗	✗	✓	✗
BGs (Noé et al., 2019)	✓	✓	✓	✗
Continuous BGs Köhler et al. (2020)	✓	✓	✓	✗
SBG (ours)	✓	✓	✓	✓

We instantiate SBG using exactly invertible architectures by utilizing a modernized *non-equivariant* Transformer architecture as the backbone and use best-in-class models in TarFlow (Zhai et al., 2024). We demonstrate that exactly invertible architectures, because of fast and exact log-likelihood computations, benefit from inference-scaling. We emphasize this is in stark contrast to continuous normalizing flows that power prior SOTA Boltzman generators which require both simulation of the 2nd order divergence operator and differentiating through an ODE solver. Furthermore, we demonstrate that enforcing equivariance softly along with appropriate normalization strategies enables us to stably scale the size of proposal flows in SBG. On a theoretical front, we study the added bias of common numerical tricks in the literature such as thresholding, and propose an automatic scheme to find the optimal thresholding parameter. Empirically, we observe SBG achieve state-of-the-art results across all metrics, and due to the enhanced computational efficiency far outperform continuous BG's on all datasets. In particular, SBG is the first method to solve tripeptides, tetrapeptides, hexapeptides, and makes progress towards equilibrium sampling of decapeptides in Cartesian coordinates while past BG methods were intractable beyond dipeptides.

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 $\mathbb{R}^{n\times 3}$:

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

The Boltzmann distribution is defined for a system and includes the Boltzmann constant k_{B}, and is specified for a given 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 $\mathcal{Z}$ associated to the Boltzmann distribution in general is intractable to evaluate.

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 $\mathbb{D}_{\mathrm{KL}}(p_{\mathcal{D}}||p_\theta)$—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,

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

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.

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 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 importance weights for each sample $w(x^i) = \exp\left(\frac{-\mathcal{E}(x^i)}{k_B T}\right)/p_\theta(x^i)$. Leveraging the collection of importance weights we can compute a Monte-Carlo approximation to any test function $\phi(x)$ 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 adding a non-equilibrium transport method that leads to higher-quality samples and better importance weights. We begin by identifying the key limitation in current BG's as importance sampling with a suboptimal proposal. Indeed, while the self-normalized importance sampling estimator is consistent, its' fidelity is highly dependent on the quality of the actual proposal p_θ. In fact, the optimal proposal distribution 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 framework is trained on a biased dataset $\mathcal{D}$ the importance weights computed typically exhibit large variance—resulting in a small effective sample size (ESS).

We address the need for more flexible proposals in §3.1 with modernized scalable training recipes for normalizing flows. In §3.2 we outline our novel application of non-equilibrium processes and Annealed Importance Sampling that powers our inference scaling algorithm that drives proposal samples and their importance weight towards the metastable states of $\mathcal{E}(x)$. 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 known to be 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 BG's by departing from exactly equivariant maps by instead considering soft-equivariance which opens up the usage of more 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 Blockwise 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 a standard deviation of one. 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 $(n-1)\times d$ 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-equivariant prior distribution such as the standard normal $p_0 = \mathcal{N}(0, I)$. The next proposition outlines the family of permissible noise.

Proposition 1. *Given an* $\mathrm{SE}(3)$*-invariant* $\mu_{target}(x)$ *and the noise-adjusted distribution* $\mu'_{target}(x)$*. Consider the decomposition of a data sample into its constituent mean-free component,* $\tilde{x}$ *and center of mass* $c \in \mathbb{R}^3$*,* $x = \tilde{x} + c$*, where* $c \sim \mu(c)$ *and* $\mu(c)$ *is* $\mathrm{SO}(3)$*-invariant. Then* $\mu_{target}(\tilde{x}) = \mu'_{target}(\tilde{x})$ *if* $\mu'_{target}(x) = \mu(\tilde{x})\mu(\|c\|)$.

We prove Proposition 1 in §B.1, which tells us that any noise distribution that acts on the norm of the center of mass does not operationally change the target. As a result, we choose to add small amounts of Gaussian noise $c \sim \mathcal{N}(0, \sigma)$ to the center of mass of a given data sample. The impact of this noise is that during reweighting we must account for $\mu(\|c\|)$ which follows a $\chi(3)$ distribution. Consequently, we must adjust the model energy to account for the impact of CoM noise during reweighting as follows:

$$\log p^c_\theta(x) = \log p_\theta(x) - \left(\log\left(\frac{\|c^2\|}{\sigma^3}\right) + \frac{\|c\|^2}{2\sigma^2} + C\right),$$

where $C = -\log\left(\sqrt{2}\Gamma\left(\frac{3}{2}\right)\right)$ and Γ is the gamma function.

3.2 INFERENCE TIME SCALING OF BOLTZMANN GENERATORS

Given a trained BG with proposal flow p_θ, the simple 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}.

Improving samples through non-equilibrium transport. 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 Langevin dynamics with reweighting through a continuous-time variant of Annealed Importance Sampling (AIS). Concretely, we consider the following SDE that

drives proposal samples towards metastable states of the Boltzmann target:

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

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, 2024) 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 often no need for *additional learning* during this step which we view as extending the inference capabilities of the original Boltzmann generator $\mu_\theta(x)$.

To compute test functions for the transported samples, and thus reweighting, we use a well-known and celebrated result known as Jarzynski's equality that enables the calculation of equilibrium statistics from non-equilibrium processes. We recall the main result, originally derived in Vaikuntanathan & Jarzynski (2008), and recently re-derived in continuous-time in the context of learning to sample by Vargas et al. (2024); Albergo & Vanden-Eijnden (2024) that makes explicit the time evolution of the new importance weights.

Proposition 2 (Albergo & Vanden-Eijnden (2024))**.** *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

$$\int_{\mathbb{R}^d} \phi(x) p_\tau(x) dx = \frac{\mathbb{E}[w_\tau \phi(x_\tau)]}{\mathbb{E}[w_\tau]} \tag{3}$$

and

$$\mathcal{Z}_\tau / \mathcal{Z}_1 = \mathbb{E}[e^{w_\tau}] \quad (\textit{Jarzynski's Equality}) \tag{4}$$

The final samples $x_{\tau=1}$ can then be reweighted according to final importance weights $w_{\tau=1}$ that have lower magnitudes than simple importance sampling in conventional BG's. It is crucial to highlight that through inference-time scaling, we never need to compute the high-magnitude importance weights under the prior $p_0(x_0)$, and instead the proposal $p_\theta(x_0)$ acts as a new prior for the Langevin process. It is precisely this learned proposal distribution that $d\log w_\tau$ accounts for within the parlance of Annealed Importance Sampling. To evolve the Langevin SDE we require,

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

which requires efficient gradient computation through the log-likelihood estimation under the normalizing flow p_θ. This presents the first point of distinction between finite flows and CNF's. The former class of flows trained using Eq. 1 gives fast exact likelihoods—especially for our scalable non-equivariant TarFlow model. In contrast, CNF's must simulate and differentiate through an ODE solver to compute $\nabla \log p_\theta(x_\tau)$ for each step of the Langevin SDE in Eq. 2. 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.

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**
  $x_{\tau+\Delta} \leftarrow x_\tau - \epsilon_\tau \nabla \mathcal{E}_\tau(x_\tau) d\tau + \sqrt{2\epsilon_\tau} dW_\tau$
  $\log w_{\tau+\Delta} = \log w_\tau - \partial_\tau \mathcal{E}(x_\tau) d\tau$
  $\tau \leftarrow \tau + \Delta$
  if ESS $<$ ESS$_{\text{threshold}}$ **then**
    $x_\tau \leftarrow \text{RESAMPLE}(x_\tau, w_\tau)$
    $w_\tau \leftarrow 0$
  end if
end for

To further enable a reduced 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. 7 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,

$$\log \mathcal{Z} = \log \mathbb{E}_{x\sim p_\theta(x)} \left[\frac{\exp\left(\frac{-\mathcal{E}(x)}{k_B T}\right)}{p_\theta(x)} \right] \geq \mathbb{E}_{x\sim p_\theta(x)} \left[\frac{-\mathcal{E}(x)}{k_B T} - \log p_\theta(x) \right] = \log \hat{\mathcal{Z}}. \tag{5}$$

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 Appendix 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:*

$$\gamma \geq \frac{1}{\lambda} \log \left(\frac{Kb}{12\rho \mathbb{E}[\exp(-\lambda X)]} \right) + \log \hat{\mathcal{Z}}. \tag{6}$$

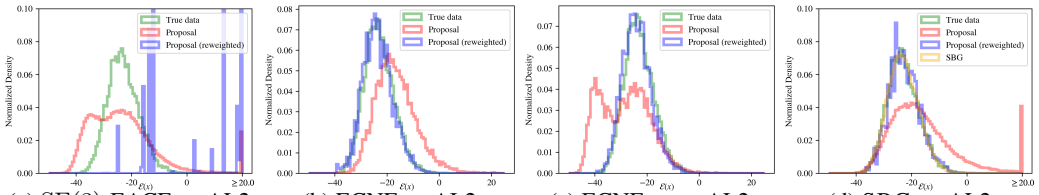
The proof for Proposition 3 is located in §B.3. Proposition 3 allows us to appropriately set a energy threshold γ as a function of tolerance ρ that depends on ESS. In practice, this allows us to negotiate the amount of acceptable bias when dropping initial samples that obtain high-energy before any further AIS correction. Moreover, this gives a firmer theoretical foundation to existing practices of thresholding high-energy 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 force-fields as the energy function with exact experimental setups described in §D. To generate samples and their corresponding weights we follow Algorithm 1 with resampling (lines 6-9) which is run on initial proposal samples and is equivalent to performing SMC (Doucet et al., 2001).

Datasets. We consider small peptides composed of varying numbers of alanine amino acids, with some systems additionally incorporating an acetyl group and an N-methyl group. We investigate alaine systems of up to 6 amino acids. All datasets are generated from a single MD simulation in implicit solvent using a classical force field. For each system, the first 1ns is used for training, the next 0.2ns 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 significantly larger protein Chignolin, consisting of 10 amino acids generated with the Anton supercomputer in Lindorff-Larsen et al. (2011). We provide additional dataset details in §E.



(a) SE(3)-EACF on AL2 (b) ECNF on AL2 (c) ECNF++ on AL2 (d) SBG on AL2

Figure 2: Energies of samples generated with different methods on alanine dipeptide (ALDP).

Baselines. For baselines, we train prior state-of-the-art equivariant Boltzmann generators. Specifically, we use SE(3)-augmented coupling flow (Midgley et al., 2023a) as the exactly invertible and equivariant architecture and the equivariant ECNF employed in Transferrable Boltzmann Generators (Klein & Noé, 2024). We also include an improved equivariant CNF (ECNF++) as a stronger baseline, see §D.3 for full details, which uses improved flow matching loss, improved data normalization, a larger network, improved learning rate schedule, and optimizer.

Metrics. We report interatomic distances as a normalized density between the ground truth data, and initial proposal samples, as well as the energy histogram of the system for ground truth, initial proposal samples, transported samples, and the reweighted energy histogram. We also include Ramachandran plots Ramachandran et al. (1963) for each molecular system studied that visualizes dihedral angles' distribution for the ground truth data distribution and the generated samples. We include additional quantitative metrics that provide a finer grained evaluation of each method. Concretely, we compute the ESS, Wasserstein-1 distance on the energy distribution, and the Wasserstein-2 distance of the dihedral angles used in the Ramachandran plot.

4.1 Results

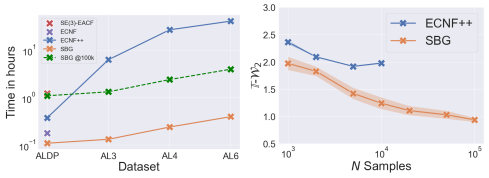
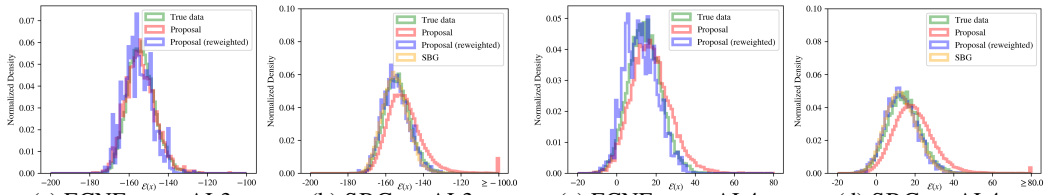


Figure 3: **Left:** Time in hours for sampling and reweighing 10k points. **Right:** $\mathbb{T}$-$\mathcal{W}_2$ on AL3 as a function of inference samples.

We evaluate SBG and our chosen baselines on alanine dipeptide (ALDP), trialanine (AL3), alanine tetrapeptide (AL4), and hexaalanine (AL6) with quantitative metrics summarized in Table 2 and Table 3. In SBG @ 10k we generate $10k$ samples and directly report metrics on these samples. In SBG @100k we generate $100k$ samples and subsample to $10k$ after SMC to compute directly comparable metrics. For SE(3)-EACF we retrain this baseline on our more challenging version of ALDP and observe that performance degrades substantially at the selected 0.2% weight clipping threshold (c.f. §F for higher clipping thresholds). Furthermore, we find that on ALDP, our improved ECNF++ baseline obtains a 177% relative improvement in ESS over the previous SOTA ECNF from Klein & Noé (2024). Importantly, we observe SBG is the best method on the Wasserstein-1 energy distance $\mathcal{E}$-$\mathcal{W}_1$ and Wasserstein-2 distance on dihedreal angles $\mathbb{T}$-$\mathcal{W}_2$. As SBG involves resamples on a finite set of points, we observe that the higher number of particles (100k) results in consistently improved $\mathcal{E}$-$\mathcal{W}_1$ and $\mathbb{T}$-$\mathcal{W}_2$. These results are further substantiated in Figure 2 which depicts the energy histograms of SBG in relation to the ground truth energy of the system and depicts a near perfect overlap.

For tripeptides, tetrapeptides, and hexapeptides we remark that the SE(3)-EACF baseline is too computationally expensive and thus does not scale (c.f. Table 4). Consequently, we report metrics for our improved ECNF, ECNF++, and SBG. We observe that ECNF fails to learn effectively on the tri and tetrapeptides with $\mathcal{E}$-$\mathcal{W}_1$ exploding over 10^4, while our improved ECNF++ is orders of magnitude better. We highlight that SBG is the best method across all metrics, and in particular, we highlight that the improvements are more prominently driven by inference time scaling of proposal samples as observed in Section 4.1, Figure 4, and Figure 5. 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 BG's. We also report in §F.1 the Ramachandran plots for each method. Finally, we include additional ablations such as the utility of CoM augmentation in Appendix F.

Inference scaling. To illustrate the scalability of SBG in relation to other methods we plot in Figure 3 the log-scale inference time for each dataset. In particular, for inference, we include the time to generate and reweight $10k$ samples. We observe an almost exponential scaling of ECNF as the size of the peptide grows, while SBG is dramatically faster at inference. As an ablation, we also



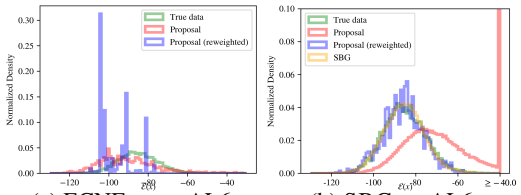
(a) ECNF++ on AL3 (b) SBG on AL3 (c) ECNF++ on AL4 (d) SBG on AL4

Figure 4: Energy plots for trialanine (AL3) and tetrapeptide (AL4).

plot $\mathbb{T}$-$\mathcal{W}_2$ metric on AL3 as a function of inference samples which shows a monotonic decrease as the number of samples increase, which is computationally tractable due to cheap inference of SBG.

Table 3: Results on trialanine, alanine tetrapeptide, and hexapeptide. *Indicates ESS after resampling.

Datasets →	Tripeptide (AL3)			Tetrapeptide (AL4)			Hexapeptide (AL6)		
Algorithm ↓	ESS ↑	$\mathcal{E}$-$\mathcal{W}_1$ ↓	$\mathbb{T}$-$\mathcal{W}_2$ ↓	ESS ↑	$\mathcal{E}$-$\mathcal{W}_1$ ↓	$\mathbb{T}$-$\mathcal{W}_2$ ↓	ESS ↑	$\mathcal{E}$-$\mathcal{W}_1$↓	$\mathbb{T}$-$\mathcal{W}_2$ ↓
ECNF	$< 10^{-4}$	$> 10^4$	7.010	$< 10^{-4}$	$> 10^4$	3.853	-	-	-
ECNF++ (Ours)	0.036 ± 0.027	1.759 ± 0.788	1.967 ± 0.062	0.123 ± 0.006	4.229 ± 1.284	2.414 ± 0.000	0.015 ± 0.003	8.954 ± 0.646	5.405 ± 0.069
SBG (Ours) @10k	0.732* ± 0.189	1.676 ± 0.138	1.244 ± 0.108	0.898* ± 0.072	2.155 ± 0.066	2.099 ± 0.004	0.989* ± 0.014	1.573 ± 0.464	3.785 ± 0.140
SBG (Ours) @100k	0.882* ± 0.193	**1.384 ± 0.245**	**0.940 ± 0.040**	0.890* ± 0.072	**1.837± 0.377**	**1.804 ± 0.022**	0.940* ± 0.048	**0.474 ± 0.141**	**3.303 ± 0.078**



(a) ECNF++ on AL6 (b) SBG on AL6

Figure 5: Energy plots for hexapeptide (AL6).

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 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), which often includes coarse-graining. 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., 2024; Jing et al., 2024b; Daigavane et al., 2024).

Table 2: Results on ALDP. *Indicates resampled ESS.

Datasets →	Alanine dipeptide (ALDP)		
Algorithm ↓	ESS ↑	$\mathcal{E}$-$\mathcal{W}_1$ ↓	$\mathbb{T}$-$\mathcal{W}_2$ ↓
SE(3)-EACF	$< 10^{-4}$	14.70	1.738
ECNF	0.084	0.984	0.391
ECNF++ (Ours)	0.233 ± 0.042	0.825 ± 0.038	0.349 ± 0.050
SBG (Ours) @10k	0.893* ± 0.167	**0.571 ± 0.451**	0.476 ± 0.010
SBG (Ours) @100k	0.880* ± 0.177	**0.394 ± 0.298**	**0.306 ± 0.025**

6 Conclusion

In this paper, we introduce SBG an extension to the Boltzmann generator framework that scales inference through the use of non-equilibrium transport. Unlike past BG's 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 transport 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 orf future work.

REFERENCES

Osama Abdin and Philip M Kim. Pepflow: direct conformational sampling from peptide energy landscapes through hypernetwork-conditioned diffusion. *bioRxiv*, pp. 2023–06, 2023. (Cited on page 8)

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pp. 1–3, 2024. (Cited on page 2)

Sergios Agapiou, Omiros Papaspiliopoulos, Daniel Sanz-Alonso, and Andrew M Stuart. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, pp. 405–431, 2017. (Cited on page 14)

Tara Akhound-Sadegh, Jarrid Rector-Brooks, Joey Bose, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcin Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, Nikolay Malkin, and Alexander Tong. Iterated denoising energy matching for sampling from boltzmann densities. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 760–786. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/akhound-sadegh24a.html. (Cited on page 2)

Michael S. Albergo and Eric Vanden-Eijnden. Nets: A non-equilibrium transport sampler, 2024. URL https://arxiv.org/abs/2410.02711. (Cited on pages 2, 5, and 13)

Dosovitskiy Alexey. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the 9th International Conference on Learning Representations*, 2021. (Cited on page 4)

Avishek Joey Bose, Marcus Brubaker, and Ivan Kobyzev. Equivariant finite normalizing flows. *arXiv preprint arXiv:2110.08649*, 2021. (Cited on pages 2 and 4)

Ignasi Buch, Toni Giorgino, and Gianni De Fabritiis. 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)

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Neural Information Processing Systems (NIPS)*, 2018. (Cited on page 2)

Ameya Daigavane, Bodhi P Vani, Saeed Saremi, Joseph Kleinhenz, and Joshua Rackers. Jamun: Transferable molecular conformational ensemble generation with walk-jump sampling. *arXiv preprint arXiv:2410.14621*, 2024. (Cited on page 8)

Volker L Deringer. 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. URL https://dx.doi.org/10.1088/2515-7655/abb011. (Cited on page 1)

Manuel Dibak, Leon Klein, Andreas Krämer, and Frank Noé. Temperature steerable flows and Boltzmann generators. *Phys. Rev. Res.*, 4:L042005, Oct 2022. doi: 10.1103/PhysRevResearch.4.L042005. (Cited on pages 8 and 24)

Juan Viguera Diez, Mathias Schreiner, Ola Engkvist, and Simon Olsson. Boltzmann priors for implicit transfer operators. *arXiv preprint arXiv:2410.10605*, 2024. (Cited on page 8)

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

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024. (Cited on page 16)

Arnaud Doucet, Nando De Freitas, Neil James Gordon, et al. *Sequential Monte Carlo methods in practice*, volume 1. Springer, 2001. (Cited on page 6)

Peter Eastman, Jason Swails, John D Chodera, Robert T McGibbon, Yutong Zhao, Kyle A Beauchamp, Lee-Ping Wang, Andrew C Simmonett, Matthew P Harrigan, Chaya D Stern, et al. Openmm 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS computational biology*, 13(7):e1005659, 2017. (Cited on page 21)

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. URL https://arxiv.org/abs/2403.03206. (Cited on page 20)

Xiang Fu, Tian Xie, Nathan J Rebello, Bradley Olsen, and Tommi S Jaakkola. Simulate time-integrated coarse-grained molecular dynamics with multi-scale graph networks. *Transactions on Machine Learning Research*, 2023. (Cited on page 8)

Victor Garcia Satorras, Emiel Hoogeboom, Fabian Fuchs, Ingmar Posner, and Max Welling. E(n) equivariant normalizing flows. *Neural Information Processing Systems (NeurIPS)*, 2021. (Cited on page 4)

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: free-form continuous dynamics for scalable reversible generative models. *ICLR*, 2019. (Cited on page 20)

M.F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 19(2):433–450, 1990. doi: 10.1080/03610919008812866. URL https://doi.org/10.1080/03610919008812866. (Cited on page 20)

Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional diffusion for molecular conformer generation. *Advances in Neural Information Processing Systems*, 35:24240–24253, 2022. (Cited on page 8)

Bowen Jing, Bonnie Berger, and Tommi Jaakkola. Alphafold meets flow matching for generating protein ensembles. *arXiv preprint arXiv:2402.04845*, 2024a. (Cited on page 8)

Bowen Jing, Hannes Stärk, Tommi Jaakkola, and Bonnie Berger. Generative modeling of molecular dynamics trajectories. *arXiv preprint arXiv:2409.17808*, 2024b. (Cited on page 8)

Rafał Karczewski, Markus Heinonen, and Vikas Garg. Diffusion models as cartoonists! the curious case of high density regions. *arXiv preprint arXiv:2411.01293*, 2024. (Cited on pages 13, 17, and 18)

Leon Klein and Frank Noé. Transferable boltzmann generators. In *Advances in Neural Information Processing Systems*, 2024. (Cited on pages 6, 7, 8, 19, and 21)

Leon Klein, Andrew YK Foong, Tor Erlend Fjelde, Bruno Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noé, and Ryota Tomioka. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics. *Neural Information Processing Systems (NeurIPS)*, 2023a. (Cited on pages 8 and 21)

Leon Klein, Andreas Krämer, and Frank Noé. Equivariant flow matching. *Neural Information Processing Systems (NeurIPS)*, 2023b. (Cited on pages 2 and 8)

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. *International Conference on Machine Learning (ICML)*, 2020. (Cited on pages 2 and 4)

Jonas Köhler, Andreas Krämer, and Frank Noé. Smooth normalizing flows. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 2796–2809. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/167434fa6219316417cd4160c0c5e7d2-Paper.pdf. (Cited on page 8)

Jonas Köhler, Michele Invernizzi, Pim De Haan, and Frank Noé. Rigid body flows for sampling molecular crystal structures. *International Conference on Machine Learning (ICML)*, 2023. (Cited on page 8)

Holden Lee, Chirag Pabbaraju, Anish Prasad Sevekari, and Andrej Risteski. Universal approximation using well-conditioned normalizing flows. *Advances in Neural Information Processing Systems*, 34:12700–12711, 2021. (Cited on page 4)

Sarah Lewis, Tim Hempel, José Jiménez Luna, Michael Gastegger, Yu Xie, Andrew YK Foong, Victor García Satorras, Osama Abdin, Bastiaan S Veeling, Iryna Zaporozhets, et al. Scalable emulation of protein equilibrium ensembles with generative deep learning. *bioRxiv*, pp. 2024–12, 2024. (Cited on page 8)

Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, and David E Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011. (Cited on pages 1, 6, and 24)

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation, 2024. URL https://arxiv.org/abs/2309.06380. (Cited on page 20)

I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. (Cited on page 21)

Bálint Máté and François Fleuret. Learning interpolations between boltzmann densities. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=TH6YrEcbth. (Cited on page 5)

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

Laurence I Midgley, Vincent Stimper, Javier Antorán, Emile Mathieu, Bernhard Schölkopf, and José Miguel Hernández-Lobato. SE(3) equivariant augmented coupling flows. *Neural Information Processing Systems (NeurIPS)*, 2023a. (Cited on pages 2, 6, 7, 8, and 19)

Laurence Illing Midgley, Vincent Stimper, Gregor NC Simm, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Flow annealed importance sampling bootstrap. *International Conference on Learning Representations (ICLR)*, 2023b. (Cited on pages 6 and 8)

Frank Noé, Christof Schütte, Eric Vanden-Eijnden, Lothar Reich, and Thomas R Weikl. 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)

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019. (Cited on pages 1, 2, 3, and 8)

Art B. Owen. *Monte Carlo theory, methods and examples*. https://artowen.su.domains/mc/, 2013. (Cited on page 3)

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

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

G N Ramachandran, C Ramakrishnan, and V Sasisekharan. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, pp. 95–99, 1963. (Cited on page 7)

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

Andrea Rizzi, Paolo Carloni, and Michele Parrinello. Multimap targeted free energy estimation. *arXiv preprint arXiv:2302.07683*, 2023. (Cited on page 8)

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. *International Conference on Machine Learning (ICML)*, 2021. (Cited on page 2)

Maximilian Schebek, Michele Invernizzi, Frank Noé, and Jutta Rogal. Efficient mapping of phase diagrams with conditional boltzmann generators. *Machine Learning: Science and Technology*, 2024. (Cited on page 8)

Mathias Schreiner, Ole Winther, and Simon Olsson. Implicit transfer operator learning: Multiple time-resolution models for molecular dynamics. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=1kZx7JiuA2. (Cited on page 8)

Marta Skreta, Lazar Atanackovic, Avishek Joey Bose, Alexander Tong, and Kirill Neklyudov. The superposition of diffusion models using the it\^ o density estimator. *arXiv preprint arXiv:2412.17762*, 2024. (Cited on pages 13 and 17)

Takeshi Teshima, Isao Ishikawa, Koichi Tojo, Kenta Oono, Masahiro Ikeda, and Masashi Sugiyama. Coupling-based invertible neural networks are universal diffeomorphism approximators. *Advances in Neural Information Processing Systems*, 33:3362–3373, 2020. (Cited on page 4)

Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023. (Cited on page 19)

Suriyanarayanan Vaikuntanathan and Christopher Jarzynski. Escorted free energy simulations: Improving convergence by reducing dissipation. *Phys. Rev. Lett.*, 100:190601, May 2008. doi: 10.1103/PhysRevLett.100.190601. URL https://link.aps.org/doi/10.1103/PhysRevLett.100.190601. (Cited on page 5)

Francisco Vargas, Shreyas Padhy, Denis Blessing, and Nikolas Nüsken. Transport meets variational inference: Controlled Monte Carlo diffusions. *International Conference on Learning Representations (ICLR)*, 2024. (Cited on page 5)

Peter Wirnsberger, Andrew J Ballard, George Papamakarios, Stuart Abercrombie, Sébastien Racanière, Alexander Pritzel, Danilo Jimenez Rezende, and Charles Blundell. Targeted free energy estimation via learned mappings. *The Journal of Chemical Physics*, 153(14):144112, 2020. (Cited on pages 1 and 8)

Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Navdeep Jaitly, and Josh Susskind. Normalizing flows are capable generative models. *arXiv preprint arXiv:2412.06329*, 2024. (Cited on pages 2, 4, and 21)

A Alternate Paths

A.1 Proposal free Langevin dynamics

We can also modify the Langevin SDE in Eq. 2 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, 2024, 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, 2024) 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:

$$\begin{aligned} dx_\tau &= -\epsilon_\tau \nabla \mathcal{E}_t(x_\tau) d\tau + \nu_\tau(x_\tau) d\tau + \sqrt{2\epsilon_\tau} dW_\tau \\ &= -\epsilon_\tau \nabla \left(\frac{\mathcal{E}(x_\tau)}{k_B T} \right) d\tau + \sqrt{2\epsilon_\tau} dW_\tau. \end{aligned} \tag{7}$$

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. 7 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. Furthermore, while recent advances in divergence-free density estimation via the Itô density estimator (Skreta et al., 2024; 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 appendix §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* $\mu_{target}(x)$ *and the noise-adjusted distribution* $\mu'_{target}(x)$*. Consider the decomposition of a data sample into its constituent mean-free component,* $\tilde{x}$ *and center of mass* $c \in \mathbb{R}^3$*,* $x = \tilde{x} + c$*, where* $c \sim \mu(c)$ *and* $\mu(c)$ *is* $\mathrm{SO}(3)$*-invariant. Then* $\mu_{target}(\tilde{x}) = \mu'_{target}(\tilde{x})$ *if* $\mu'_{target}(x) = \mu(\tilde{x})\mu(\|c\|)$*.*

Proof. We start by noting $x = \tilde{x} + c$ and thus we can construct the target as a marginalization over c

$$\mu_{\text{target}}(\tilde{x}) = \int \mu_{\text{target}}(\tilde{x}, c) dc = \int \mu_{\text{target}}(\tilde{x}|c)\mu(c) dc \tag{8}$$

Now select $\mu(c) = \mu(\|c\|)\mu(\phi)\mu(\psi)$ which gives:

$$\int \mu_{\text{target}}(\tilde{x}|c)\mu(c) dc = \int \mu_{\text{target}}(\tilde{x}|c)\mu(\|c\|)\mu(\phi)\mu(\psi) dc. \tag{9}$$

But the target distribution is $\mathrm{SE}(3)$-invariant and thus this results in the following result,

$$\int \mu_{\text{target}}(\tilde{x}|c)\mu(\|c\|) dc = \int \mu_{\text{target}}(\tilde{x})\mu(\|c\|) dc = \mu'_{\text{target}}(x). \tag{10}$$

□

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:

$$\hat{p}(x) := \mathbb{P}(p(x) \geq \delta) = \frac{p(x)\mathbb{I}\{p(x) \geq \delta\}}{\int \mathbb{I}\{p(x) \geq \delta\} p(x) dx}, \tag{11}$$

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:

$$\text{TV}(p, \hat{p}) = \sup_{\phi \in \Phi} \left| \mathbb{E}_{x \sim p(x)}[\phi(x)] - \mathbb{E}_{\hat{x} \sim \hat{p}(x)}[\phi(\hat{x})] \right| = \frac{1}{2} \int |p(x) - \hat{p}(x)| dx. \tag{12}$$

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 $\hat{p}(x) = 0$,

$$\frac{1}{2} \int_{R_1} |p(x) - \hat{p}(x)| dx = \frac{1}{2} \int_{R_1} p(x) dx = \frac{\beta}{2}. \tag{13}$$

A similar computation on R_2 gives,

$$\frac{1}{2} \int_{R_2} |p_\theta(x) - \hat{p}_\theta(x)| dx = \frac{1}{2} \int_{R_2} \left| p(x) - \frac{p(x)}{\alpha} \right| dx = \frac{1}{2} \int_{R_2} p(x) \left| 1 - \frac{1}{\alpha} \right| dx = \frac{\alpha(\frac{1}{\alpha} - 1)}{2} = \frac{\beta}{2}, \tag{14}$$

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:

$$\text{TV}(p, \hat{p}) = \frac{1}{2} \int |p(x) - \hat{p}(x)| dx = \frac{1}{2} \int_{R_1} |p(x) - \hat{p}(x)| dx + \frac{1}{2} \int_{R_2} |p(x) - \hat{p}(x)| dx = \beta. \tag{15}$$

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:*

$$\gamma \geq \frac{1}{\lambda} \log \left(\frac{Kb}{12\rho \mathbb{E}[\exp(-\lambda X)]} \right) + \log \hat{\mathcal{Z}}. \tag{6}$$

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 $\mu(x)$:

$$\sup_{\|\phi\|_\infty \leq 1} \left| \mathbb{E}\left[\mu_\theta^K(\phi) - \mu_{\text{target}}(\phi)\right] \right| \leq \frac{12\rho}{K}, \quad \rho \approx \frac{K}{\text{ESS}} = \frac{K \sum_j^K w(x^j)^2}{\left(\sum_i^K w(x^i)\right)^2} \tag{16}$$

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. 5. 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:

$$\sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\mu_\theta^K(\phi) - \mu_{\text{target}}(\phi)\right]\right| \leq \sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\mu_\theta^K(\phi) - \hat{\mu}_{\text{target}}(\phi)\right]\right| + \sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\hat{\mu}_{\text{target}}(\phi) - \mu_{\text{target}}(\phi)\right]\right| \tag{17}$$

$$\leq \frac{12\hat{\rho}}{K} + \beta_1 \tag{18}$$

$$\leq \frac{12\rho}{K} + \beta_1. \tag{19}$$

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,

$$\sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\hat{\mu}_\theta^K(\phi) - \mu_{\text{target}}(\phi)\right]\right| \leq \frac{12\rho}{K} + \beta_1 \leq \frac{12\rho}{K} + \exp\left(\frac{\lambda\gamma}{\log \hat{\mathcal{Z}}}\right) \mathbb{E}[\exp(-\lambda X)]. \tag{20}$$

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

$$\gamma \geq \frac{1}{\lambda} \log\left(\frac{Kb}{12\rho\mathbb{E}[\exp(-\lambda X)]}\right) + \log \hat{\mathcal{Z}}. \tag{21}$$

□

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:*

$$\delta \geq \frac{1}{\lambda} \log\left(\frac{Kb}{12\rho\mathbb{E}[\exp(-\lambda X)]}\right). \tag{22}$$

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 $\mu_\theta(x)$:

$$\sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\mu_\theta^K(\phi) - \mu_{\text{target}}(\phi)\right]\right| \leq \frac{12\rho}{K}, \quad \rho \approx \frac{K}{\text{ESS}} = \frac{K\sum_j^K w(x^j)^2}{\left(\sum_i^K w(x^i)\right)^2} \tag{23}$$

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:

$$\text{TV}(\mu_\theta, \hat{\mu}_\theta) = \sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\mu_\theta^K(\phi) - \hat{\mu}_\theta^K(\phi)\right]\right| \tag{24}$$

$$= \sup_{\|\phi\|_\infty \leq 1} \left| \mathbb{E}_{x^i \sim p_\theta}\left[\sum_{i=1}^K \bar{w}(x^i)\phi(x^i)\right] - \mathbb{E}_{x^j \sim \hat{p}_\theta}\left[\sum_{j=1}^K \bar{w}(x^j)\phi(x^j)\right]\right| \tag{25}$$

$$= \frac{1}{2}\left(\mathbb{E}_{x^i \sim p_\theta}\left[\sum_{i=1}^K \bar{w}(x^i)\right] - \mathbb{E}_{x^j \sim \hat{p}_\theta}\left[\sum_{j=1}^K \bar{w}(x^j)\right]\right) \tag{26}$$

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,

$$\text{TV}(\mu, \hat{\mu}) = \frac{1}{2}\left(\mathbb{E}_{x^i \sim p_\theta}\left[\sum_{i=1}^K \bar{w}(x^i)\right] - \mathbb{E}_{x^j \sim \hat{p}_\theta}\left[\sum_{j=1}^K \bar{w}(x^j)\right]\right) \tag{27}$$

$$= \beta_2, \tag{28}$$

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

$$\sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\mu_\theta^K(\phi) - \mu_{\text{target}}(\phi)\right]\right| \leq \sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\hat{\mu}_\theta^K(\phi) - \mu_{\text{target}}(\phi)\right]\right| + \sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\mu_\theta^K(\phi) - \hat{\mu}_\theta^K(\phi)\right]\right| \tag{29}$$

$$\leq \frac{12\hat{\rho}}{K} + \beta_2 \tag{30}$$

$$\leq \frac{12\rho}{K} + \beta_2. \tag{31}$$

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,

$$\sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\mu_\theta^K(\phi) - \mu_{\text{target}}(\phi)\right]\right| \leq \frac{12\rho}{K} + \beta_2 \leq \frac{12\rho}{K} + \exp(\lambda\delta)\mathbb{E}[\exp(-\lambda X)]. \tag{32}$$

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

$$\delta \geq \frac{1}{\lambda}\log\left(\frac{Kb}{12\rho\mathbb{E}[\exp(-\lambda X)]}\right). \tag{33}$$

□

C Itô Filtering

C.1 Flow Matching SDE

As shown in Domingo-Enrich et al. (2024) we can write Flow Matching with Gaussian conditional paths and Diffusion models under a unified SDE framework given a reference flow:

$$x_t = \beta_t x_0 + \alpha_t x_1, \tag{34}$$

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:

$$x_t = (1-t)x_0 + tx_1. \tag{35}$$

The unified SDE for both flow matching and continuous-time diffusion models as introduced in Domingo-Enrich et al. (2024) is then:

$$dx_t = \kappa_t x + \left(\frac{\sigma_t^2}{2} + \eta_t\right)\mathfrak{s}(x_t, t) + \sigma_t dW_t, \quad \kappa_t = \frac{\dot{\alpha}_t}{\alpha_t}, \eta_t = \beta_t\left(\frac{\dot{\alpha}_t}{\alpha_t}\beta_t - \dot{\beta}_t\right) \tag{36}$$

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

$$dx_t = \left(2f_{t,\theta}(t, x_t) - \frac{x_t}{t}\right) dt + \sigma_t dW_t, \quad \sigma_t = \sqrt{(2(1-t)t)} \tag{37}$$

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

$$\nabla \log p_t(x_t) = \frac{t f_{t,\theta}(t, x_t) - x_t}{1-t}, \quad f_{t,\theta}(t, x_t) = \frac{x_t + (1-t)\nabla \log p_t(x_t)}{t} \tag{38}$$

Rewriting Eq. 37 in terms of the score function we get,

$$dx_t = \frac{x_t}{t} + \sigma_t^2 \nabla \log p_t(x_t) + \sigma_t dW_t. \tag{39}$$

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:*

$$dx_t = \frac{x_t}{t} + \sigma_t^2 \nabla \mathfrak{s}_\theta(t, x_t) + \sigma_t dW_t, \quad \sigma_t = \sqrt{(2(1-t)t)}. \tag{40}$$

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:

$$\sup_{\|\phi\|_\infty \leq 1} \left| \mathbb{E}\left[\mu^K_{r,\theta}(\phi) - \mu_{target}(\phi)\right] \right| \leq \frac{12\rho}{K} + \beta_3 + \beta_4, \tag{41}$$

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,

$$dx_t = f_t(t, x_t) + g_t dW_t, \tag{42}$$

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:

$$dh(t, x_t) = \left(\frac{\partial}{\partial t} h(t, x_t) + \frac{\partial}{\partial x} h(t, x_t)^T f_t(t, x_t) + \frac{1}{2}\sigma_t^2 \Delta_x h(t, x_t)\right) dt + \sigma_t \frac{\partial}{\partial x} h(t, x_t) dW_t. \tag{43}$$

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., 2024; Karczewski et al., 2024) but for flow models

$$d \log p_t(x_t) = \left(\frac{\partial}{\partial t} \log p_t(x_t) + \frac{\partial}{\partial x} \log p_t(x_t)^T f(t, x_t) + \frac{1}{2}\sigma_t^2 \Delta_x \log p_t(x_t)\right) dt + \sigma_t \frac{\partial}{\partial x} \log p_t(x_t) dW_t, \tag{44}$$

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

$$\frac{\partial}{\partial t} \log p_t(x) = -\nabla \cdot (f(t, x)) + \frac{1}{2}\sigma_t^2 \Delta_x \log p_t(x) - \nabla_x \log p_t(x)^T \left(f(t, x) - \frac{1}{2}\sigma_t^2 \nabla_x \log p_t(x)\right) \tag{45}$$

in the general case we end with:

$$d \log p_t(x_t) = \left(-\nabla \cdot \left(f(t, x_t) - \sigma_t^2 \nabla_x \log p_t(x_t)\right) + \frac{1}{2}\sigma_t^2 \|\nabla_x \log p_t(x_t)\|^2\right) dt + \sigma_t \nabla_x \log p_t(x_t)^T dW_t. \tag{46}$$

We now apply this to the flow-matching SDE Eq. 39 written in terms of the score function. In particular, we have,

$$\begin{aligned} d \log p_t(x_t) &= \left(-\nabla \cdot \left(\sigma_t^2 \nabla_x \log p_t(x_t) + \frac{x_t}{t} - \sigma_t^2 \nabla_x \log p_t(x_t)\right) + \frac{1}{2}\sigma_t^2 \|\nabla_x \log p_t(x_t)\|^2\right) dt \\ &\quad + \sigma_t \nabla_x \log p_t(x_t)^T dW_t \\ d \log p_t(x_t) &= \left(-d/t + \frac{1}{2}\sigma_t^2 \|\nabla_x \log p_t(x_t)\|^2\right) dt + \sigma_t \nabla_x \log p_t(x_t)^T dW_t. \end{aligned} \tag{47}$$

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. 38. When working with imperfect score estimates we have the following SDE:

$$dx_t = \frac{x_t}{t} + \sigma_t^2 \nabla \mathfrak{s}_\theta(t, x_t) + \sigma_t dW_t. \tag{48}$$

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]:

$$\log r_0(x_0) = \log p_0(x_0) + Y \tag{49}$$

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:

$$\mathbb{E}[Y] = \underbrace{\frac{1}{2}\mathbb{E}_{t\sim\mathcal{U}(0,1), x_t\sim p_t(x_t)}\left[\sigma_t^2 ||\mathfrak{s}_\theta(t, x_t) - \nabla \log p_t(x_t)||^2\right]}_{\geq 0} \tag{50}$$

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 $\mu_{r,\theta}^K$

$$\begin{aligned}
\mathrm{TV}(\mu_{r,\theta}, \mu_\theta) &= \sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\mu_{r,\theta}^K(\phi) - \mu_\theta^K(\phi)\right]\right| && (51)\\
&= \sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}_{x^i\sim p_\theta}\left[\sum_{i=1}^K \bar{w}_r(x^i)\phi(x^i)\right] - \mathbb{E}_{x^j\sim p_\theta}\left[\sum_{j=1}^K \bar{w}(x^j)\phi(x^j)\right]\right| && (52)\\
&= \frac{1}{2}\left(\mathbb{E}_{x^i\sim p_\theta}\left[\sum_{i=1}^K \bar{w}_r(x^i)\right] - \mathbb{E}_{x^j\sim p_\theta}\left[\sum_{j=1}^K \bar{w}(x^j)\right]\right) && (53)\\
&= \frac{1}{2}\left(\mathbb{E}_{x^i\sim p_\theta}\left[\sum_{i=1}^K \exp\left(\frac{1}{2}\mathbb{E}_{t\sim\mathcal{U}(0,1), x_t\sim p_t(x_t)}\left[\sigma_t^2 ||\mathfrak{s}_\theta(t, x_t) - \nabla \log p_t(x_t)||^2\right]\right)\right]\right) && (54)\\
&:= \beta_3 && (55)
\end{aligned}$$

The total bias is then

$$\sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\mu_{r,\theta}^K(\phi) - \mu_{\text{target}}(\phi)\right]\right| \leq \frac{12\rho}{K} + \beta_3. \tag{56}$$

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 $\mathrm{TV}(\mu_{r,\theta}, \hat{\mu}_{r,\theta}) = \beta_4$ to the overall bias:

$$\sup_{\|\phi\|_\infty \leq 1} \left|\mathbb{E}\left[\mu_{r,\theta}^K(\phi) - \mu_{\text{target}}(\phi)\right]\right| \leq \frac{12\rho}{K} + \beta_3 + \beta_4. \tag{57}$$

D Experimental Details

D.1 Further details on experimental setup

Metrics and sampling setup. For all metrics we first generate samples, then resample to 10k samples, and finally compute metrics to control for the error in distribution metrics from empirical sample size. For all models we draw 10k samples unless otherwise noted. For AL6 ECNF++ due to compute cost we instead draw 1k samples.

Sampling time calculations. For the sampling time, we compute all times on a single A100. For the continuous normalizing flow models we use a the maximum power of two batch size that fits on an A100, and average over at least 10 batches.

Training time. For training times we compute all times on a single A100 80Gb GPU except for SE(3)-EACF which is trained on a single H100. We compute the total time in hours till convergence for all methods and report it in the table below.

Table 4: Training time (in hours) for all methods.

Model	ALDP	AL3	AL4	AL6	Chignolin
SE(3)-EACF	160	-	-	-	-
ECNF	4.17	5.83	8.89		
ECNF++	9.72	12.5	17.17	76.94	
SBG	16.83	24.67	41.67	57.5	427.33

D.2 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. We choose the most stable Spherical-projection EACF with a 20-layer configuration. Each layer has two ShiftCoM layer and two core-transformation blocks. The EGNN used in the core transformation block consists of three 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 has been trained for 50 epochs with a batch size of 20 using Adam optimizer and peak learning rate of 1e-4. We use the default 20 samples to estimate likelihoods using importance sampling.

EACF as a Boltzmann generator. EACF is augmented, 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

$$q(x) = \mathbb{E}_{a\sim\pi(\cdot|x)}\left[\frac{q(x,a)}{\pi(a|x)}\right] \tag{58}$$

however, this is only a consistent estimator of the likelihood and for finite sample sizes has variance. This makes this 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.

D.3 ECNF++ Implementation details

D.3.1 Network and training

Equivariant continuous normalizing flow (ECNF) (Klein & Noé, 2024). We use the supplied pretrained model from Klein & Noé (2024) for our ECNF baseline on ALDP. 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 five improvements to the ECNF, which together substantially improve performance.

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., 2023, Proposition 3.3).

ECNF uses $\sigma = 0.01$ where we use $\sigma = 0$. We find that $\sigma > 0$ in this case causes some 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. **Data normalization strategy.** in previous work, data was normalized to Ånstrom (Å) scale. We find that this is too small for stable neural network training. We employ the standard scheme of standardization based on the training data. Specifically, we subtract the center of mass of each atom, and divide by the standard deviation of the data. Typically this would be done with a per-dimension standard deviation. However, to maintain $\mathrm{SE}(3)$ equivariance we use a single standard deviation for the whole dataset. On ALDP, the standard deviation of the normalized training data is approximately 0.16. This means we scale up the training data roughly $6.25\times$ on ALDP. We find this greatly improves the training dynamics and final structure precisions.

3. **4x wider layers.** Empirically, we find the EQCNF to be underparameterized. We did a grid search over parameter widths and depths to find a balance between performance and speed on ALDP. We found empirically that a width of 256 with 5 blocks provided the best tradeoff between speed and performance on ALDP. We used the same parameters for larger moleculer systems.

4. **Improved optimizer and LR scheduler** We find using an AdamW with fairly large weight decay 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 training. We also use a smoothly varying cosine schedule with warmup enables a larger maximum learning rate and faster training than the two step schedule used previously.

5. **Exponential moving average** we use an exponential moving average (EMA) on the weights with decay 0.999. This is a standard trick in flow models, which improves performance.

These five elements together greatly improve the ECNF training 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 likelihood which is necessary for self-normalized importance sampling.

Other parameters For both models we use default optimizer parameters for $\beta_1, \beta_2, \epsilon$, 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 continuous normalizing flow model requires calculating the trace of the divergence. This is quite an expensive operation in terms of both time and memory. 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 autograd which can be quite memory and time intensive. For example, on AL6, the maximum batch size that can fit on an 80GB A100 is 8. This batch takes around two minutes for 84 integration steps. We also use an improved vectorized Jacobian trace implementation for all continuous normalizing flows which reduces memory by roughly half and time by roughly 3x over the pre-TORCH.VMAP implementation which loops over dimensions. We note that these numbers are approximate and depend heavily on both the batch size and the input dimension.

On using a CNF 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 suck an approach. The largest of which, is efficiency. CNFs are extremely computationally inefficient to sample a likelihood from as previously discussed. 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 we are aleady at the edge of tractibility for the current importance sampling estimates. We leave it to future work to consider faster CNFs and note that our SBG algorithm could be applied there immediately.

Dataset	Layers per Block	Number Blocks	Channels	Number Parameters (M)
ALDP	4	4	256	12.7
3-peptide	6	6	256	28.5
4-peptide	6	6	384	64.0
5-peptide	6	6	384	64.0
6-peptide	6	6	384	64.0
10-peptide	8	8	512	202.0

Table 5: TarFlow configurations across different datasets

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

D.4 SBG Implementation details

As advised by Zhai et al. (2024) we scale the layers per block alongside the number of blocks.

E Datasets

For all datasets besides AD2 we use a training set of 100k contiguous samples (1ns simulation time) from a single MCMC chain, a validation set of the next 20k contiguous samples, and a test set of 100k uniformly sampled samples from the rest of the dataset. Since these are highly multimodal energy functions, this leaves us with biased training data relative to the Boltzmann distribution. We split this way to test the model in a challenging and realistic setting — where some biased samples from MD exist and we would like to generate more uncorrelated and unbiased samples. We describe the datasets below and present the simulation parameters in Table 7.

Alanine Dipeptide (AD2). 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.

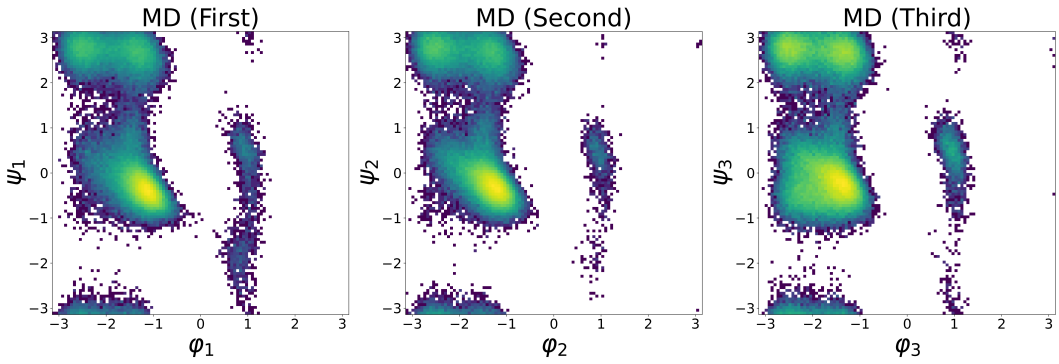
Trialanine (AD3) and Hexaalanine (AD6). For the peptides composed of multiple alanine amino acids, we generate MD trajectories using the *OpenMM* library (Eastman et al., 2017). All simulations are conducted in implicit solvent, with the simulation parameters detailed in Table 7. These systems do not include any additional capping groups, such as those present in alanine dipeptide (AD2) and alanine tetrapeptide (AD4), as they are generated in the same manner as described in Klein et al.

Table 6: Overview of continuous normalizing flow training setup.

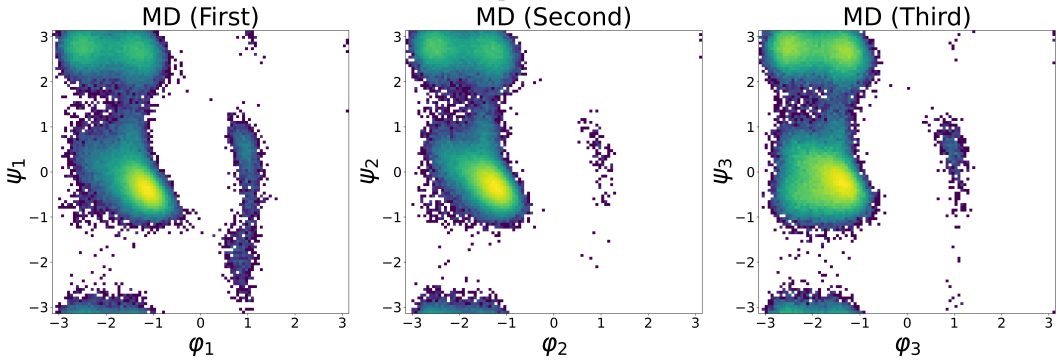
Training Parameter	ECNF	ECNF++
Optimizer	ADAM	ADAM-W (Loshchilov, 2017)
Learning Rate	5×10^{-4}	5×10^{-4}
Weight Decay	0.0	0.01
width	64	256
n blocks	5	5
EMA Decay	1.0	0.999
Parameters	152 K	2.317 M

Table 7: Overview of simulation parameters

Peptide	Force field	Temperature	Time step
Alanine dipeptide (ALDP)	Amber ff99SBildn	300K	1fs
Trialanine (AL3)	Amber 14	310K	1fs
Alanine tetrapeptide (AL4)	Amber ff99SBildn	300K	1fs
Hexaalanine (AL6)	Amber 14	310K	1fs

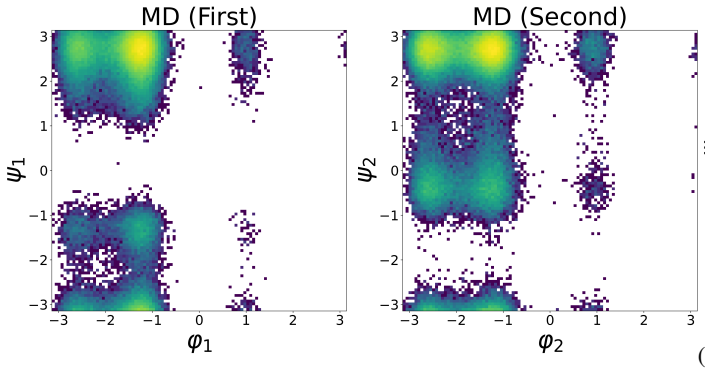


(a) Ramachandran plots for AL4 test dataset

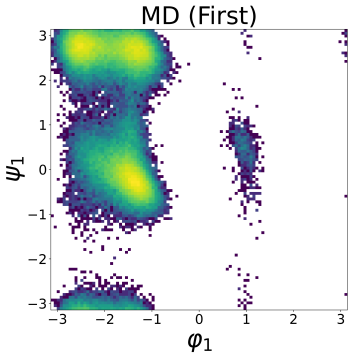


(b) Ramachandran plots for AL4 train dataset

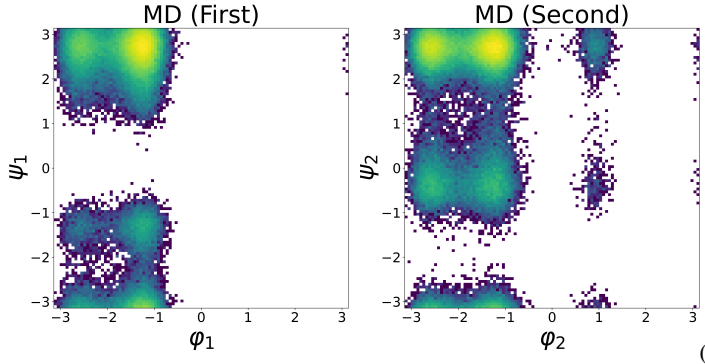
Figure 6: Ramachandran plots for the AL4 dataset with test (a) and training (b) histograms over φ and ψ angles. We can see that the training set is slightly biased with underrepresentation of the small right mode at $\psi_2 \approx 0$ and $\varphi_2 \approx 1$ in the training set as compared to the test set samples.



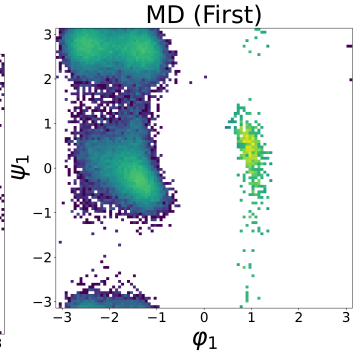
(a) Ramachandran plots for AL3 test dataset



(b) Ramachandran plots for AL2 test dataset



(c) Ramachandran plots for AL3 train dataset



(d) Ramachandran plots for AL2 train dataset

Figure 7: Ramachandran plots for the AL3 test (a) and train (c) histograms over φ and ψ angles and the AL2 dataset with the test (b) and train (d) datasets. For AL3 we can see that the training set is completely missing the right mode in ψ_1, φ_1. For AL2 we can see that the right mode has been oversampled relative to that of the test set.

(2023a). There are two peptide bonds in alanine tripeptide (AD3) and five in alanine hexapeptide (AD6), resulting in two and five distinct 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 distinct Ramachandran plots.

Chignolin. In addition to the small peptide systems, we also investigate the small protein Chignolin, which consists of ten amino acids. Generating a fully converged all-atom simulation for this system is computationally expensive. Therefore, we use the trajectory provided by (Lindorff-Larsen et al., 2011), which was generated using a specialized supercomputer. In contrast to our other datasets, this simulation was performed in explicit solvent and with a different force field. Since our models do not incorporate additional water molecules, we treat the dataset as if it were in implicit solvent and use the same force field as for the other datasets, namely Amber 14. As a result, the trajectory originates from a slightly different distribution than given by the force-field, likely introducing some bias. Therefore, the task is to generate samples from the equilibrium distribution in implicit solvent while only having access to training data obtained from explicit solvent simulations. As before, we use only the first 100k samples for training. This again highlights the strength of Boltzmann generator based methods, which do not require equilibrium training data. However, it also presents an evaluation challenge, as we lack access to equilibrium samples for the implicit solvent simulation to serve as a reference.

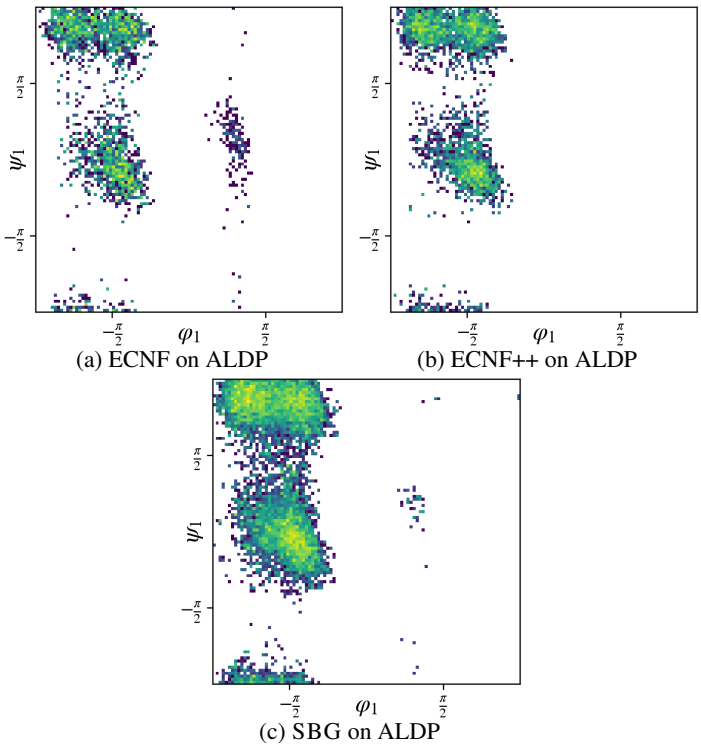
F Additional Results

F.1 Ramachandran Plots

In this appendix we include the Ramachandran plots for each model on each peptide. Please note that the ground truth training and test Ramachandran plots are located in Appendix E.

To create the Ramachandran plots we take our samples and try to enforce equal chirality. We find some samples either have the wrong topology or are the wrong chirality. For chirality we attempt to fix all chiralities to the same direction. If we fail to do so, then we filter this point out. This leaves us with Ramachandran plots with fewer samples, but all with the correct chirality. The incorrect chirality can show up as a symmetric mode on Ramachandran plots. We note that this filtering step can lead to blank Ramachandran plots if all points are sampled out. This is the case for the ECNF for AL3 and AL4.

Alanine dipeptide (ALDP). In Figure 8 we can see the Ramachandran plot for resampled points. We find that that ECNFF++ models the distribution well, but drops the right mode. Which is quite interesting as the right mode is oversampled in the training data (see Figure 7).



(a) ECNF on ALDP (b) ECNF++ on ALDP

(c) SBG on ALDP

Figure 8: Alanine dipeptide Ramachandran plots for various models.

Trialanine (AL3).

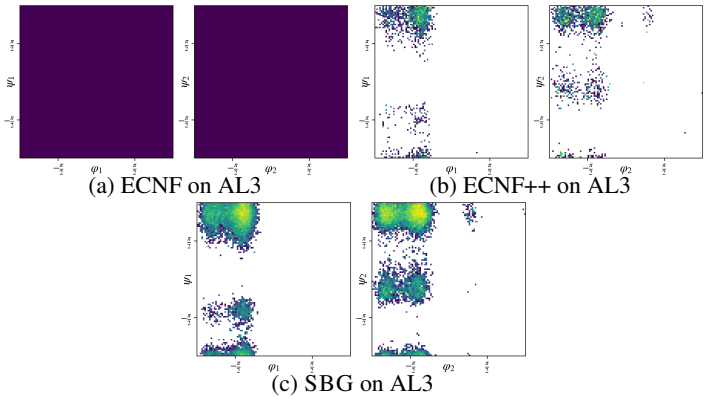
Alanine tetrapeptide (AL4).

Hexaalanine (AL6).

F.2 Ablation studies

Center of Mass Augmentation. We now ablate the utility of performing the CoM augmentation with the corresponding model energy adjustment. Specifically, we ablate the CoM augmentation as a function of number of samples used during inference and also as a function of a number of inference timesteps. Each of these ablations is performed on the trialanine tripeptide (AL3). We find that CoM augmentation reduces the Torus Wasserstein distance ($\mathbb{T}$-$\mathcal{W}_2$) fairly consistently across timesteps and number of samples.

Ablation on EACF Importance Weight Clipping. We report the additional results on EACF trained on ALDP dataset. We chose to use 0.2% clip threshold on the importance weights for fair comparison.

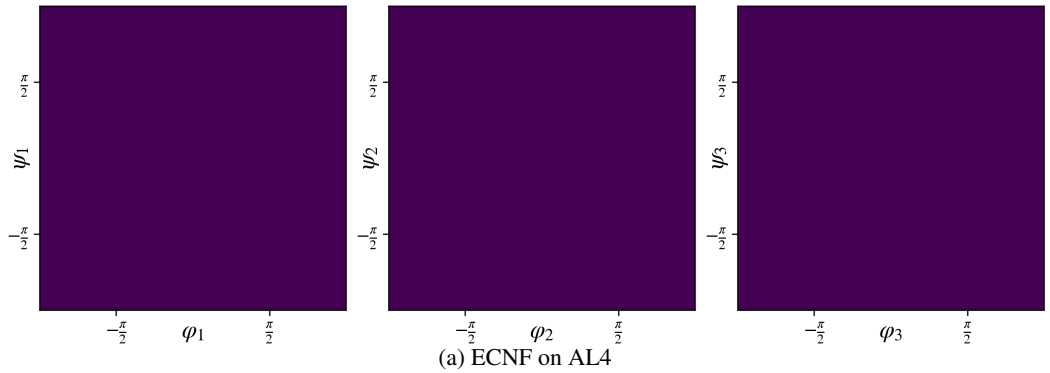


(a) ECNF on AL3 (b) ECNF++ on AL3

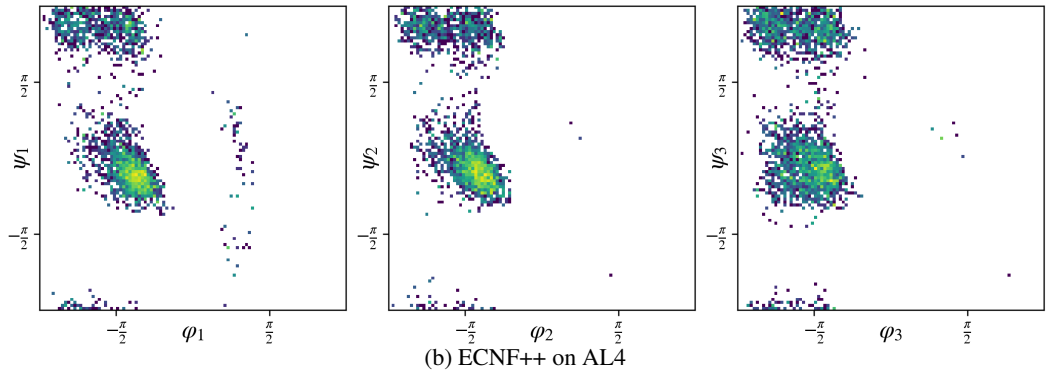
(c) SBG on AL3

Figure 9: Trialanine Ramachandran plots for various models. For ECNF none of the samples pass our filtering thresholds, and thus the Ramachandran plot for ECNF depicts zero samples.

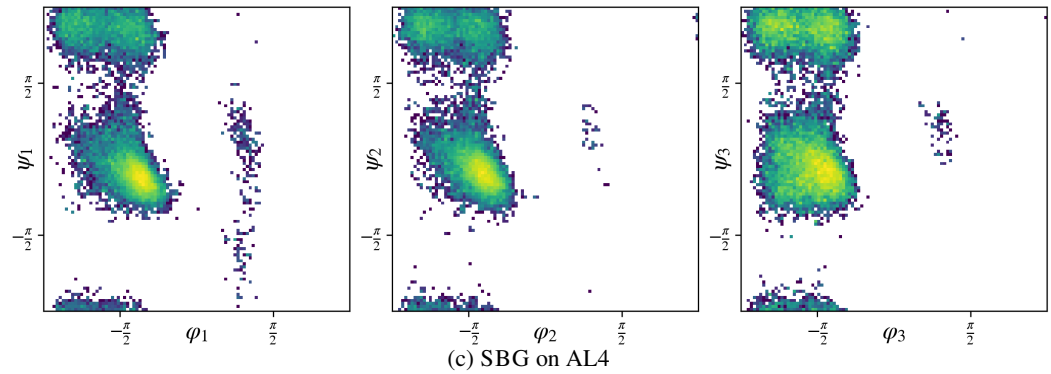
Nevertheless, in the resampling process, we observe a significant degradation in sample diversity, as evidenced by the energy histograms and Rama plots. From the qualitative results in Figure 13, we can see that EACF generates highly unreliable importance weights, particularly visible in the energy histograms where there are extreme spikes and poor alignment with the true data distribution. This leads to poor resampling quality, as demonstrated in the corresponding Rama 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) ECNF on AL4

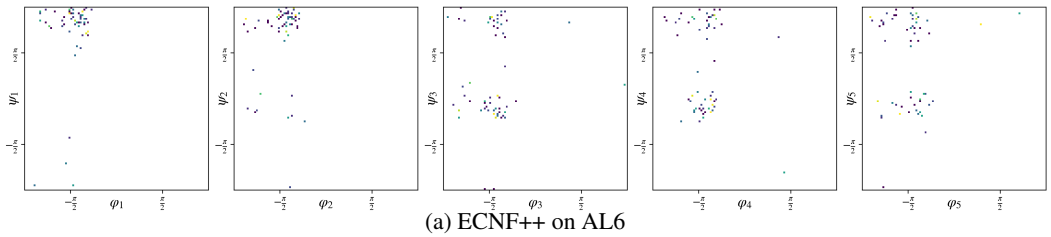


(b) ECNF++ on AL4

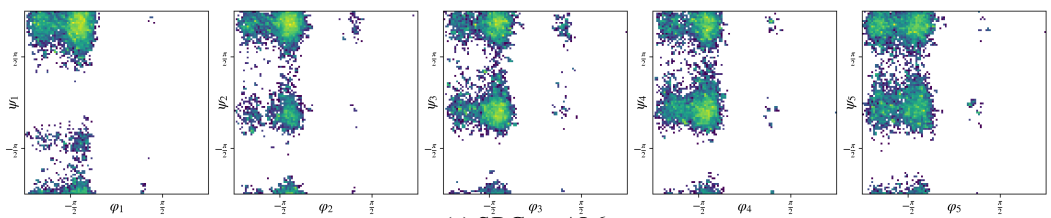


(c) SBG on AL4

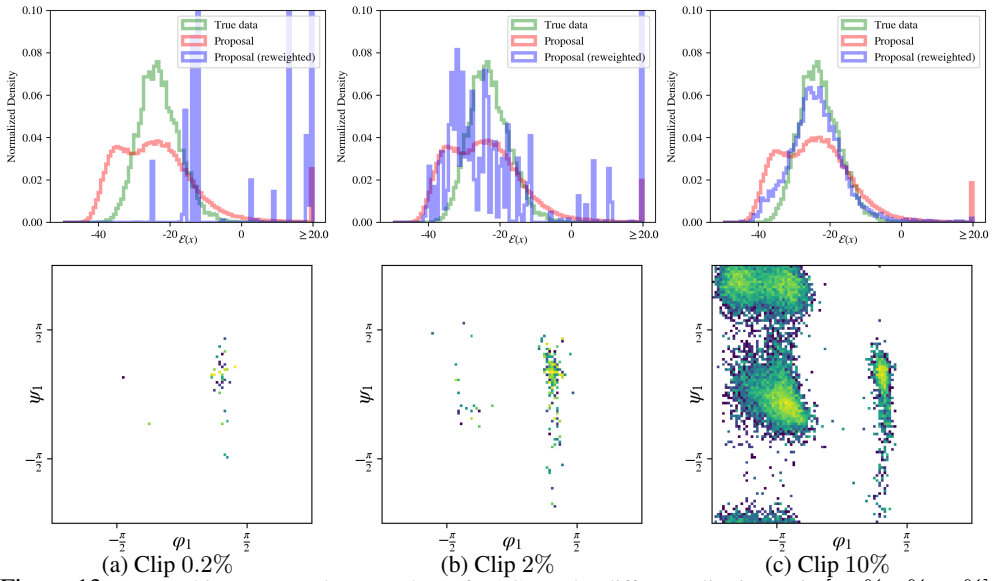
Figure 10: Ramachandran plots for various models on AL4 dataset. We note that for the ECNF model none of the samples passed our filter. We find that SBG manages to still capture the right mode where ECNF++ often drops this mode.



(a) ECNF++ on AL6



(a) SBG on AL6



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

Figure 13: Energy histogram and Rama Plots of EACF under different clipping ratio $[0.2\%, 2\%, 10\%]$.

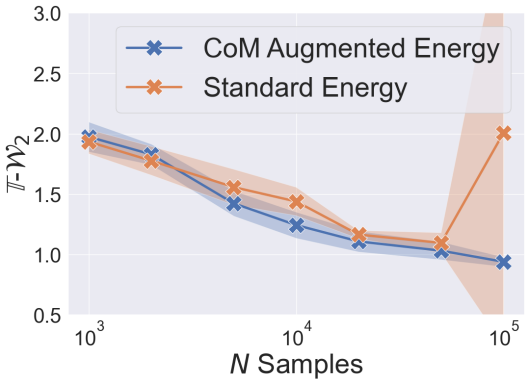


Figure 14: Standard and center of mass augmented energy at a variety of sampling set sizes.

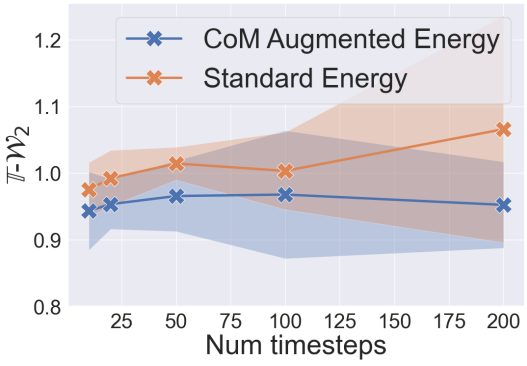


Figure 15: Standard and center of mass augmented energy at a variety of Langevin time discretizations.

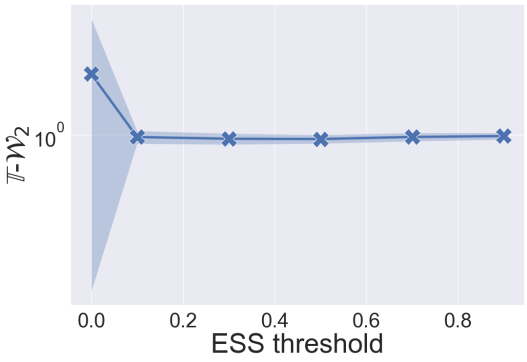


Figure 16: Standard and center of mass augmented energy at a variety of ESS resampling thresholds.