Conditional Normalizing Flows for Active Learning of Coarse-Grained Molecular Representations

Henrik Schopmans [1 2] **Pascal Friederich** [1 2]

Abstract

Efficient sampling of the Boltzmann distribution of molecular systems is a long-standing challenge. Recently, instead of generating long molecular dynamics simulations, generative machine learning methods such as normalizing flows have been used to learn the Boltzmann distribution directly, without samples. However, this approach is susceptible to mode collapse and thus often does not explore the full configurational space. In this work, we address this challenge by separating the problem into two levels, the fine-grained and coarse-grained degrees of freedom. A normalizing flow conditioned on the coarse-grained space yields a probabilistic connection between the two levels. To explore the configurational space, we employ coarse-grained simulations with active learning which allows us to update the flow and make all-atom potential energy evaluations only when necessary. Using alanine dipeptide as an example, we show that our methods obtain a speedup to molecular dynamics simulations of approximately 15.9 to 216.2 compared to the speedup of 4.5 of the current state-of-the-art machine learning approach.

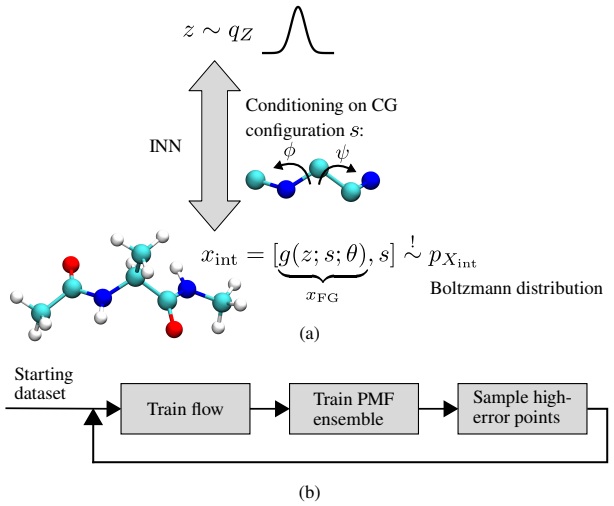


Figure 1. (a) The conditional normalizing flow transforms the latent variable z of the latent distribution to the target configuration x_{int} conditioned on the CG configuration s. (b) Illustration of the iterative three-step active learning cycle.

1. Introduction

Coarse-graining (CG) of molecular dynamics (MD) simulations is a powerful technique to bridge time and length scales that are often infeasible to study on an atomistic level. Numerous processes, especially large-scale conformational changes in biomolecules such as the folding of proteins, occur on the microseconds to seconds timescale (Spiriti et al., 2012). Since a typical time step used in all-atom molecular dynamics simulations is 1-2 fs, covering such time scales is currently computationally very expensive and infeasible for many systems of interest.

Bottom-up coarse-graining of all-atom simulations introduces effective sites that describe the collective motion of one or multiple atoms of the all-atom description. The potential of mean force (PMF) allows simulation of the CG system and averages out many of the fast degrees of freedom, making the PMF smoother than the all-atom potential energy (Jin et al., 2022). This allows larger time steps to be used than for typical all-atom simulations. The reduced number of degrees of freedom, combined with a larger time step, allows for studying larger systems with extended simulation times.

Next to classical CG force fields such as Martini (Marrink et al., 2023) or UNICORN (Liwo et al., 2020), machine learning (ML) models have been successfully used to

[1]Institute of Nanotechnology, Karlsruhe Institute of Technology, Kaiserstr. 12, 76131 Karlsruhe, Germany [2]Institute of Theoretical Informatics, Karlsruhe Institute of Technology, Kaiserstr. 12, 76131 Karlsruhe, Germany. Correspondence to: Pascal Friederich <pascal.friederich@kit.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

parametrize the PMF as a function of the CG coordinates (Husic et al., 2020; Thaler et al., 2022; Zhang et al., 2018; Majewski et al., 2023; Charron et al., 2023; Wang et al., 2019). Typically, a long and thus very costly all-atom MD trajectory is required to learn the PMF on the CG level. Although the CG model enables the simulation of significantly longer simulation times, it is unlikely to reveal novel statistics not already present in the all-atom training data, such as interesting transitions and new metastable states.

In this work, we demonstrate how active learning (AL) can be used to iteratively refine the PMF while efficiently exploring the configurational space on the CG level. This removes the need for long all-atom simulations to obtain the PMF. Implementing an active learning workflow for coarse-grained simulations yields a critical challenge: PMF values in high-error regions of the CG space cannot be directly evaluated and need to be obtained from conditionally sampling the all-atom Boltzmann distribution in the high-error CG configurations. Duschatko et al. (2024) showed that this can be accomplished using constrained MD simulations. However, this approach is limited to small systems since an MD simulation needs to be performed for each high-error configuration. In this work, we use a simulation-free alternative based on a conditional normalizing flow (Winkler et al., 2023; Xiao et al., 2019; Ardizzone et al., 2019) to sample the all-atom level given a high-error CG configuration.

Noé et al. (2019) showed that a normalizing flow can not only be trained with samples from MD, but also by using the Boltzmann distribution $\sim \exp(-\frac{E(x)}{k_\mathrm{B}T})$ directly in the loss function without samples (training by energy). This allows for simulation-free learning of the potential energy surface. Recently, this idea has been extended in other studies (Mahmoud et al., 2022; Midgley et al., 2023b). However, this form of training a normalizing flow typically suffers from mode collapse. While there are approaches that improve the tendency of mode collapse when training without samples (Midgley et al., 2023b; Felardos et al., 2023; Vaitl et al., 2022), a universal solution is yet to be found, especially for large systems or systems with very large energy barriers. The preprint by Zhang et al. (2023) used a conditional normalizing flow for a coarse-graining task without active learning, however no benchmarks or implementation details are available.

We solve the aforementioned challenges by developing a normalizing flow trained by energy to generate the fine-grained degrees of freedom conditioned on the CG space. We further show how, based on the trained conditional normalizing flow, an ensemble of PMF models can be trained in the CG space. We iteratively sample new high-error points from the PMF ensemble and subsequently refine the conditional normalizing flow. Since the main modes of the Boltzmann distribution are encapsulated in the CG space and the normalizing flow only learns the conditional "soft" fine-grained degrees of freedom, we avoid mode collapse.

Using alanine dipeptide with a two-dimensional CG space consisting of the dihedral angles ϕ and ψ, we demonstrate that our methodology produces PMF maps of higher accuracy while using approximately two orders of magnitude less potential energy evaluations compared to running all-atom MD simulations and one order of magnitude less compared to the state-of-the-art ML approach of learning the Boltzmann distribution without samples (Midgley et al., 2023b). Our methods obtain a speedup to molecular dynamics simulations of approximately 15.9 to 216.2 compared to the speedup of 4.5 of the method by Midgley et al. (2023b). To the best of our knowledge, this is the first time that active learning on the CG side is demonstrated without the use of costly constrained MD simulations to evaluate the PMF. We believe that extending this methodology to more complicated CG spaces and molecular systems will yield a powerful technique to efficiently generate coarse-grained potentials and simulations. Furthermore, while we focus on molecular systems in this work, our approach can be applied to any general sampling problem of an unnormalized probability density, where a meaningful CG space that contains the main modes of the probability distribution can be defined.

2. Background

2.1. Normalizing Flows

A normalizing flow transforms a latent probability distribution $q_Z(z)$, for example, a standard Gaussian $\mathcal{N}(z; 0, \mathbf{I})$, using a transformation $x = g(z; \theta)$ with parameters θ. The transformed probability density $q_X(x; \theta)$ can be expressed using the change of variables formula (Dinh et al., 2015):

$$q_X(x;\theta) = q_Z\left(f(x;\theta)\right) \left|\det J_{x\mapsto z}\right| \tag{1}$$

$$\text{with the Jacobian } J_{x\mapsto z} = \frac{\partial f(x;\theta)}{\partial x^T} \tag{2}$$

This requires the inverse of $g(z;\theta)$, $f(x;\theta) \equiv z = g^{-1}(x;\theta)$. A popular choice to parametrize such an invertible function (invertible neural network, INN) is a stack of coupling layers, where the dimensions of the input $x_{1:D}$ are split into two parts, $x_{1:d}$ and $x_{d+1:D}$. The first part undergoes an invertible transformation conditioned on the second part, which stays unchanged (see SI Section A.3 for details).

The parameters of the flow can be trained in such a way that the generated distribution $q_X(x;\theta)$ approximates the Boltzmann distribution $p_X(x)$. Normalizing flows have a vital advantage over most other generative machine learning

methods in that they can express the exact probability density efficiently. This allows training both with samples from the target distribution using the forward Kullback–Leibler divergence (KLD) and without samples using the reverse KLD.[1]

To derive both modes of training, we first define the following probability distributions:

- $q_Z(z)$ is the latent distribution we sample from.
- $q_X(x;\theta)$ is the generated distribution of the flow (see Equation 1).
- $p_X(x) \sim \exp\left(-E(x)/(k_\mathrm{B}T)\right)$ is the target Boltzmann distribution, where $E(x)$ is the potential energy of the molecular system, with temperature T and Boltzmann constant k_B.
- $p_Z(z;\theta) = p_X(g(z;\theta))\,|\det J_{z\mapsto x}|$ is the Boltzmann probability transformed by the INN to the latent space.

Training by Example. Using the forward KLD as the loss function, we can train a normalizing flow using samples from the target distribution (Noé et al., 2019):

$$\mathrm{KL}_\theta\left[p_X \| q_X\right] = C - \int p_X(x) \log q_X(x;\theta)\,\mathrm{d}x \tag{3}$$

$$= C - \mathbb{E}_{x\sim p_X}\left[\log q_Z\left(f(x;\theta)\right) + \log|\det J_{x\mapsto z}|\right] \tag{4}$$

Training by Energy. We can also use the reverse KLD to train the flow directly using the target energy function of the system without samples from the target density (Noé et al., 2019):

$$\mathrm{KL}_\theta\left[q_Z \| p_Z\right] = C - \int q_Z(z) \log p_Z(z;\theta)\,\mathrm{d}z \tag{5}$$

$$= C - \mathbb{E}_{z\sim q_Z}\left[\underbrace{\log p_X\left(g(z;\theta)\right)}_{-\frac{1}{k_\mathrm{B}T}E(g(z;\theta))+C_1} + \log|\det J_{z\mapsto x}|\right] \tag{6}$$

In this way, one samples from the flow and adjusts the weights of the transformation to fit the Boltzmann distribution given by the energy function.

2.2. Coarse-Graining

Coarse-graining combines the atom coordinates $x \in \mathbb{R}^{3N}$ of a molecular system into $m < 3N$ coordinates using a CG mapping $s = \xi(x)$, $s \in \mathbb{R}^m$. Typically, the coordinates s are obtained from a linear transformation $\mathbb{R}^{3N} \to \mathbb{R}^{m=3M}$. A popular choice is a "slicing" mapping (Yang et al., 2023) that simply uses a selection M of the N all-atom coordinates as CG coordinates, such as the coordinates of the backbone atoms in a protein. Here, we will consider general, potentially nonlinear CG mappings.[2]

Our goal is to construct a potential of mean force (PMF) $U_\mathrm{PMF}(s)$ that can be used to sample from the CG space, e.g., through Langevin dynamics or Metropolis Monte Carlo (MC). To ensure thermodynamic consistency in the CG configurational space we need (Wang et al., 2019; Noid et al., 2008)

$$U_\mathrm{PMF}(s) \equiv -k_\mathrm{B}T \ln \underbrace{\frac{\int p_X(x)\delta(s-\xi(x))\,\mathrm{d}x}{\int p_X(x)\,\mathrm{d}x}}_{p^\mathrm{CG}(s)}. \tag{7}$$

Due to the form of Equation 7, the potential of mean force is also called the (conditional) free energy.

Unfortunately, we cannot directly train a model to predict the potential of mean force, since training labels are not directly available. We can, however, use the all-atom forces projected to the CG space, $h(x)$. The expectation value of these projected forces for a given CG configuration yields the negative gradient of the PMF (CG mean force), which can be derived directly from Equation 7 (Kalligiannaki et al., 2015; Noid et al., 2008; Izvekov & Voth, 2005):

$$-\nabla_s U_\mathrm{PMF}(s) = \mathbb{E}_{x\sim p_X}\left[h(x) \mid \xi(x) = s\right] \tag{8}$$

If one uses a "slicing" CG mapping, the projected all-atom forces $h(x)$ are simply the all-atom forces of the chosen CG atoms. One can, however, also find an expression for $h(x)$ in other scenarios, even in the general nonlinear case (Kalligiannaki et al., 2015).

Evaluating the conditional expectation value in Equation 8 is not straightforward. Instead, one can show that minimizing the surrogate loss (Noid et al., 2008; Izvekov & Voth, 2005)

$$\chi(W) = \left\langle \|h(x) + \nabla_s U_\mathrm{PMF}(\xi(x);W)\|^2 \right\rangle_{x\sim p_X} \tag{9}$$

also yields the correct U_PMF model with parameters W. This is called multiscale force-matching.

Köhler et al. (2023) recently showed an alternative to the force-matching method called "flow-matching" that uses a normalizing flow trained directly on the distribution of

[1] Recently, diffusion models have also been used for training with the reverse KLD (Jing et al., 2022).

[2] Low-dimensional nonlinear CG mappings are often called collective variables in the literature.

all-atom MD samples transformed to the CG space. The PMF of the CG representation is then given by $U_{\text{PMF}}(s) = -k_{\text{B}}T \ln p^{\text{CG}}(s)$, where $p^{\text{CG}}(s)$ can be obtained directly from the normalizing flow using Equation 1.

Both approaches, multiscale force-matching and flow-matching, require a comprehensive all-atom dataset from the Boltzmann distribution for the training of the PMF.

3. Methods

Now, we show how an iterative active learning workflow for coarse-graining of molecular systems can be constructed without the need to perform long and costly all-atom simulations.

We define an internal coordinate representation $x_{\text{int}}(x) = [x_{\text{FG}}(x), \xi(x)]$ (with $[\,]$ denoting concatenation), which splits the all-atom cartesian coordinates x into fine-grained coordinates $x_{\text{FG}}(x)$ and coarse-grained coordinates $s = \xi(x)$. $x_{\text{FG}}(x)$ are the remaining fine-grained coordinates needed to reconstruct x given s, $x = x([x_{\text{FG}}, s])$.[3] In the case of "slicing" mappings, this can be a simple splitting of the atom coordinates, but it can also be an actual internal coordinate representation, which we use for our experiments.

A normalizing flow can now be used to parametrize the probability distribution of the fine-grained coordinates conditioned on the CG coordinates (see also Figure 1):

$$q_{X_{\text{FG}}}\left(x_{\text{FG}} \mid s; \theta\right) = q_Z\left(f(x_{\text{FG}}; s; \theta)\right) \left|\det J_{x_{\text{FG}} \mapsto z}\right| \quad (10)$$

Training by energy of this conditional flow can be performed using the loss in Equation 6 in the space $X = X_{\text{FG}}$ with conditioning on s. Here, the target distribution is given by

$$p_{X_{\text{FG}}}(x_{\text{FG}} \mid s) \sim \quad (11)$$
$$\frac{1}{p^{\text{CG}}(s)} \underbrace{\exp\left(\frac{-E(x([g(z; s; \theta), s]))}{k_{\text{B}}T}\right) \left|\det J_{x_{\text{int}} \to x}\right|}_{\sim p_{X_{\text{int}}}(x_{\text{int}})},$$

where $p^{\text{CG}}(s)$ is independent of flow parameters and can be absorbed into the constant of the reverse KLD loss.

3.1. Active Learning Workflow

We start with an initial dataset obtained from a short all-atom simulation. This is used to initially train the conditional normalizing flow by example (Equation 4). The starting dataset will typically only cover a small fraction of the full configurational space of the system.[4] Furthermore, the CG configurations $s = \xi(x)$ of the starting dataset are used as the initial dataset of the active learning workflow. Then, each active learning iteration consists of the following steps (see also Figure 1b):

1. We first train the conditional normalizing flow by energy (Equation 6). For the conditioning, we use the N high-error CG configurations added at the end of the previous AL iteration (or from the starting dataset in the first iteration). Additionally, in each epoch, we select $\gamma \cdot N$ with $\gamma = 0.3$ random CG configurations from older iterations. This stabilizes the flow in areas of previous iterations and gives flexibility to still adapt slightly if needed.

2. We then train an ensemble of 10 PMF models. The training strategy to obtain the PMF models is described in Section 3.3.

3. In the last step of each iteration, we sample points in the CG space that exceed a defined threshold of the ensemble standard deviation using Metropolis Monte Carlo (MC). We uniformly broaden the obtained high-error points by sampling uniformly in a hypersphere around them in CG space. The broadened points are added to the AL dataset, where 80 % are used for training, and 20 % as test samples.

After finishing each AL iteration, we stop the active learning workflow if the forward KLD of the PMF using the test dataset is smaller than a defined threshold or the trajectories of the MC sampling reach a specified maximum length. The normalizing flow is initialized only once at the beginning of the AL workflow and is then progressively updated during the AL iterations. The PMF ensemble is reinitialized every iteration.

Training a normalizing flow by energy to match the Boltzmann distribution of a molecular system can be difficult, since high-energy configurations, e.g., due to clashes between atoms, can yield very high loss values (Equation 6). This is especially problematic in the beginning of training, but also later, since the flow will never exactly resemble the Boltzmann distribution without any clashes. Previous works (Noé et al., 2019; Midgley et al., 2023b) mitigated the problem by using a regularized potential energy function that applies a logarithm above a threshold E_{high} and cuts off the energy above a threshold E_{max}. We found empirically that in our case removing a few of the highest loss values from each batch of samples yields more stable experiments than energy regularization. Whether this procedure

[3] If the potential energy is rotation- and translation-invariant, the orientation and translation do not have to be reconstructed.

[4] One can, of course, also use two starting datasets of distinct minima to find transition paths in the CG space.

also improves the training of non-conditional flows may be explored in future work.

3.2. Grid Conditioning

If the chosen CG space is low-dimensional, it is also possible to simply uniformly cover the CG space instead of running the proposed AL workflow (see discussion in Section 5). Therefore, we further show the results of training a conditional normalizing flow by energy on a grid in the CG space. In these experiments, we again first train by example on the starting dataset and then by energy using the CG configurations of the grid as conditioning.

3.3. Obtaining the Potential of Mean Force

Obtaining the PMF for coarse-grained simulations is typically done using force-matching or flow-matching as described in Section 2. Since the samples from the trained flow approximately follow the Boltzmann distribution, we could use force-matching to train our PMF models. In our scenario, however, since we have access to the probability distribution learned by the conditional normalizing flow $q_{X_{\mathrm{FG}}}(x_{\mathrm{FG}} \mid s; \theta)$, we can alternatively obtain the PMF values directly using its definition in Equation 7. We will discuss the advantages of this approach in Section 5. As we show in SI Section A.2, we can express the PMF using the following expectation value:

$$U_{\mathrm{PMF}}(s) = -k_{\mathrm{B}}T \ln \mathbb{E}_z \left[\frac{p_{X_{\mathrm{int}}}([g(z;s;\theta), s])}{q_{X_{\mathrm{FG}}}(g(z;s;\theta) \mid s;\theta)} \right], \tag{12}$$

$$p_{X_{\mathrm{int}}}(x_{\mathrm{int}}) \sim \exp\left(-\frac{E(x([g(z;s;\theta), s]))}{k_{\mathrm{B}}T} \right) |\det J_{x_{\mathrm{int}} \to x}| \tag{13}$$

This allows us to get the PMF $U_{\mathrm{PMF}}(s)$ for a given CG configuration s by sampling from the conditional normalizing flow and evaluating the potential $E(x)$ to obtain $p_{X_{\mathrm{int}}}(x_{\mathrm{int}})$. In order to obtain accurate PMF values, we only need sufficient overlap between the distribution of the conditional normalizing flow and the Boltzmann distribution, since Equation 12 includes a form of reweighting.

While training the flow by energy, we hold $k = 30$ copies of each CG configuration in the AL dataset. Each time one of the copies is selected in a batch when training the flow by energy, we store the evaluated potential energy and the probability $q_{X_{\mathrm{FG}}}(x_{\mathrm{FG}} \mid s; \theta)$ for that copy. This means that we always store the latest $k = 30$ samples for each CG configuration. This allows us to train an ensemble of PMF models at the end of each active learning iteration without making additional potential energy evaluations. Subsequently, this PMF ensemble can then be used to sample new high-error configurations for the next iteration.

We want to emphasize that it is also possible to train a PMF model without evaluating the expectation value in Equation 12 explicitly. This approach uses a surrogate loss function to match the free energies and can be derived using a generalization of the multiscale force-matching proof. It is discussed in more detail in SI Section A.2.2. In SI Section A.2.1 we further present an alternative to Equation 12 that has been previously used by Zhang et al. (2023). While we obtained the most accurate results with our approach, a systematic comparison of the two approaches is needed in future work.

4. Experiments

4.1. Müller-Brown Potential

We first test the AL workflow on the 2D Müller-Brown potential (Müller & Brown, 1979) (Figure 2 (top, contour lines)), starting the exploration in the global minimum. As a CG mapping, we use the 45° rotated coordinate axis $s = \xi(x) = x_1 - x_2$ (blue axis in Figure 2 (top)). The normalizing flow describes the conditional probability distribution $q_{S_\perp}(s_\perp \mid s; \theta)$, where $s_\perp = x_1 + x_2$ is the "fine-grained" coordinate orthogonal to s. This relatively simple setup serves as a first proof-of-concept of our methodology.

Since most coupling layers cannot directly be used to transform 1D probability densities, we do not use a conventional normalizing flow, but simply transform the latent distribution $z \sim \mathcal{N}(0, 1)$ using the following conditional transformation:

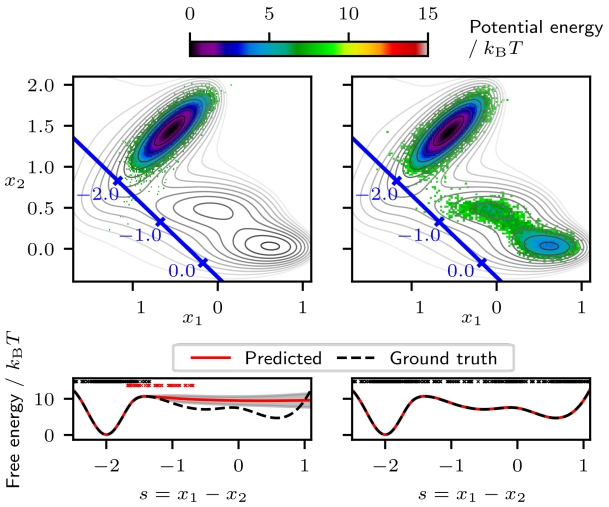
$$s_\perp = z \cdot \mathrm{NN}_{\mathrm{scale}}(s) + \mathrm{NN}_{\mathrm{mean}}(s) \tag{14}$$

Here, $\mathrm{NN}_{\mathrm{scale}}$ and $\mathrm{NN}_{\mathrm{mean}}$ are fully connected neural networks. This simple transformation suffices to obtain an accurate potential of mean force since our approach of obtaining it (Section 3.3) merely requires overlap of the distribution of the flow with the target distribution.

Figure 2 shows two exemplary iterations of an AL experiment in the Müller-Brown system. One can see that after the final iteration 8, the learned PMF (bottom, red) is almost identical to the ground-truth PMF (bottom, black).

Our AL experiments required $(1.13 \pm 0.04) \times 10^5$ potential energy evaluations with $(4.01 \pm 0.95) \times 10^5$ MC steps in the CG potential and obtained a forward KLD of the PMF of $(2.04 \pm 0.23) \times 10^{-4}$ (16 experiments were performed). We further compare the AL workflow with the conventional CG approach, where one first samples from the potential (here, using MC) and subsequently extracts a PMF. Since the CG space of the Müller-Brown system is only 1D, this is done using a histogram of the CG configurations in the

MC simulation[5]. We performed an MC simulation with 1×10^6 steps which yields a significantly larger KLD of the PMF of 1.02×10^{-2} compared to our AL experiments. This comes from poor convergence of the MC simulation since this simulation length only yields zero to a few transitions (more details in SI). The AL workflow learns the PMF more accurately using 10 times less potential energy evaluations compared to the "all-atom" MC simulation of 1×10^6 steps.



(a) Iteration 3 (b) Iteration 8 (final)

Figure 2. Visualization of two exemplary iterations of the AL workflow applied to the Müller-Brown system. Bottom: PMF and its standard deviation. Training points from previous AL iterations are marked as black "x" at the top of the PMF, and new high-error points added in the current AL iteration are marked as red "x". Top: Backmapped potential energy $\ln q_{S_\perp}(s_\perp \mid s; \theta) p^{\mathrm{CG}}(s)$ from cascaded sampling of the PMF and sampling of the flow. The blue axis represents the 1D CG coordinate s. The fine-grained coordinate $s_\perp$ is orthogonal to the CG coordinate. See Figure 6 in SI for a visualization of all iterations of this experiment.

4.2. Alanine Dipeptide

Now, we consider a more complex scenario where we explore the configurational space of the 22-atom molecule alanine dipeptide using the CG space $s = (\phi, \psi)^\top$ (see Figure 1 for a visualization of the molecule and definition of the dihedral angles ϕ and ψ). We use the same fully internal coordinate representation as Midgley et al. (2023b), where the molecule is described using $3 \cdot 22 - 6 = 60$ internal coordinates (bond distances, angles, and dihedral angles). This representation obeys the symmetries of the potential energy function since it is invariant to translations and rotations of the molecule.

The conditional normalizing flow parametrizes the probability distribution $q_{x_{\mathrm{FG}}}(x_{\mathrm{FG}} \mid s; \theta)$, where x_{FG} are the remaining 58 internal fine-grained degrees of freedom. Similarly to Midgley et al. (2023b) we use a normalizing flow built from 12 monotonic rational-quadratic spline coupling layers (Durkan et al., 2019) with residual networks as parameter networks (details in SI Section A.6.5). Dihedral angles of freely rotating bonds are treated as circular coordinates (Rezende et al., 2020; Midgley et al., 2023b). Furthermore, the circular coordinates in the input of the parameter network and the periodic conditioning variables ϕ and ψ of the parameter network are treated using the periodic representation $(\cos\eta, \sin\eta)^\top$ for each periodic variable η. During training, we further filter the chirality of each batch and only train on structures in the L-form (Midgley et al., 2023b).

Figure 3c-e shows the PMF for different iterations of an active learning experiment applied to alanine dipeptide. Figure 3b shows the PMF obtained using a grid conditioning experiment on a 100x100 grid in the CG space. As one can see visually, the resulting PMF after the last iteration of AL and the PMF from the grid conditioning experiment are not only identical to the ground truth PMF (Figure 3a), but in addition cover regions that were not sampled at all in the reference MD simulation. Especially in transition regions of high energy, e.g., at $\phi \approx 120°$, our obtained PMF is much more detailed and complete than the PMF from the ground truth MD dataset, where even after 2.3×10^{10} steps, large parts of the high-energy regions are completely missing.

We now consider a quantitative comparison of our approach with previous methods. Table 1 includes the results of Midgley et al. (2023b), where annealed importance sampling with α-divergence with $\alpha = 2$ (Flow AIS Bootstrap) was used to directly train a normalizing flow by energy to sample from the configurational space of alanine dipeptide. This is currently the only publication that succeeds in learning the full Boltzmann distribution of alanine dipeptide with a generative model without samples and without mode collapse. Furthermore, we include the results of training a normalizing flow with the reverse KLD (Midgley et al., 2023b), where mode collapse can be observed. Besides these ML approaches, we also include a comparison with a separate MD simulation for two MD simulation lengths. As the main metric to judge the accuracy of the PMF, we use the forward KLD of the PMF calculated on the test dataset.

In addition to the forward KLD of the PMF, we also provide metrics for the generated all-atom distribution: The all-atom log-likelihood $\mathbb{E}_{p_X(x)} \log q_X(x; \theta)$ and the reverse effective sample size (ESS) (see Section A.6.2 for details). We find the all-atom log-likelihood is similar across all methods (except for the flow with reverse KLD training, where mode collapse is observed), while Flow AIS Bootstrap (Midgley

[5]In higher-dimensional CG spaces one typically uses force-matching or flow-matching to obtain the PMF, as discussed in Section 2.

et al., 2023b) achieves better reverse ESS compared to our results. However, the main goal of this study is to obtain an accurate PMF model that can be used to run coarse-grained simulations. Therefore, we now focus on the forward KLD of the PMF.

Next to Table 1, the forward KLD of the PMF as a function of the number of potential energy evaluations is further visualized in Figure 4. We find that the forward KLD of the PMF as a function of the number of potential energy evaluations of our methods decreases substantially faster than Flow AIS Bootstrap and MD. Compared to the Flow AIS Bootstrap (Midgley et al., 2023b) results, our AL workflow uses approximately one order of magnitude less potential energy evaluations while obtaining a substantially smaller KLD. The experiments with grid conditioning show similar results while reaching the final Flow AIS Bootstrap KLD almost two orders of magnitude faster. When running MD simulations, the KLD is significantly higher, even when using $2-3$ orders of magnitude more potential energy evaluations (1×10^9) (see also Figure 4). As discussed before, our method yields (accurate) PMF maps also in regions where no ground truth data is available. This makes the quantification of the increase in accuracy difficult. Therefore, we quantify the speedup of our method at a fixed accuracy (KLD value): A KLD of 2.5×10^{-3} (Flow AIS Bootstrap) is reached after approximately 9.1×10^8 MD steps (reference), 2×10^8 steps in Flow AIS Bootstrap (speedup of 4.5), 2.9×10^7 potential energy evaluations + 2.8×10^7 CG MC steps (speedup of 15.9), and 4.2×10^6 grid conditioning steps (speedup of 216.2). Here, for the AL workflow, we generously counted the cost of one CG step to be equal to one all-atom potential energy evaluation. We discuss the sampling efficiency in the CG space in more detail in Section 5. We assume that the accuracy reached by our methods is higher than that of the available ground truth data, making the aforementioned numbers lower bounds of the actual speedup values.

We note that at the accuracies we are achieving here, the reported KLD is not anymore a representative measure of the increasing accuracy of the PMF, because (a) it is logarithmic so the accuracy of the PMF in high-energy regions is influencing the KLD in a negative exponential way, and (b) we do not have any good ground truth data as reference in the KLD, specifically in those high energy regions. That means we cannot quantitatively measure if our PMF becomes more accurate than the PMF shown in Figure 3a, while visual inspection of Figure 3e and 3b clearly indicates that we achieve substantially improved PMF maps compared to the MD-derived ground truth, at a fraction of the cost.

5. Discussion

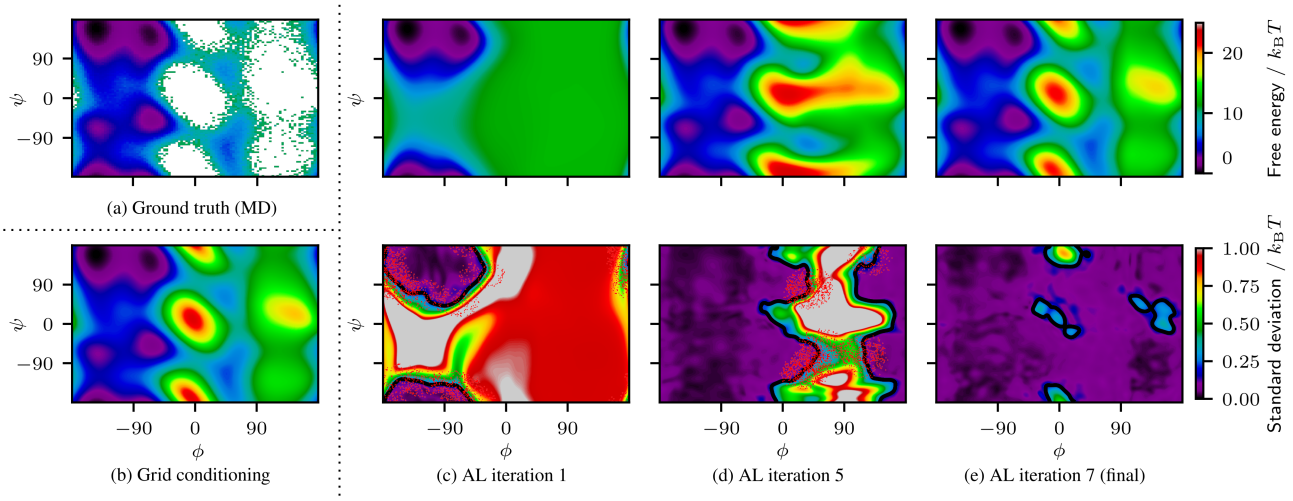
Methods such as all-atom MD or normalizing flows trained by energy struggle with correctly describing rare transition regions of molecular systems, since they are not sampled frequently in the Boltzmann distribution. Our approach of using a normalizing flow conditioned on CG coordinates circumvents this problem due to the conditional sampling of the Boltzmann distribution. This allows the correct description of the PMF even in very high-energy regions. Furthermore, using the proposed AL workflow, the configurational space can be explored more efficiently in the lower-dimensional smoother PMF of the CG space compared to all-atom simulations.

We need to emphasize that in the low-dimensional CG spaces of the two systems that our work covers, one can simply sample the CG coordinates on a grid, as shown in Table 1 for alanine dipeptide. In higher-dimensional CG spaces, other means of sampling the CG space instead of using a grid become strictly necessary, e.g., using Langevin dynamics or Metropolis Monte Carlo-based exploration. As discussed in detail in SI Section A.6.8, already for our example alanine dipeptide - where implicit solvation energy and force evaluations are very fast - the surrogate PMF model can be sampled faster and thus more efficiently than the all-atom level.

In the future, we envision our approach to be used for more complex systems and CG mappings. This includes conventional CG mappings, such as using only the backbone atoms of a molecule as CG coordinates, where Langevin dynamics can be used to sample in the CG space and find new high-error configurations. While the transition barriers of alanine dipeptide are relatively low and it is possible to sample sufficiently with all-atom MD, applying the active learning methodology to systems with large energy barriers will allow efficient exploration of the configurational space in a relatively low-dimensional space. When using density functional theory or an active-learned ML surrogate thereof on the all-atom level, sampling the lower-dimensional PMF will be much faster than sampling the all-atom level.

Scaling to higher-dimensional problems, such as using C_α CG mappings of proteins, will yield new challenges. Since the reverse KLD is mode-seeking, it can only be used to learn distributions that are approximately unimodal. It needs to be seen if the conditional side-chain distribution of proteins and other systems of interest fulfills this constraint. Potentially, other optimization objectives that are not mode-seeking, such as the α-divergence with $\alpha = 2$ used by Midgley et al. (2023b), need to be used instead of the reverse KLD.

Also non-conventional, nonlinear CG mappings such as folding coordinates for proteins or known reaction coordi-



(a) Ground truth (MD) (b) Grid conditioning (c) AL iteration 1 (d) AL iteration 5 (e) AL iteration 7 (final)

Figure 3. (a) Ground truth PMF of dihedral angles ϕ and ψ of alanine dipeptide from MD test dataset with 2.3×10^{10} steps. (b) PMF from grid conditioning experiment after 2.4×10^7 steps. (c-e) Three exemplary AL steps for the alanine dipeptide system. Top: PMF of ϕ and ψ at the end of the AL iteration. Bottom: Standard deviation of the PMF. The contour line of the threshold in the standard deviation when sampling new points ($0.2\, k_\mathrm{B}T$) is drawn using a black line. Newly added points after sampling using the PMF of this iteration are shown as red dots.

Table 1. Results of the experiments with alanine dipeptide. We compare the performance of different approaches: A normalizing flow trained with the reverse KLD (Midgley et al., 2023b), Flow AIS Bootstrap with buffer (Midgley et al., 2023b), MD simulations of two different lengths, our active learning workflow, and our grid conditioning experiments - once fully converged and once with fewer number of evaluations. As the main metric to estimate the accuracy of the PMF we use the forward KLD of the PMF calculated on the test dataset. Additionally, we provide metrics for the generated all-atom distribution: The all-atom log-likelihood $\mathbb{E}_{p_X(x)} \log q_X(x;\theta)$ and the reverse effective sample size (ESS). For the given metrics, ↑ indicates higher is better, ↓ indicates lower is better. For our experiments, we provide the mean and standard error over 8 experiments. Note that the AL experiments required, additionally to the all-atom potential energy evaluations, $(3.01 \pm 0.83) \times 10^7$ MC steps.

METHOD	POT. ENERGY EVALUATIONS ↓	FORWARD KLD PMF ↓	ALL-ATOM METRICS ↑	
			$\mathbb{E}_{p_X(x)} \log q_X(x;\theta)$	REVERSE ESS / %
FLOW WITH REVERSE KLD	2.5×10^8	3.15 ± 0.19	100 ± 32	54 ± 12
FLOW AIS BOOTSTRAP	2×10^8	$(2.51 \pm 0.39) \times 10^{-3}$	**211.54 ± 0.00**	**92.8 ± 0.1**
MD (LONG)	1×10^9	2.32×10^{-3}	-	-
MD (SHORT)	1×10^8	1.87×10^{-2}	-	-
AL (OURS)	$(3.35 \pm 0.10) \times 10^7$	$(9.29 \pm 0.24) \times 10^{-4}$	211.18 ± 0.04	43.83 ± 3.81
GRID COND. (OURS, CONVERGED)	2.40×10^7	**$(6.32 \pm 0.28) \times 10^{-4}$**	211.49 ± 0.01	83.92 ± 1.15
GRID COND. (OURS, SHORT)	**4.85×10^6**	$(1.68 \pm 0.13) \times 10^{-3}$	210.93 ± 0.02	25.58 ± 1.02

nates in chemical reactions might be interesting for efficient sampling of the configurational space using our proposed methodology - either with active learning or using the grid conditioning strategy. This can yield accurate PMF profiles along collective variables of interest. Zhang et al. (2023) already showed first results in this direction, though performance comparisons and implementation details are missing. If the all-atom coordinates cannot be directly reconstructed in such a way that it obeys the conditioning (which is the case for most nontrivial nonlinear reaction coordinates), one needs to introduce an additional consistency loss term to make sure that $\xi(x)$ of the generated configurations is equal to the conditioning s (see (Zhang et al., 2023)).

To scale the shown methodology to larger systems in the future, some recent advances in the architecture of normalizing flows can be used (Midgley et al., 2023a; Köhler et al., 2020; Garcia Satorras et al., 2021; Draxler et al., 2023), which includes continuous normalizing flows with equivariant graph neural networks. Moving away from the here-used internal coordinate representation further allows the usage of one conditional normalizing flow across multiple molecular systems. This would allow the transfer of a pre-trained

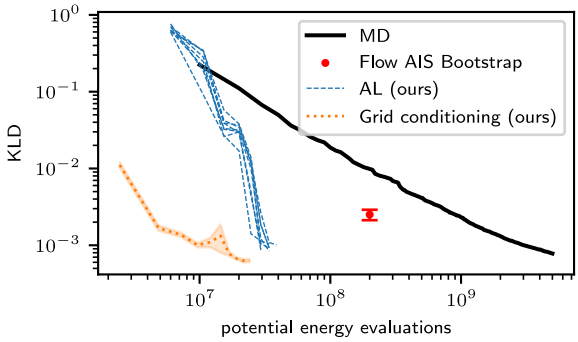


Figure 4. Forward KLD of the PMF of alanine dipeptide as a function of the number of potential energy evaluations. For grid conditioning and Flow AIS Bootstrap we additionally show the standard error.

normalizing flow and CG potential to novel systems, where the active learning workflow can be used to further refine the CG potential when necessary.

Besides the application in the active learning workflow, the approach of learning the conditional probability $p_{X_{\text{FG}}}(x_{\text{FG}} \mid s)$ and then extracting the PMF (see Section 3.3) might have advantages in data efficiency over both multiscale force-matching and flow-matching even in the non-active-learning scenario (which requires a comprehensive all-atom MD dataset). Further details can be found in the SI in Section A.4.

6. Conclusion

In this work, we showed how conditional normalizing flows can be used to build an active learning workflow for coarse-grained simulations. When testing our method with the 22-atom molecule alanine dipeptide, we demonstrated a speedup to molecular dynamics simulations of approximately 15.9 to 216.2 compared to the speedup of 4.5 of the current state-of-the-art ML approach by Midgley et al. (2023b). We obtain a higher accuracy using approximately an order of magnitude less potential energy evaluations compared to Midgley et al. (2023b). Compared to performing all-atom MD simulations to extract the PMF we achieve higher accuracy with two orders of magnitude less potential energy evaluations. Since all current ML coarse-graining approaches require such an extensive MD simulation to train the PMF, our active learning approach outperforms them in terms of data efficiency.

We see our work as a first demonstration of the potential that lies in performing active learning in the coarse-grained space of molecular systems. For the first time, this is possible without first needing a long all-atom trajectory or constrained MD simulations. Our results show that if one can construct a (potentially nonlinear) lower-dimensional CG space that includes the main modes of a molecular system, a conditional normalizing flow can efficiently learn this conditional probability distribution - either with uniform coverage of the CG space if it is low-dimensional, or with an explorative active learning approach for higher-dimensional CG spaces. We are confident that this will boost the utility and application of machine-learned coarse-grained potentials and hope to see further developments in this direction.

Data and Code Availability

Our reference implementation of the described active learning workflow can be found on `https://github.com/aimat-lab/coarse-graining-AL` (v1.0). Code to reproduce all experiments is provided.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. H.S. acknowledges financial support by the German Research Foundation (DFG) through the Research Training Group 2450 “Tailored Scale-Bridging Approaches to Computational Nanoscience”. P.F. acknowledges funding from the Klaus Tschira Stiftung gGmbH and the pilot program Core-Informatics of the Helmholtz Association (HGF). The authors acknowledge support by the state of Baden-Württemberg through bwHPC. Parts of this work were performed on the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Ardizzone, L., Lüth, C., Kruse, J., Rother, C., and Köthe, U. Guided Image Generation with Conditional Invertible Neural Networks, July 2019.

Charron, N. E., Musil, F., Guljas, A., Chen, Y., Bonneau, K., Pasos-Trejo, A. S., Venturin, J., Gusew, D., Zaporozhets, I., Krämer, A., Templeton, C., Kelkar, A., Durumeric, A. E. P., Olsson, S., Pérez, A., Majewski, M., Husic, B. E., Patel, A., De Fabritiis, G., Noé, F., and Clementi, C. Navigating protein landscapes with a machine-learned transferable coarse-grained model, October 2023.

D.A. Case, H.M. Aktulga, K. Belfon, I.Y. Ben-Shalom, J.T. Berryman, S.R. Brozell, D.S. Cerutti, T.E. Cheatham, III, V.W.D. Cruzeiro, T.A. Darden, N. Forouzesh, G. Giambasu, T. Giese, M.K. Gilson, H. Gohlke, A.W. Goetz, J. Harris, S. Izadi, S.A. Izmailov, K. Kasavajhala, M.C. Kaymak, E. King, A. Kovalenko, T. Kurtzman, T.S. Lee, P. Li, C. Lin, J. Liu, T. Luchko, R. Luo, M. Machado, V. Man, M. Manathunga, K.M. Merz, Y. Miao, O. Mikhailovskii, G. Monard, H. Nguyen, K.A. O'Hearn, A. Onufriev, F. Pan, S. Pantano, R. Qi, A. Rahnamoun, D.R. Roe, A. Roitberg, C. Sagui, S. Schott-Verdugo, A. Shajan, J. Shen, C.L. Simmerling, N.R. Skrynnikov, J. Smith, J. Swails, R.C. Walker, J. Wang, J. Wang, H. Wei, X. Wu, Y. Wu, Y. Xiong, Y. Xue, D.M. York, S. Zhao, Q. Zhu, and P.A. Kollman. Amber 2023. University of California, San Francisco, 2023.

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

Dinh, L., Krueger, D., and Bengio, Y. NICE: Non-linear Independent Components Estimation, April 2015.

Draxler, F., Sorrenson, P., Zimmermann, L., Rousselot, A., and Köthe, U. Free-form Flows: Make Any Architecture a Normalizing Flow, October 2023.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural Spline Flows, December 2019.

Duschatko, B. R., Vandermause, J., Molinari, N., and Kozinsky, B. Uncertainty driven active learning of coarse grained free energy models. *npj Comput Mater*, 10(1):1–12, January 2024. ISSN 2057-3960. doi: 10.1038/s41524-023-01183-5.

Eastman, P., Swails, J., Chodera, J. D., McGibbon, R. T., Zhao, Y., Beauchamp, K. A., Wang, L.-P., Simmonett, A. C., Harrigan, M. P., Stern, C. D., Wiewiora, R. P., Brooks, B. R., and Pande, V. S. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLOS Computational Biology*, 13(7):e1005659, July 2017. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1005659.

Felardos, L., Hénin, J., and Charpiat, G. Designing losses for data-free training of normalizing flows on Boltzmann distributions, January 2023.

Garcia Satorras, V., Hoogeboom, E., Fuchs, F., Posner, I., and Welling, M. E(n) Equivariant Normalizing Flows. In *Advances in Neural Information Processing Systems*, volume 34, pp. 4181–4192. Curran Associates, Inc., 2021.

Husic, B. E., Charron, N. E., Lemm, D., Wang, J., Pérez, A., Majewski, M., Krämer, A., Chen, Y., Olsson, S., de Fabritiis, G., Noé, F., and Clementi, C. Coarse graining molecular dynamics with graph neural networks. *J Chem Phys*, 153(19):194101, November 2020. ISSN 0021-9606. doi: 10.1063/5.0026133.

Izvekov, S. and Voth, G. A. A Multiscale Coarse-Graining Method for Biomolecular Systems. *J. Phys. Chem. B*, 109 (7):2469–2473, February 2005. ISSN 1520-6106. doi: 10.1021/jp044629q.

Jin, J., Pak, A. J., Durumeric, A. E. P., Loose, T. D., and Voth, G. A. Bottom-up Coarse-Graining: Principles and Perspectives. *J. Chem. Theory Comput.*, 18(10):5759–5791, October 2022. ISSN 1549-9618. doi: 10.1021/acs.jctc.2c00643.

Jing, B., Corso, G., Chang, J., Barzilay, R., and Jaakkola, T. S. Torsional Diffusion for Molecular Conformer Generation. In *Advances in Neural Information Processing Systems*, October 2022.

Kalligiannaki, E., Harmandaris, V., Katsoulakis, M. A., and Plecháč, P. The geometry of generalized force matching and related information metrics in coarse-graining of molecular systems. *The Journal of Chemical Physics*, 143(8):084105, August 2015. ISSN 0021-9606. doi: 10.1063/1.4928857.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization, January 2017.

Koblents, E. and Míguez, J. A population Monte Carlo scheme with transformed weights and its application to stochastic kinetic models. *Stat Comput*, 25(2):407–425, March 2015. ISSN 1573-1375. doi: 10.1007/s11222-013-9440-2.

Köhler, J., Klein, L., and Noé, F. Equivariant flows: Exact likelihood generative learning for symmetric densities. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20, pp. 5361–5370. JMLR.org, July 2020.

Köhler, J., Chen, Y., Krämer, A., Clementi, C., and Noé, F. Flow-Matching: Efficient Coarse-Graining of Molecular Dynamics without Forces. *J. Chem. Theory Comput.*, 19(3):942–952, February 2023. ISSN 1549-9618. doi: 10.1021/acs.jctc.3c00016.

Liwo, A., Czaplewski, C., Sieradzan, A. K., Lubecka, E. A., Lipska, A. G., Golon, Ł., Karczyńska, A., Krupa, P., Mozolewska, M. A., Makowski, M., Ganzynkowicz, R., Giełdoń, A., and Maciejczyk, M. Scale-consistent approach to the derivation of coarse-grained force fields for simulating structure, dynamics, and thermodynamics

of biopolymers. *Prog Mol Biol Transl Sci*, 170:73–122, 2020. ISSN 1878-0814. doi: 10.1016/bs.pmbts.2019.12.004.

Mahmoud, A. H., Masters, M., Lee, S. J., and Lill, M. A. Accurate Sampling of Macromolecular Conformations Using Adaptive Deep Learning and Coarse-Grained Representation. *J. Chem. Inf. Model.*, 62(7):1602–1617, April 2022. ISSN 1549-9596. doi: 10.1021/acs.jcim.1c01438.

Majewski, M., Pérez, A., Thölke, P., Doerr, S., Charron, N. E., Giorgino, T., Husic, B. E., Clementi, C., Noé, F., and De Fabritiis, G. Machine learning coarse-grained potentials of protein thermodynamics. *Nat Commun*, 14(1):5739, September 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-41343-1.

Marrink, S. J., Monticelli, L., Melo, M. N., Alessandri, R., Tieleman, D. P., and Souza, P. C. T. Two decades of Martini: Better beads, broader scope. *WIREs Computational Molecular Science*, pp. e1620, 2023. ISSN 1759-0884. doi: 10.1002/wcms.1620.

Martino, L., Elvira, V., and Louzada, F. Effective Sample Size for Importance Sampling based on discrepancy measures. *Signal Processing*, 131:386–401, February 2017. ISSN 01651684. doi: 10.1016/j.sigpro.2016.08.025.

Midgley, L. I., Stimper, V., Antorán, J., Mathieu, E., Schölkopf, B., and Hernández-Lobato, J. M. SE(3) Equivariant Augmented Coupling Flows. In *Thirty-Seventh Conference on Neural Information Processing Systems*, August 2023a.

Midgley, L. I., Stimper, V., Simm, G. N. C., Schölkopf, B., and Hernández-Lobato, J. M. Flow Annealed Importance Sampling Bootstrap. In *The Eleventh International Conference on Learning Representations*, 2023b. doi: 10.48550/arXiv.2208.01893.

Müller, K. and Brown, L. D. Location of saddle points and minimum energy paths by a constrained simplex optimization procedure. *Theoret. Chim. Acta*, 53(1):75–93, March 1979. ISSN 1432-2234. doi: 10.1007/BF00547608.

Nicoli, K. A., Anders, C. J., Funcke, L., Hartung, T., Jansen, K., Kessel, P., Nakajima, S., and Stornati, P. Estimation of Thermodynamic Observables in Lattice Field Theories with Deep Generative Models. *Phys. Rev. Lett.*, 126(3): 032001, January 2021. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.126.032001.

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

Noid, W. G., Chu, J.-W., Ayton, G. S., Krishna, V., Izvekov, S., Voth, G. A., Das, A., and Andersen, H. C. The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models. *J. Chem. Phys.*, 128(24):244114, June 2008. ISSN 0021-9606. doi: 10.1063/1.2938860.

Rezende, D. J., Papamakarios, G., Racaniere, S., Albergo, M., Kanwar, G., Shanahan, P., and Cranmer, K. Normalizing Flows on Tori and Spheres. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 8083–8092. PMLR, November 2020.

Spiriti, J., Kamberaj, H., and Van Der Vaart, A. Development and application of enhanced sampling techniques to simulate the long-time scale dynamics of biomolecular systems. *International Journal of Quantum Chemistry*, 112(1):33–43, 2012. ISSN 1097-461X. doi: 10.1002/qua.23139.

Stimper, V., Midgley, L. I., Simm, G. N. C., Schölkopf, B., and Hernández-Lobato, J. M. Alanine dipeptide in an implicit solvent at 300K. August 2022. doi: 10.5281/zenodo.6993124.

Stimper, V., Liu, D., Campbell, A., Berenz, V., Ryll, L., Schölkopf, B., and Hernández-Lobato, J. M. Normflows: A PyTorch Package for Normalizing Flows. *JOSS*, 8(86): 5361, June 2023. ISSN 2475-9066. doi: 10.21105/joss.05361.

Thaler, S., Stupp, M., and Zavadlav, J. Deep coarse-grained potentials via relative entropy minimization. *The Journal of Chemical Physics*, 157(24):244103, December 2022. ISSN 0021-9606. doi: 10.1063/5.0124538.

Vaitl, L., Nicoli, K. A., Nakajima, S., and Kessel, P. Gradients should stay on path: Better estimators of the reverse- and forward KL divergence for normalizing flows. *Mach. Learn.: Sci. Technol.*, 3(4):045006, October 2022. ISSN 2632-2153. doi: 10.1088/2632-2153/ac9455.

Wang, J., Olsson, S., Wehmeyer, C., Pérez, A., Charron, N. E., de Fabritiis, G., Noé, F., and Clementi, C. Machine Learning of Coarse-Grained Molecular Dynamics Force Fields. *ACS Cent. Sci.*, 5(5):755–767, May 2019. ISSN 2374-7943. doi: 10.1021/acscentsci.8b00913.

Winkler, C., Worrall, D., Hoogeboom, E., and Welling, M. Learning Likelihoods with Conditional Normalizing Flows, November 2023.

Xiao, Z., Yan, Q., and Amit, Y. A Method to Model Conditional Distributions with Normalizing Flows, November 2019.

Yang, W., Templeton, C., Rosenberger, D., Bittracher, A., Nüske, F., Noé, F., and Clementi, C. Slicing and Dicing: Optimal Coarse-Grained Representation to Preserve Molecular Kinetics. *ACS Cent. Sci.*, January 2023. ISSN 2374-7943. doi: 10.1021/acscentsci.2c01200.

Zhang, J., Lin, X., E, W., and Gao, Y. Q. Machine-Learned Invertible Coarse Graining for Multiscale Molecular Modeling, May 2023.

Zhang, L., Han, J., Wang, H., Car, R., and E, W. DeePCG: Constructing coarse-grained models via deep neural networks. *J. Chem. Phys.*, 149(3):034101, July 2018. ISSN 0021-9606. doi: 10.1063/1.5027645.

A. Appendix

A.1. Application to New Problems

In this manuscript, we focused on the application of our coarse-graining approach to the sampling of molecular tasks. However, the described workflow (both the active learning and the grid conditioning approach) can be applied to any sampling problem of an unnormalized probability density, where a meaningful CG space that includes the main modes of the distribution can be defined. For example, our method could also be applied to sampling problems of lattice field theories, such as the ϕ^4 theory (Nicoli et al., 2021).

The main steps to apply our workflow to a new system are the following:

- Implement the target log probability (energy) function and the CG mapping function (see *System* class in our source code)
- Decide upon a normalizing flow architecture, such as RNVP or rational-quadratic splines
- Implement a PMF model architecture that incorporates potential symmetries of the CG space
- Generate a small starting dataset using Metropolis Monte Carlo or other sampling techniques
- If needed, update the active learning sampler that is used to obtain new high-error configurations. For example, when applying our method to higher-dimensional molecular systems with backbone CG mappings, one might consider Langevin dynamics to sample the CG space.
- Update config options to match the system details

A.2. Obtaining the Potential of Mean Force

Here, we derive Equation 12 of the main text which we use to train the PMF models.

We start with the definition of the potential of mean force:

$$
\begin{aligned}
U_{\text{PMF}}(s) &= -k_{\text{B}}T \ln \left[\int_x \exp\left(-\beta E(x)\right) \delta\left(s - \xi(x)\right) \, \mathrm{d}x \right] + C_0 && (15)\\
&\underset{C_0=0}{=} -k_{\text{B}}T \ln \left[\int_x Z \cdot p_X(x) \delta\left(s - \xi(x)\right) \, \mathrm{d}x \right] && (16)\\
&= -k_{\text{B}}T \ln \left[\int_x \frac{q_{X_{\text{FG}}}\left(x_{\text{FG}} \mid s; \theta\right)}{q_{X_{\text{FG}}}\left(x_{\text{FG}} \mid s; \theta\right)} p_X(x) \delta\left(s - \xi(x)\right) \, \mathrm{d}x \right] + C_1 && (17)\\
&\underset{C_1=0}{=} -k_{\text{B}}T \ln \left[\int_z q_Z(z) \frac{1}{q_{x_{\text{FG}}}\left(g(z; s; \theta) \mid s; \theta\right)} p_{X_{\text{int}}}([g(z; s; \theta), s]) \, \mathrm{d}z \right] && (18)\\
&= -k_{\text{B}}T \ln \mathbb{E}_{z \sim q_Z} \underbrace{\left[\frac{p_{X_{\text{int}}}\left([g(z; s; \theta), s]\right)}{q_{x_{\text{FG}}}\left(g(z; s; \theta) \mid s; \theta\right)} \right]}_{\equiv G_1(s,z)} && (19)
\end{aligned}
$$

Here, $p_{X_{\text{int}}}(x_{\text{int}}) \sim \exp\left(-\frac{E(x(x_{\text{int}}))}{k_{\text{B}}T}\right) \left|\det J_{x_{\text{int}} \to x}\right|$.

If the distribution of the conditional normalizing flow sufficiently overlaps with the target Boltzmann distribution, this expectation value can be used to obtain accurate PMF values for training of the PMF ensemble.

A.2.1. Alternative Formulation

Here, we describe an alternative to Equation 12 to obtain the PMF from the conditional normalizing flow. Again, we start with the definition of the PMF:

$$U_{\text{PMF}}(s) = -k_{\text{B}}T \ln \left[\underbrace{\int_x \exp\left(-\beta E(x)\right) \delta\left(s - \xi(x)\right) \, \mathrm{d}x}_{\equiv Z(s)} \right] + C \underset{C=0}{=} -k_{\text{B}}T \ln Z(s) \tag{20}$$

$$= \int_x \underbrace{\delta\left(s - \xi(x)\right) \frac{\exp(-\beta E(x))}{Z(s)}}_{p(x|s)} E(x) \, \mathrm{d}x + k_{\text{B}}T \int_x \underbrace{\delta\left(s - \xi(x)\right) \frac{\exp(-\beta E(x))}{Z(s)}}_{p(x|s)} \ln \left[\underbrace{\frac{\exp(-\beta E(x))}{Z(s)}}_{p(x|s)} \right] \mathrm{d}x \tag{21}$$

$$= \int_x p(x \mid s) E(x) \, \mathrm{d}x + k_{\text{B}}T \int_x p(x \mid s) \ln p(x \mid s) \, \mathrm{d}x \tag{22}$$

$$= \int_{x_{\text{int}}} p(x_{\text{int}} \mid s) E(x(x_{\text{int}})) \, \mathrm{d}x_{\text{int}} + k_{\text{B}}T \int_{x_{\text{int}}} p(x_{\text{int}} \mid s) \ln p(x(x_{\text{int}}) \mid s) \, \mathrm{d}x_{\text{int}} \tag{23}$$

$$\approx \int_z q_Z(z) E(x(\underbrace{[g(z; s; \theta), s]}_{x_{\text{int}}})) \, \mathrm{d}z + k_{\text{B}}T \int_z q_Z(z) \ln \left[q_{X_{\text{FG}}}(g(z; s; \theta) \mid s; \theta) \cdot |\det J_{x \to x_{\text{int}}}| \right] \mathrm{d}z \tag{24}$$

$$= \mathbb{E}_{z \sim q_Z} \underbrace{\left[E(x(\underbrace{[g(z; s; \theta), s]}_{x_{\text{int}}})) + k_{\text{B}}T \ln \left(q_{X_{\text{FG}}}(g(z; s; \theta) \mid s; \theta) \cdot |\det J_{x \to x_{\text{int}}}| \right) \right]}_{\equiv G_2(s,z)} \tag{25}$$

This formula is the same as used by Zhang et al. (2023). In the same way as Equation 12, it allows us to estimate the PMF $U_{\text{PMF}}(s)$ of a CG configuration s using the conditional normalizing flow $q_{X_{\text{FG}}}\left(x_{\text{FG}} \mid s; \theta\right)$. In practice, we found that this alternative formula does not work as well as using Equation 12 directly. However, further comparisons of the different approaches need to be done in the future.

A.2.2. Surrogate Loss

Here, we show that it is also possible to obtain the PMF without explicitly evaluating the expectation value in Equation 12 or Equation 25. When using a mean squared error loss, we can show that we will obtain the correct PMF even without the contraction along z. This is analogous to how we can estimate the expectation value of the projected all-atom forces in multiscale force-matching using a mean squared error surrogate loss. This approach can be applied to both the expectation value in Equation 12 and the alternative version in Equation 25.

In general, we want to obtain the following expectation value:

$$H_i(s) = \mathbb{E}_{z \sim q_Z} G_i(s, z) \tag{26}$$

Then, $U_{\text{PMF}}(s) = -k_{\text{B}}T \ln H_1(s)$ for version 1 and $U_{\text{PMF}}(s) = H_2(s)$ for the alternative version 2.

We can formulate a surrogate loss to train a model $H(s; W)$ with parameters W to match the expectation value:

$$\chi(W) = \langle [G_i(s, z) - H(s; W)]^2 \rangle_{s,z} \tag{27}$$

$$= \langle [G_i(s, z) - \langle G_i(s, z) \rangle_z + \langle G_i(s, z) \rangle_z - H(s; W)]^2 \rangle_{s,z} \tag{28}$$

$$= \underbrace{\langle [G_i(s, z) - \langle G_i(s, z) \rangle_z]^2 \rangle_{s,z}}_{\equiv \chi_{\text{Noise}}} + \underbrace{\langle [\langle G_i(s, z) \rangle_z - H(s; W)]^2 \rangle_{s,z}}_{\equiv \chi_{\text{Expectation}}} \tag{29}$$

$$+ \underbrace{2 \langle (G_i(s, z) - \langle G_i(s, z) \rangle_z)(\langle G_i(s, z) \rangle_z - H(s; W)) \rangle_{s,z}}_{\equiv \chi_{\text{Mixed}}} \tag{30}$$

As one can see, the surrogate loss χ can be decomposed into three parts: First, there is a noise term χ_{Noise}, which is independent of the free energy model parameters W. This is equivalent to the noise term that we find in the multiscale force-matching objective (Wang et al., 2019). Furthermore, there is our main objective $\chi_{\text{Expectation}}$. This is the loss term that we actually need, as it matches $H(s; W)$ onto the expectation value $\langle G_i(s, z)\rangle_z$.

We now have to show that the mixed term χ_{mixed} is zero, which is possible because the distribution of z is independent of s:

$$\frac{\chi_{\text{mixed}}}{2} = \langle\langle G_i(s,z)\langle G_i(s,z)\rangle_z\rangle_z\rangle_s - \langle\langle\langle G_i(s,z)\rangle_z\langle G_i(s,z)\rangle_z\rangle_z\rangle_s \tag{31}$$

$$- \langle\langle G_i(s,z)H(s;W)\rangle_z\rangle_s + \langle\langle\langle G_i(s,z)\rangle_z H(s;W)\rangle_z\rangle_s \tag{32}$$

$$= 0 \tag{33}$$

As one can see, we can obtain the PMF even when not explicitly contracting along z. In practice, we found that contraction should still be done if possible, as it lowers the noise significantly and makes training much easier. However, in scenarios in which one does not want to evaluate the same CG configuration in the flow multiple times, one can still use the mean squared error to obtain the correct PMF.

A.3. Invertible Neural Networks: Coupling Layers

As discussed in the main text, a normalizing flow requires an invertible function approximator (invertible neural network, INN), which can be constructed using a stack of coupling layers. Here, the dimensions of the input $x_{1:D}$ are split into two parts, $x_{1:d}$ and $x_{d+1:D}$ (usually, this splitting is done using a random mask). The first part is transformed elementwise conditioned on the second part, while the second part is unchanged (Midgley et al., 2023a):

$$x'_{1:d} = B\left(x_{1:d}; x_{d+1:D}\right), \tag{34}$$

$$x'_{d+1:D} = x_{d+1:D}. \tag{35}$$

If B is invertible (monotonic) and since $x_{d+1:D}$ is unchanged and thus given when transforming forward and backward, this yields an invertible architecture. The Jacobian matrix of such a coupling transform is lower triangular and thus the Jacobian determinant can be efficiently computed using the diagonal elements of the Jacobian matrix (Durkan et al., 2019).

A.4. Comparison with Multiscale Force-Matching and Flow-Matching

Besides the active learning workflow, we want to shortly discuss a possible application of the here-described approach to train PMF models in the non-active-learning scenario, where training data from MD trajectories is present. Here, our approach of learning $p_{X_{\text{FG}}}(x_{\text{FG}} \mid s)$ and then extracting the PMF (see Section 3.3) has potential advantages over both multiscale force-matching and flow-matching.

Multiscale force-matching requires large amounts of training data since transition regions need to be sufficiently covered to get a good gradient estimate. If this is not the case, distinct minima on the PMF surface appear biased in energy. Flow matching mitigates this problem by not working with solely gradient information. $p^{\text{CG}}(s)$ is directly (or indirectly in the case of student-teacher training (Köhler et al., 2023)) learned from the distribution of the training data in the CG space. However, this requires a well-converged MD trajectory where different minima are correctly occupied. Otherwise, the flow will learn this wrong occupation.

In our here-described approach, both the shortcomings of multiscale force-matching and flow-matching are addressed. By learning $p_{X_{\text{FG}}}(x_{\text{FG}} \mid s)$ from the training data, one does not depend on gradient information alone. Furthermore, since $p^{\text{CG}}(s)$ is not directly obtained from the occupation of the different minima in the dataset, the occupations do not have to be fully converged in the trajectory. We only need enough training data in the regions of interest (where the PMF should be estimated) to learn $p_{X_{\text{FG}}}(x_{\text{FG}} \mid s)$. In this way, one can use our approach in the non-active-learning scenario, where training data in the form of an MD trajectory is present. Closer investigation and comparison of the data efficiencies and accuracies between the different approaches is not the focus of this work and should be investigated in future work.

A.5. Müller-Brown Potential

A.5.1. Potential Energy

The Müller-Brown potential is defined through the following formula (Müller & Brown, 1979) (see Figure 5 for a visualization):

$$E_{\text{MB}}(x_1, x_2) = \sum_{i=1}^{4} A_i \exp\Big[a_i\,(x_1 - \bar{x}_i)^2 + b_i\,(x_1 - \bar{x}_i)\,(x_2 - \bar{y}_i) + c_i\,(x_2 - \bar{y}_i)^2\Big] \tag{36}$$

$$\text{with} \quad A = (-200, -100, -170, 15); \quad a = (-1, -1, -6.5, 0.7) \tag{37}$$

$$b = (0, 0, 11, 0.6); \quad c = (-10, -10, -6.5, 0.7) \tag{38}$$

$$\bar{x} = (1, 0, -0.5, -1); \quad \bar{y} = (0, 0.5, 1.5, 1). \tag{39}$$

We use $\beta = \frac{1}{k_\text{B}T} = 0.1$ to evaluate the PMF and run the MC sampling of the AL workflow.

Since the Müller-Brown potential is only 2D, the ground-truth PMF of our CG coordinate $s_\perp = x_1 - x_2$ can be simply obtained using numerical integration of Equation 7. Using this ground-truth PMF, the forward KLD values reported in Section 4.1 have been calculated in the range $s \in [-2.5, 1.1]$ using a grid of 100 points.

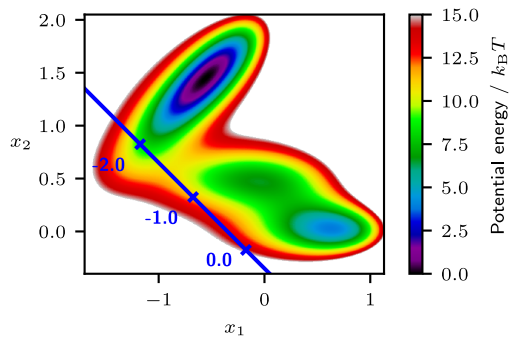


Figure 5. Müller-Brown potential. The blue axis represents the 1D CG coordinate s. The fine-grained coordinate $s_\perp$ is orthogonal to the CG coordinate.

A.5.2. Starting Dataset

To form the initial starting dataset for the Müller-Brown system we use a short MC simulation starting in the lower minimum of the potential ($x_1 = -0.25$, $x_2 = 1.5$). We perform 500 MC steps with a Gaussian proposal distribution of scale 0.2. We use 100 randomly chosen unique positions from this MC dataset as the starting dataset of the AL workflow.

A.5.3. Architecture

Normalizing Flow. As described in the main text, we use the following 1D transformation with the latent distribution $z \sim \mathcal{N}(0, 1)$ for the Müller-Brown system:

$$s_\perp = z \cdot \text{NN}_\text{scale}(s) + \text{NN}_\text{mean}(s) \tag{40}$$

The subnets NN_scale and NN_mean are fully connected neural networks with layer dimensions $[1, 64, 64, 1]$ and sigmoid hidden layer activation functions. We use a standard scaler $s' = \frac{s - \bar{s}}{\sigma_s}$ based on the starting dataset of the AL workflow for the input of the two neural networks.

PMF. To predict the PMF and its standard deviation, we use an ensemble of 10 fully connected neural networks with layer dimensions $[1, 64, 64, 1]$ and sigmoid hidden layer activation functions. We use a standard scaler $s' = \frac{s-\bar{s}}{\sigma_s}$ calculated from the current dataset of the respective AL iteration.

To improve ensemble diversity, we initialize the weights and biases of each model in the ensemble using a Gaussian of mean 0 and standard deviation ν. ν is sampled uniformly in the range $[0.1, 3.0]$ for each model in the ensemble.

A.5.4. Hyperparameters

Normalizing Flow. We use the Adam optimizer (Kingma & Ba, 2017) to train the normalizing flow.

When training by example on the starting dataset, we use a batch size of 16 and a learning rate of 5×10^{-4}.

When training the flow by energy, we use a batch size of 8 and a learning rate of 5×10^{-3}. We further clip gradients above a gradient norm of 20. The first AL iteration trains by energy for 12 epochs, all subsequent iterations use 7 epochs.

PMF. To train the PMF ensemble, we use the Adam optimizer with a learning rate of 0.001 and a batch size of 5. Training is performed for 1000 epochs.

We use the bagging strategy to select the training data of each model in the ensemble.

Monte Carlo Sampling. We perform Monte Carlo Sampling in the CG space to find new high-error configurations. We use a Gaussian proposal distribution of scale 0.1 and an error threshold of $0.4\, k_\mathrm{B}T$.

In each AL iteration, we search for 1 high-error point using 50 MC trajectories in parallel. The trajectories have a minimum length of 10 steps, before which we do not accept high-error configurations. The trajectories of the first AL iteration start in the global minimum of the potential. All trajectories of subsequent iterations start at the high-error configuration of the previous iteration. Each high-error point sampled using MC is subsequently broadened with a uniform distribution of width 1.0.

In this way, we sample 65 points for each high-error configuration, resulting in $1 \cdot 65 = 65$ added CG configurations in each AL iteration. Since each CG configuration has 30 copies in the dataset (see Section 3.3), this yields 1950 new points. 80 % are subsequently used for training by energy in the next iteration, and 20 % for calculation of a test loss.

If any of the trajectories reaches a length of 30 000 steps, we stop the active learning workflow. The reported total number of MC steps do not include the steps of this final fixed-length MC exploration that terminates the workflow due to reaching the maximum specified length.

A.6. Alanine Dipeptide

A.6.1. Force-Field and Ground Truth Simulations

We used the force-field AMBER ff96 with OBC GB/SA for implicit solvation (D.A. Case et al., 2023) for the ground truth simulations and potential energy evaluations during the training of the normalizing flow. Energy evaluations and simulations were performed using OpenMM 8.0.0 with the reference platform (Eastman et al., 2017). All simulations were performed at the temperature 300 K and included an iterative energy minimization (gradient descent) before starting the simulations from the obtained minimum energy configuration.

As a ground-truth test dataset, we used the dataset provided by Stimper et al. (2022) and Midgley et al. (2023b), which was generated using replica exchange MD simulations with a total of 2.3×10^{10} potential energy and force evaluations.

Additionally to this test dataset, we created an additional MD trajectory of length 5 µs with time step 1 fs (5×10^9 potential energy and force evaluations) to create the MD entries in Table 1 and the visualization in Figure 4.

A.6.2. Evaluation Metrics

Forward KLD. The ground truth test dataset mentioned in the previous section was used to calculate the forward KLD of the PMF on a 100x100 grid (see KLD values in Table 1).

All-Atom Log-Likelihood. The all-atom log-likelihoods in Table 1 have been calculated using the ground truth test dataset as

$$\mathbb{E}_{p_X(x)} \log q_X(x;\theta) = \mathbb{E}_{p_X(x)} \log \left(q_{X_{\mathrm{FG}}}\left(x_{\mathrm{FG}} \mid s;\theta\right) \cdot p(s) \cdot \left|\det J_{x\to x_{\mathrm{int}}}\right| \right) . \tag{41}$$

Here, $p(s) \sim \exp\left(-\frac{1}{k_{\mathrm{B}}T} U_{\mathrm{PMF}}(s)\right)$ was obtained from the PMF and normalized on a 500x500 grid in the CG space.

ESS. The reverse effective sample size (ESS) in Table 1 has been calculated in the following way (Midgley et al., 2023a; Martino et al., 2017):

$$n_{\mathrm{e,rv}} = \frac{1}{\sum_{i=1}^{n} \bar{w}\left(x_i\right)^2} \quad \text{with} \quad x_i \sim q_X\left(x_i;\theta\right) , \tag{42}$$

$$\bar{w}(x_i) = \frac{w(x_i)}{\sum_{i=1}^{n} w(x_i)} \text{ and } w(x_i) = \frac{p_X(x_i)}{q_X(x_i;\theta)} \tag{43}$$

Analogous to the all-atom log-likelihood calculation, we used $q_X(x;\theta) = q_{X_{\mathrm{FG}}}\left(x_{\mathrm{FG}} \mid s;\theta\right) \cdot p(s) \cdot \left|\det J_{x\to x_{\mathrm{int}}}\right|$, where $p(s) \sim \exp\left(-\frac{1}{k_{\mathrm{B}}T} U_{\mathrm{PMF}}(s)\right)$ was again obtained from the PMF and normalized on a 500x500 grid in the CG space. To sample from $q_X(x;\theta)$, we employed rejection sampling in the CG space and subsequently used the conditional normalizing flow to obtain all-atom samples. We used 1×10^7 samples drawn from $q_X(x;\theta)$ to estimate the reverse ESS. Analogous to Midgley et al. (2023b), due to outliers in the importance weights, we clipped the 1×10^3 highest importance weights to the lowest value among them.

A.6.3. Coordinate Transformation

We use a fully internal coordinate representation with $3 \cdot 22 - 6 = 60$ internal coordinates (bond distances, angles, and dihedral angles). The internal coordinates are shifted using the minimum energy reference structure obtained using gradient descent. Furthermore, we normalize the scale of the internal coordinates with the fixed parameters 0.005 nm for bond lengths, 0.15 rad for bond angles, and 0.2 rad for dihedral angles. The circular periodic coordinates (those with freely rotating dihedral angles) were not scaled. This internal coordinate representation is identical to the representation used by Midgley et al. (2023b).

A.6.4. Starting Dataset

To form the starting dataset we use a short MD trajectory of 50 ps (50 000 potential energy evaluations). Every 10th frame of this trajectory is used in the starting dataset.

A.6.5. Architecture

Normalizing Flow. Our architecture is very similar to the one by Midgley et al. (2023b), which builds upon the *normflows* python package (Stimper et al., 2023). We use a normalizing flow built from 12 monotonic rational-quadratic spline coupling layers (Durkan et al., 2019). A monotonic rational-quadratic spline coupling layer maps the interval $[-B, B]$ to $[-B, B]$ using monotonically increasing rational-quadratic functions in K bins.

We use $K = 8$ bins and treat the freely rotating dihedral angles with periodic boundaries in the range $[-\pi, \pi]$ (Rezende et al., 2020). For these, a uniform latent distribution in $[-\pi, \pi]$ is used. The non-periodic coordinates use $B = 5$, an identity mapping outside of the range $[-B, B]$, and a standard Gaussian latent distribution.

We group two consecutive coupling layers to use opposite random masks to define the identity and transform features. After such a pair of coupling layers, we apply a random shift of $\pi \cdot U(0.5, 1.5)$ with subsequent wrapping in $[-\pi, \pi]$ to the coordinates with periodic boundaries.

Each coupling layer uses a residual network as the parameter network of the spline transformation. The detailed architecture can be found in Figure 8. The circular coordinates (which include the CG variables ϕ and ψ) use the periodic representation $(\cos(\eta), \sin(\eta))^\top$ for each periodic variable η.

We found that training of the grid conditioning experiments is more stable and yields better results when not modeling the topology of ϕ and ψ explicitly in the flow, which is why we removed the periodic conditioning representation and simply condition the flow on ϕ and ψ for the grid conditioning experiments.

PMF. To predict the PMF and its standard deviation, we use an ensemble of 10 fully connected neural networks with layer dimensions $[4, 256, 128, 32, 1]$ and sigmoid hidden layer activation functions. Since ϕ and ψ are 2π-periodic, we use the input representation $(\cos\phi, \sin\phi, \cos\psi, \sin\psi)^\top$.

A.6.6. Hyperparameters

Normalizing Flow.

- Training by example: The initial training by example is performed using the Adam optimizer with a learning rate of 1×10^{-4} and a batch size of 256 for 50 epochs.
- Training by energy:
  - As discussed in Section 3, we found that removing a few of the highest loss values from each batch significantly improves training stability. Therefore, we removed the highest 5 loss values.
  - Active learning experiments
    - Adam optimizer with batch size 64
    - Training for 50 epochs in each iteration
    - Learning rate is linearly warmed up from 0 to 1×10^{-5} in the first 15 epochs of the initial iteration (directly after training by example) and in the first 11 epochs for all subsequent AL iterations.
    - Gradient norm clipping with value 100 for the first 15 epochs of the initial iteration (directly after training by example) and in the first 11 epochs for all subsequent iterations. Then, gradient norm clipping with value 1000 is used.
  - Grid conditioning experiments
    - 80% of the CG grid points (100x100) are used for training, 20% for testing.
    - Adam optimizer with batch size 64
    - Training for 100 epochs in total
    - Learning rate is linearly warmed up from 0 to 1×10^{-4} in the first 30 epochs .
    - Gradient norm clipping with a value of 100 for the first 30 epochs, then with a value of 1000

PMF. To train the PMF ensemble, we use the Adam optimizer with a learning rate of 5×10^{-4} and batch size 256. Training is performed for 1500 epochs. Due to possible atom clashes and numerical instabilities (Koblents & Míguez, 2015; Dibak et al., 2022) when sampling from the flow, when evaluating Equation 12 for each CG configuration s, we clip the highest 3 of the 30 values in the expectation value to the lowest value among them. We find empirically that this approach yields very accurate PMF values.

Each model in the ensemble receives a random fraction of 80 % of the current training data in the AL dataset for training and the remaining 20 % for testing.

Monte Carlo Sampling. We perform Monte Carlo Sampling in the CG space to find new high-error configurations. We use a Gaussian proposal distribution of scale 0.1 and an error threshold of $0.2\, k_\mathrm{B}T$.

In each AL iteration, we search for 30 high-error points using 200 MC trajectories in parallel. We use the high-error points found in the previous AL iteration as the starting configurations of these trajectories (the trajectories of the first AL iteration start in the global minimum). To obtain a more uniform coverage of the CG space, we first pick one random sample from these 30 configurations and then iteratively choose the point with the largest Euclidean distance to all already chosen points - until we have sampled a total of 15 points.

Each of these points is subsequently broadened with a uniform distribution in a circle with a radius of 0.6 rad. In this way, we sample 200 points for each high-error configuration, yielding $200 \cdot 15 = 3000$ added points in each iteration. Since each CG configuration has 30 copies in the dataset (see Section 3.3), this yields 90 000 new points. 80 % are subsequently used for training by energy in the next iteration, 20 % for calculation of a test loss.

If any of the trajectories reaches a length of 500 000 steps, we stop the active learning workflow. The reported total number of MC steps do not include the steps of this final fixed-length MC exploration that terminates the workflow due to reaching the maximum specified length.

Furthermore, if the forward KLD from the test dataset to the learned PMF in a given iteration is smaller than 1×10^{-3}, we stop the active learning workflow. When applying the presented methodology to systems without a ground-truth dataset in the future, one will have to solely rely on the maximum length or other more sophisticated stopping criteria.

A.6.7. Training Stability

As discussed in Section 3, we found that for alanine dipeptide removing the 5 highest loss values from each batch of 64 samples yields more stable experiments than energy regularization. While this yields stable AL workflows most of the time, we observe small artifacts in approximately one out of eight AL experiments (see Figure 9). These artifacts are in the form of vertical lines, mostly but not always at $\phi \approx 40°$. We found that optimizing training hyperparameters to get more stable experiments helps the problem, but does not completely remove it when repeating experiments many times. Therefore, we performed a total of 8 AL experiments with different random seeds in the alanine dipeptide system to obtain reliable performance metrics in Table 1.

For the grid conditioning experiments, we observed similar artifacts for some choices of hyperparameters, but the artifacts do not occur with the presented hyperparameters.

Since the number of sampled molecules of R-chirality (which are filtered) spikes at the same time as such an artifact typically occurs, we suspect that the artifact appears due to the hard cutoff of the chirality filtering. This is an effect of the chosen CG mapping, which has ambiguous chirality for a given ϕ-ψ-configuration. If one constructs the internal coordinate representation in such a way that only structures of L-chirality can be sampled (in combination with constraining the sampling range of the respective degree of freedom in the flow (Rezende et al., 2020)), the described effects should be avoidable in the future.

A.6.8. Performance Analysis

Here, we compare the performance of the MC simulations in our AL experiments with running all-atom MD using OpenMM. For this, we compare the time over the number of evaluations in Figure 10. As one can see, it takes more than 500 CPU cores to match the sampling speed of the PMF on the GPU - considering one performs enough trajectories in parallel.

As already discussed in the main text, running AL is not strictly necessary in this low-dimensional 2D CG space, since one can simply cover it uniformly as done in the grid conditioning experiments. Furthermore, the implicit solvent simulation of alanine dipeptide in OpenMM is already quite fast, resulting in a similar execution time when evaluating the PMF only once. While the nature of parallelization of the GPU still allows faster sampling of the PMF compared to MD in practice, one can expect an even larger difference when going to all-atom systems that take longer to evaluate.

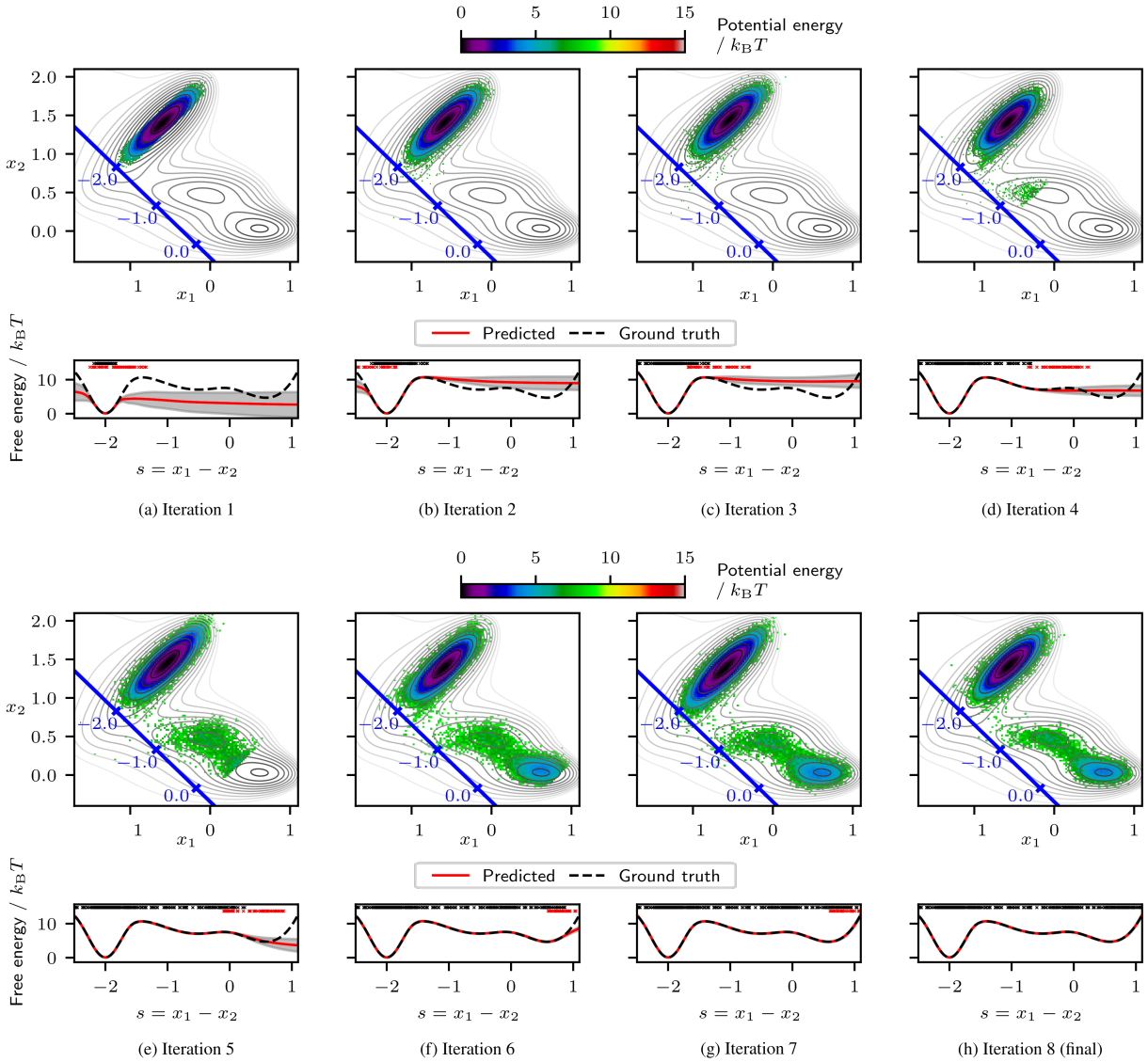
In the extreme case of using a more expensive method such as density functional theory (DFT) calculations on the all-atom side, sampling the CG surrogate will be many orders of magnitude faster. One can, of course, also replace the DFT with an all-atom ML surrogate and apply active learning also on the all-atom side. However, also in this case, the CG PMF will be lower-dimensional, require less memory, and have shorter execution times. Furthermore, the CG PMF is typically smoother than the all-atom energy surface, making larger time steps possible (Jin et al., 2022) when running, for example, Langevin dynamics.

A.7. Hardware Resources

All experiments have been performed on a NVIDIA A100 40 GB GPU. For the experiments on alanine dipeptide, we calculated the energy and gradients of the ground-truth OpenMM system using 18 workers in parallel.

Training times of the different experiments were approximately the following:

- Müller-Brown AL: 35 min
- Alanine dipeptide AL: 36 h
- Alanine dipeptide grid conditioning: 18 h



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 4

(e) Iteration 5 (f) Iteration 6 (g) Iteration 7 (h) Iteration 8 (final)

Figure 6. Visualization of all iterations of the AL workflow applied to the Müller-Brown system. Bottom: PMF and its standard deviation. Training points from previous AL iterations are marked as black "x" at the top of the PMF, and new high-error points added in the current AL iteration are marked as red "x". Top: Backmapped potential energy $\ln q_{S_\perp}(s_\perp \mid s; \theta) p^{\mathrm{CG}}(s)$ from cascaded sampling of the PMF and sampling of the flow.

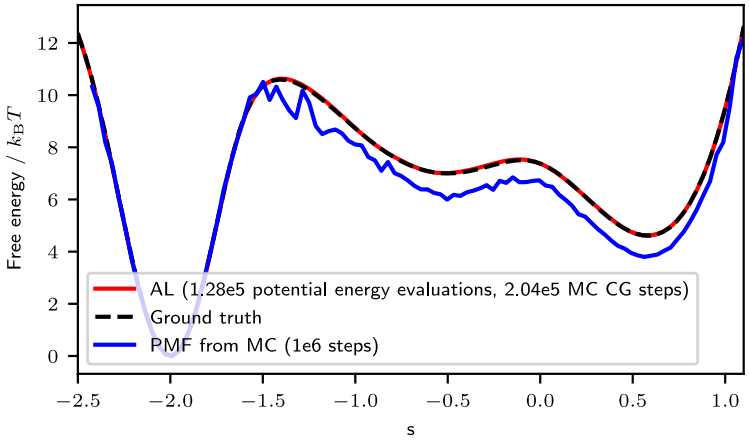


Figure 7. PMF as obtained using the AL experiment from Figure 6 (red), by numerical integration (ground truth, black), and from an "all-atom" MC simulation with 1×10^6 steps (blue). The PMF from the "all-atom" MC simulation is visibly biased, while the AL workflow yields an almost perfect PMF estimate.

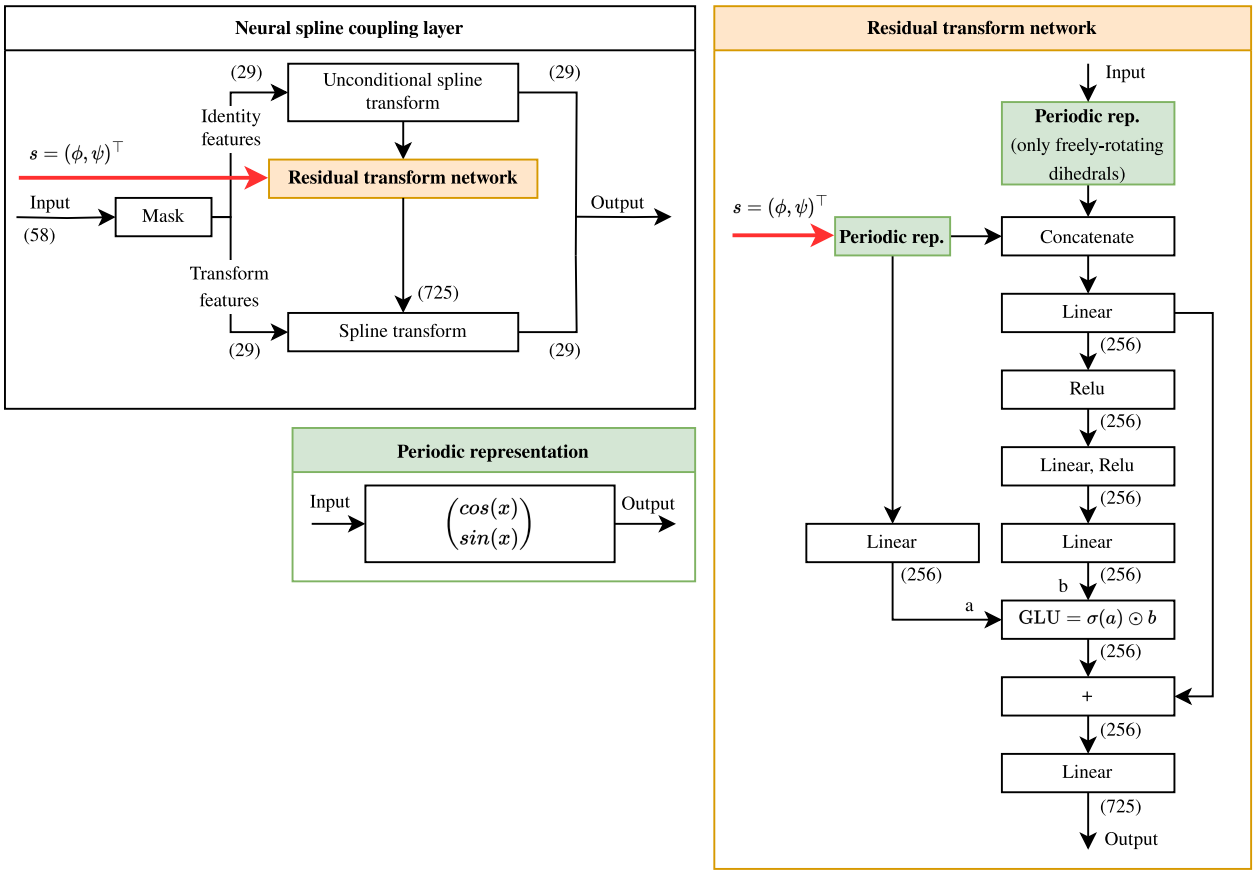


Figure 8. The architecture of the neural spline coupling layers used for alanine dipeptide. The input of each coupling layer is split into two parts, where the identity part is used to calculate the spline transform parameters of the transform part using a residual network (orange). The shown architecture is very similar to the one in (Midgley et al., 2023b).

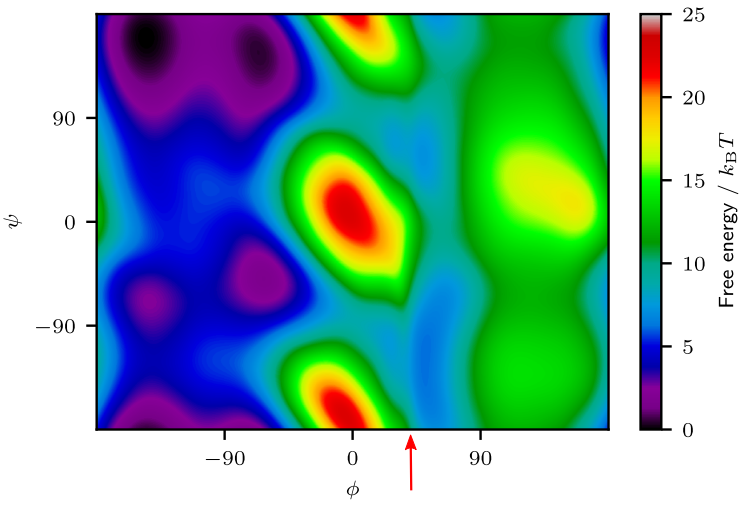


Figure 9. Small artifact (vertical line at $\phi \approx 40°$) that appears in the PMF during the active learning workflow applied to alanine dipeptide.

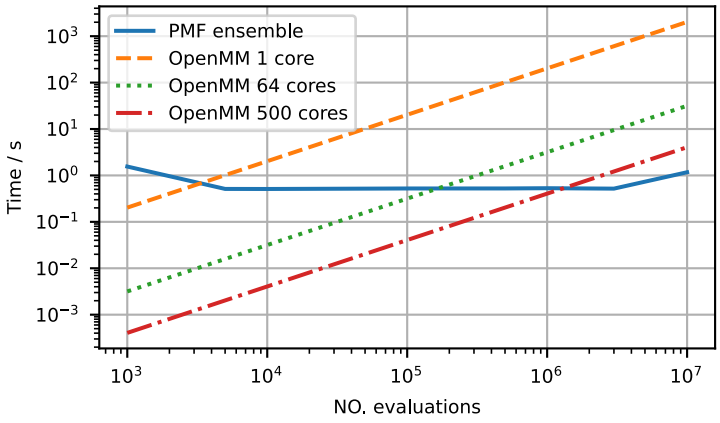
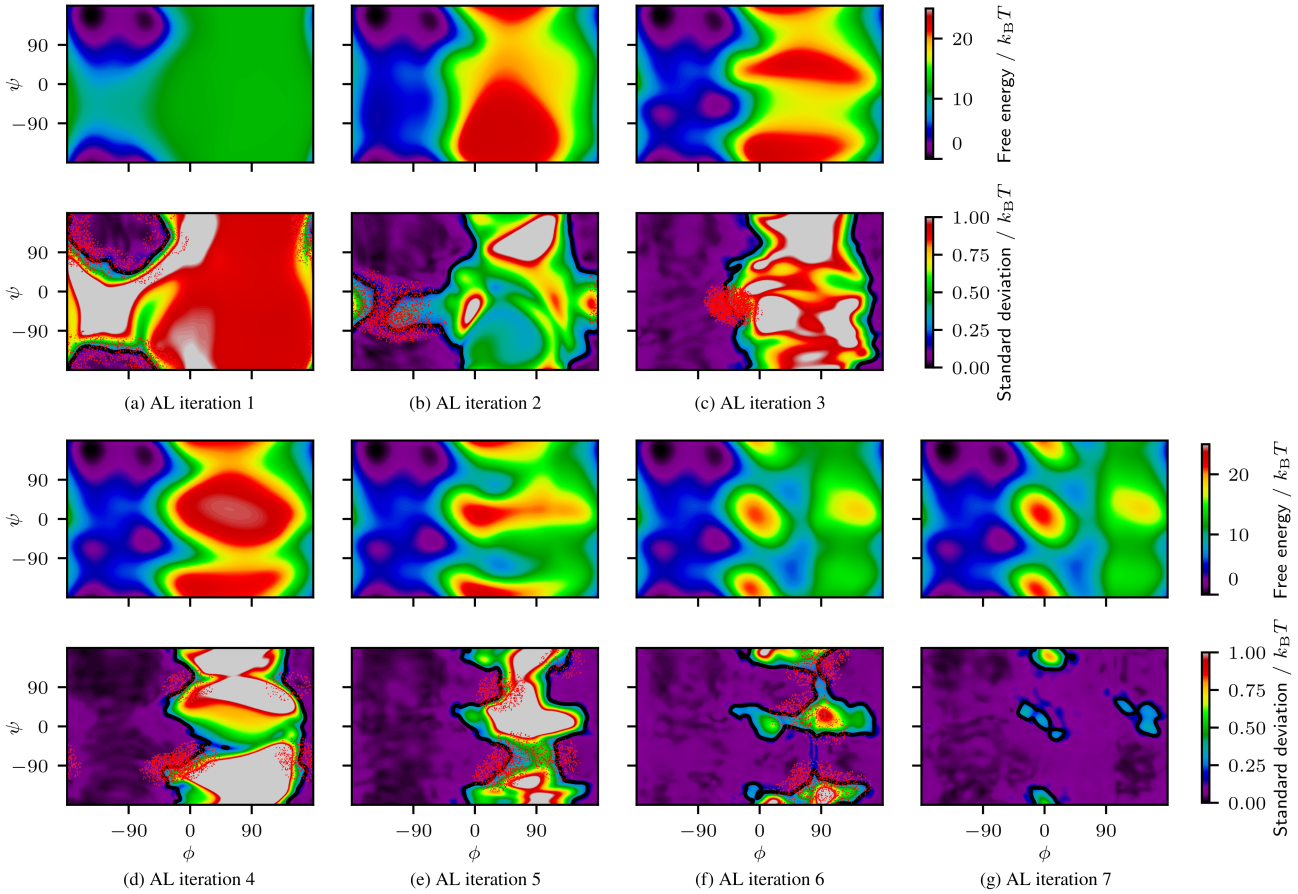


Figure 10. Benchmark that compares the time to evaluate the PMF ensemble on the GPU and the target potential energy and force using OpenMM on CPU cores. The OpenMM time was evaluated using the performance-optimized OpenMM CPU platform on a single core of an Intel Xeon Platinum 8368. Based on this time, we calculated the time for higher core numbers, assuming perfect scaling. The PMF ensemble was evaluated on a single NVIDIA A100 40 GB GPU.



(a) AL iteration 1 (b) AL iteration 2 (c) AL iteration 3

(d) AL iteration 4 (e) AL iteration 5 (f) AL iteration 6 (g) AL iteration 7

Figure 11. (a-g) AL steps for the alanine dipeptide system. Top: PMF of ϕ and ψ at the end of the iteration. Bottom: Standard deviation of the PMF. The contour line of the threshold in the standard deviation when sampling new points ($0.2\,k_\mathrm{B}T$) is drawn using a black line. Newly added points after sampling using the PMF of this iteration are shown as red dots.