# Consistent Sampling and Simulation: Molecular Dynamics with Energy-Based Diffusion Models

**Michael Plainer** [1 2 3 4] **Hao Wu** [5] **Leon Klein** [1] **Stephan Günnemann** [6] **Frank Noé** [1 7 8]

## Abstract

Diffusion models have recently gained significant attention due to their effectiveness in various scientific domains, including biochemistry. When trained on equilibrium molecular distributions, diffusion models provide both: a generative procedure to sample equilibrium conformations and associated forces derived from the model's scores. However, using the forces for coarse-grained molecular dynamics simulations uncovers inconsistencies in the samples generated via classical diffusion inference and simulation, despite both originating from the same model. Particularly at the small diffusion timesteps required for simulations, diffusion models fail to satisfy the Fokker-Planck equation, which governs how the score should evolve over time. We interpret this deviation as an indication of the observed inconsistencies and propose an energy-based diffusion model with a Fokker-Planck-derived regularization term enforcing consistency. We demonstrate the effectiveness of our approach on toy systems, alanine dipeptide, and introduce a state-of-the-art transferable Boltzmann emulator for dipeptides that supports simulation and demonstrates enhanced consistency and efficient sampling.

## 1. Introduction

Methodological advancements and increasing computational resources have allowed molecular dynamics (MD) simulations to reach biologically relevant timescales (Lindorff-Larsen et al., 2011; Wolf et al., 2020). However, scaling MD to larger or slower-changing systems remains computationally challenging. Coarse-graining (CG) methods address this by reducing the system dimensionality, but this comes at the cost of physical resolution, making it impossible to model interactions using traditional force fields accurately. Learning-based approaches (Clementi, 2008; Noid, 2013; Husic et al., 2020; Charron et al., 2023), offer an alternative by approximating these interactions.

Diffusion models (Ho et al., 2020; Song et al., 2021) have recently demonstrated considerable success in various molecular tasks, including on proteins and larger systems, usually relying on some form of coarse-graining (Abramson et al., 2024; Watson et al., 2023; Corso et al., 2023; Plainer et al., 2023b; Lewis et al., 2024). The idea behind diffusion models is to learn a reverse stochastic process that removes noise. Starting from pure noise, the model iteratively denoises the samples until it resembles the data distribution, and the neural network models the so-called *score* $\nabla_x \log p$.

When the training data accurately reflects samples from the equilibrium distribution of molecules, the learned score can be used not only for classical independent diffusion sampling, but also for MD simulations (Arts et al., 2023), providing access to thermodynamic and kinetic properties beyond static distributions. However, extracting the score from diffusion models (or the energy $\log p^{\boldsymbol{\theta}}$ for that matter) does not work well in practice, even for low-dimensional toy systems (Koehler et al., 2023; Li et al., 2023). While small local inaccuracies have little effect on independent sampling in diffusion models, using the extracted model for energy estimation reveals inconsistencies that can accumulate.

Analogously, while diffusion models should satisfy the Fokker-Planck equation (Särkkä & Solin, 2019), previous work shows that existing diffusion models violate this condition (Lai et al., 2023), especially when evaluated close to the data distribution. We hypothesize, and subsequently show empirically, that enforcing the Fokker-Planck equation significantly improves the consistency of the learned energy $\log p^{\boldsymbol{\theta}}$ and with it the alignment between independent samples and long-running simulations. We can see this behavior demonstrated on a toy example in Figure 1.

To implement this, we parameterize the score as the gradient of an energy function to ensure the learned score is

[*]Equal contribution [1]Freie Universität Berlin [2]Zuse School ELIZA [3]Technische Universität Berlin [4]Berlin Institute for the Foundations of Learning and Data [5]Shanghai Jiao Tong University [6]Technische Universität München [7]Rice University [8]Microsoft Research AI4Science. Correspondence to: Michael Plainer <michael.plainer@fu-berlin.de>, Hao Wu <hwu81@sjtu.edu.cn>.

*Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning*, Vancouver, Canada. PMLR 267, 2025. 

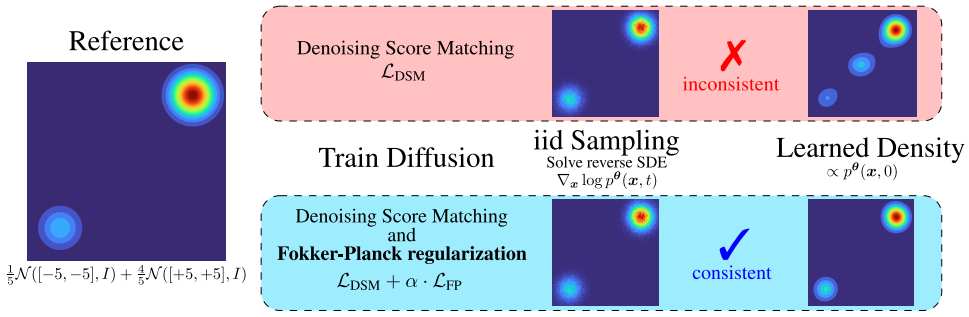


Figure 1: Training diffusion models on a simple mixture of two Gaussians reveals inconsistencies. While classical iid diffusion sampling recovers both modes, when estimating the unnormalized density at $t=0$, we observe that the model learns a third mode and an incorrect mass distribution. Using this model for simulation produces invalid results. Introducing our Fokker-Planck regularization makes the model more self-consistent.

conservative, and we have access to the energy $\log p^{\boldsymbol{\theta}}$. With this, we can introduce a Fokker-Planck-based regularization that minimizes deviations from theoretical consistency. However, directly evaluating the Fokker-Planck equation requires costly divergence computations, so we derive a computationally efficient "weak" residual formulation that requires only first-order derivatives. By further partitioning the diffusion timeline into distinct intervals handled by separate models, we can selectively apply the regularization only to the high error regions. This allows the model to learn to focus on the details and reduces training and inference costs. In Section 5, we validate our approach on a toy system, alanine dipeptide, and demonstrate its scalability by training a transferable Boltzmann emulator across dipeptides.

Our main contributions in this work are as follows:

1. We show how to regularize the energy of diffusion models using the Fokker-Planck equation, enabling consistent molecular dynamics simulations alongside traditional sampling.
2. We demonstrate that training on a small sub-interval of the diffusion process suffices for stable simulation. By combining this with smaller models trained on complementary intervals, we achieve efficient training and inference without sacrificing sampling performance.
3. We develop a state-of-the-art transferrable Boltzmann emulator for dipeptides capable of high-quality independent sampling and consistent simulation.

## 2. Background

### 2.1. Generative Score-based Modeling

**Diffusion models** (Song & Ermon, 2019; Ho et al., 2020; Song et al., 2021) are self-supervised generative models that gradually corrupt the training data with noise and learn to reverse this stochastic process. The forward process is typically defined by a stochastic differential equation (SDE)

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x},t)\,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{w}, \tag{1}$$

where $\boldsymbol{w}$ denotes the standard Wiener process, and $\boldsymbol{f}$ and $g$ define the drift and diffusion coefficient respectively. To generate samples, diffusion models simulate the corresponding reverse-time SDE

$$\mathrm{d}\boldsymbol{x} = \left[\boldsymbol{f}(\boldsymbol{x},t) - g^2(t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right]\mathrm{d}t + g(t)\,\mathrm{d}\bar{\boldsymbol{w}}, \tag{2}$$

starting from Gaussian noise at time $t=1$, they iteratively denoise until a sample is produced at $t=0$. Here, $p_t(\boldsymbol{x})$ denotes the density of $\boldsymbol{x}$ at time $t$, which the model aims to approximate. $\bar{\boldsymbol{w}}$ denotes the time-reversed Wiener process.

As for $\boldsymbol{f}$ and $g$, the choice depends on the specific diffusion formulation. In this work, we adopt the variance preserving (VP) SDE formulation introduced by (Song et al., 2021).

**Denoising score matching.** Diffusion models are typically trained using denoising score matching (Vincent, 2011; Song et al., 2021), which minimizes the squared error between a time-dependent learned score function $\boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x}(t),t)$ and the true score of the transition kernel $p_{0t}(\boldsymbol{x}(t)\,|\,\boldsymbol{x}(0))$ conditioned on the training data $\boldsymbol{x}(0)$

$$\mathbb{E}\left[\lambda(t)\left\|\boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x}(t),t) - \nabla_{\boldsymbol{x}(t)} \log p_{0t}(\boldsymbol{x}(t)\,|\,\boldsymbol{x}(0))\right\|_2^2\right], \tag{3}$$

where $\lambda(t)$ is a time-dependent weighting function. For brevity, we will denote the denoising diffusion loss $\mathcal{L}_{\text{DSM}}[\boldsymbol{s}^{\boldsymbol{\theta}}](\boldsymbol{x},t)$ as

$$\lambda(t)\left\|\boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x}(t),t) - \nabla_{\boldsymbol{x}(t)} \log p_{0t}(\boldsymbol{x}(t)\,|\,\boldsymbol{x}(0))\right\|_2^2. \tag{4}$$

**Parameterization and instabilities.** With an affine drift $\boldsymbol{f}$, we can write the closed-form solution of $p_{0t}$ as a Gaussian (Särkkä & Solin, 2019), and can efficiently evaluate the loss with

$$\mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\lambda(t)\left\|\boldsymbol{s}^{\boldsymbol{\theta}}(\mu(\boldsymbol{x}(0),t)+\sigma(t)\epsilon,t)+\frac{\epsilon}{\sigma(t)}\right\|_2^2\right], \tag{5}$$

where $\mu(\boldsymbol{x}(0),t), \sigma(t)$ depend on the concrete choices for $\boldsymbol{f}$ and $g$. By construction, $\sigma(0)=0$, ensuring interpolation between data and noise. Minimizing the denoising

loss yields an approximation of the unconditional score $\boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x}, t) \approx \nabla_x \log p_t(\boldsymbol{x})$. Details on the concrete choices for this formulation can be found in Appendix A.1.

As $t \to 0$, this parameterization introduces numerical instability, where $\sigma(t)$ vanishes and the loss explodes. This instability makes training difficult at small timescales (Kim et al., 2022) and is typically mitigated by truncating the training interval to $(\varepsilon, 1)$ for some $\varepsilon > 0$. While effective for training stability, this inherent instability in training limits the model's accuracy at small $t$, which is critical for applications requiring reliable scores close to the data manifold, as targeted in this work.

### 2.2. Boltzmann Distribution

**Langevin simulation.** Samples from the *Boltzmann distribution* of molecular systems are typically generated using MD simulations. A common approach is to simulate Langevin dynamics (Leimkuhler & Matthews, 2015), which corresponds to integrating the following set of SDEs

$$\begin{aligned} \mathrm{d}\boldsymbol{x} &= \boldsymbol{v}\,\mathrm{d}t\,, \\ \boldsymbol{M}\,\mathrm{d}\boldsymbol{v} &= -\nabla_{\boldsymbol{x}} U(\boldsymbol{x})\,\mathrm{d}t - \gamma \boldsymbol{M}\boldsymbol{v}\,\mathrm{d}t + \sqrt{2\gamma k_B T}\,\mathrm{d}\boldsymbol{w}_t. \end{aligned} \tag{6}$$

$\boldsymbol{M}$ denotes the particle masses, $\boldsymbol{v}$ the velocities, $\gamma$ is a friction constant, $k_B T$ a constant, and $\boldsymbol{w}_t$ is the standard Brownian motion. Note that $t$ here refers to the physical time instead of the diffusion time used earlier. Integration of this system requires access to the forces $-\nabla_{\boldsymbol{x}} U$. However, in settings where direct force evaluation is not feasible, such as in CG models, a surrogate force function is required. In this work, we propose using the score $\boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x}, t=0)$ for this purpose, as we describe next.

**Extracting forces.** After performing a long-running MD simulation, the samples are distributed according to the Boltzmann distribution (Boltzmann, 1868) such that $p(\boldsymbol{x}) = \exp(-\frac{U(\boldsymbol{x})}{k_B T})/Z$, with an underlying potential $U$ and a normalization constant $Z$. Training a diffusion model on this data establishes the following relation at $t = 0$

$$\begin{aligned} \boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x}, t=0) &\approx \nabla_x \log p_{t=0}(\boldsymbol{x}) \\ &= \nabla_x \log \exp\left(-\frac{U(\boldsymbol{x})}{k_B T}\right) - \nabla_{\boldsymbol{x}} \log Z \\ &= -\nabla_x \frac{U(\boldsymbol{x})}{k_B T} - 0. \end{aligned} \tag{7}$$

This reveals that the score is proportional to $-\nabla_{\boldsymbol{x}} U(\boldsymbol{x})$, the forces acting on the system. Intuitively, this means that as the diffusion process gets closer to the data, the sampling becomes more "physical". Importantly, this equivalence shows that any diffusion model trained on Boltzmann-distributed data can be used not only for independent sampling but also for simulating molecular dynamics by leveraging the learned score as a force estimator and Equation (6).

Unlike prior works that require explicit force labels for training (Husic et al., 2020; Durumeric et al., 2023; Charron et al., 2023; Krämer et al., 2023), this observation allows us to learn a model directly from equilibrium samples. This is particularly useful when force labels are unavailable.

## 3. Method

We introduce a *Fokker-Planck*-based regularization that improves the consistency between iid samples and the learned energy in diffusion models, to enable more accurate molecular dynamics. To this end, we leverage a conservative neural network parameterization to evaluate the time derivative of the model's unnormalized density to enforce consistency with the Fokker-Planck equation in training.

### 3.1. Improving Consistency with Fokker-Planck

The Fokker-Planck equation (Øksendal, 2003; Särkkä & Solin, 2019) is a partial differential equation that describes the time evolution of probability densities in stochastic processes, including diffusion models. For the diffusion SDE introduced in Equation (1), the log-density formulation of the Fokker-Planck equation (Lai et al., 2023; Hu et al., 2024) can be written as

$$\begin{aligned} \partial_t \log p_t(\boldsymbol{x}) &= \mathcal{F}_p(\boldsymbol{x}, t) \\ &\triangleq \frac{1}{2} g^2(t) \left[\mathrm{div}_{\boldsymbol{x}}(\nabla_{\boldsymbol{x}} \log p_t) + \|\nabla_{\boldsymbol{x}} \log p_t\|_2^2\right] \\ &\quad - \langle \boldsymbol{f}, \nabla_{\boldsymbol{x}} \log p_t \rangle - \mathrm{div}_{\boldsymbol{x}}(\boldsymbol{f}), \end{aligned} \tag{8}$$

where $\mathrm{div}_{\boldsymbol{x}}$ is the divergence operator $\mathrm{div}_{\boldsymbol{x}}\, \boldsymbol{F} = \mathrm{tr}\,(\partial_{\boldsymbol{x}} \boldsymbol{F})$.

**Fokker-Planck regularization.** The energy $\log p_t^{\boldsymbol{\theta}}$ of a well-trained diffusion model should satisfy Equation (8). However, we will show empirically in Section 5 that diffusion models do not fulfill the Fokker-Planck equation, particularly for small $t$. This aligns with previous findings on the self-consistency of diffusion models (Koehler et al., 2023; Lai et al., 2023; Li et al., 2023), and empirical results where MD simulations with the score do not match the data distribution (Arts et al., 2023). Building upon this prior work, we show that by minimizing this Fokker-Planck deviation, the model's score $\nabla_x \log p_0^{\boldsymbol{\theta}}$ more accurately describes the forces and aligns better with iid samples.

The natural approach to ensure that the model is consistent with Equation (8) is to introduce a regularization term to the diffusion loss. For this, we propose to minimize the error

$$\|R(\boldsymbol{x}, t)\|_2^2 = \left\|\mathcal{F}_{p^{\boldsymbol{\theta}}}(\boldsymbol{x}, t) - \partial_t \log p_t^{\boldsymbol{\theta}}(\boldsymbol{x})\right\|_2^2, \tag{9}$$

and define the corresponding loss as $\mathcal{L}_{\mathrm{FP}}[\log p^{\boldsymbol{\theta}}](\boldsymbol{x}, t) = \lambda_{FP}(t) D^{-2} \|R(\boldsymbol{x}, t)\|_2^2$, where $\boldsymbol{x} \in \mathbb{R}^D$ and $\lambda_{FP}$ is a time-dependent weighting function, which we set to be the same

as $\lambda$. This formulation is only feasible when using a conservative energy-based model, as we need to evaluate $\partial_t \log p_t^{\boldsymbol{\theta}}$. With this, the full training objective becomes

$$\mathbb{E}\left[\mathcal{L}_{\text{DSM}}[\nabla_{\boldsymbol{x}} \log p^{\boldsymbol{\theta}}](\boldsymbol{x}(t), t) + \alpha \cdot \mathcal{L}_{\text{FP}}[\log p^{\boldsymbol{\theta}}](\boldsymbol{x}(t), t)\right], \tag{10}$$

where $\alpha$ is a hyperparameter that determines the regularization strength. As we will show in Section 5, minimizing this regularized loss improves consistency between iid sampling and Langevin simulation of diffusion models, allowing for more accurate MD simulations.

**Weak residual formulation.** The exact computation of the residual $R$ involves costly higher-order derivatives, especially the divergence term $\text{div}_{\boldsymbol{x}}(\nabla_{\boldsymbol{x}} \log p^{\boldsymbol{\theta}})$ can be challenging to compute for high-dimensional data. To reduce this overhead, we introduce a series of approximations and adopt a residual in the weak formulation (Guo et al., 2022) such that

$$\widetilde{R}(\boldsymbol{x}, t) = E_{\boldsymbol{v}}\left[R(\boldsymbol{x} + \boldsymbol{v}, t)\right] \tag{11}$$

with $\boldsymbol{v} \sim \mathcal{N}(0, \sigma^2 I)$ and a small $\sigma > 0$. As $\sigma$ approaches 0, $\widetilde{R}(\boldsymbol{x}, t)$ will be equal to $R(\boldsymbol{x}, t)$.

**Theorem 3.1.** *Using the weak residual formulation, $\widetilde{R}(\boldsymbol{x}, t)$ can be estimated by the following unbiased estimator, which only requires the computation of first-order derivatives*

$$\begin{aligned}\widetilde{R}(\boldsymbol{x}, t; \boldsymbol{v}) = &\frac{1}{2} g^2(t) \left(\frac{\boldsymbol{v}}{\sigma}\right)^\top \frac{\boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{v}, t) - \boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x} - \boldsymbol{v}, t)}{2\sigma} \\ &+ \frac{1}{2} g^2(t) \left\|\boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{v}, t)\right\|_2^2 \\ &- \langle \boldsymbol{f}(\boldsymbol{x} + \boldsymbol{v}, t), \boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{v}, t)\rangle \\ &- \text{div}_{\boldsymbol{x}}(\boldsymbol{f}(\boldsymbol{x} + v, t)) \\ &- \partial_t \log p_t^{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{v}),\end{aligned} \tag{12}$$

*where $\boldsymbol{s}^{\boldsymbol{\theta}} = \nabla_{\boldsymbol{x}} \log p^{\boldsymbol{\theta}}$. An unbiased estimator of the squared residual needed for optimization is*

$$\left\|\widetilde{R}(\boldsymbol{x}, t)\right\|_2^2 \approx \widetilde{R}(\boldsymbol{x}, t; \boldsymbol{v}) \cdot \widetilde{R}(\boldsymbol{x}, t; \boldsymbol{v}'), \tag{13}$$

*with $\boldsymbol{v}, \boldsymbol{v}' \sim \mathcal{N}(0, \sigma^2 I)$.*

*Proof.* See Appendix A.2. □

We further reduce computational cost by estimating $\partial_t \log p_t^{\boldsymbol{\theta}}$ using finite differences (Fornberg, 1988):

$$\begin{aligned}\partial_t \log p_t^{\boldsymbol{\theta}} \approx &\frac{h_s^2 \log p_t^{\boldsymbol{\theta}}(\boldsymbol{x}, t + h_d) - h_d^2 \log p_t^{\boldsymbol{\theta}}(\boldsymbol{x}, t - h_s)}{h_s h_d (h_s + h_d)} \\ &+ \frac{(h_d^2 - h_s^2) \log p_t^{\boldsymbol{\theta}}(\boldsymbol{x}, t)}{h_s h_d (h_s + h_d)}.\end{aligned} \tag{14}$$

In combination with Theorem 3.1, this allows for efficient approximation of the loss $\mathcal{L}_{\text{FP}}$.

### 3.2. Physically Consistent Model Design

To model physical systems accurately and support Fokker-Planck regularization, we must enforce known physical constraints through an appropriate parameterization and choice of architecture.

**Conservative model parameterization.** Diffusion models commonly parameterize the score $\boldsymbol{s}^{\boldsymbol{\theta}} = NNET(\boldsymbol{x}, t)$ directly, whereas an energy-based parameterization of the score $\nabla_{\boldsymbol{x}} \log p_t^{\boldsymbol{\theta}} = \nabla_x NNET(\boldsymbol{x}, t)$ requires differentiation during the forward pass. While an energy-based formulation has been explored previously (Song & Ermon, 2019), nowadays it is less common in practice (Du et al., 2023), as most applications require only the score and there is no practical difference in sampling quality (Salimans & Ho, 2021). However, for MD simulations, the conservative property provided by the energy-based parameterization is crucial, since it stabilizes the simulation (Arts et al., 2023), as demonstrated in Appendix C.2. This means that access to $\nabla_{\boldsymbol{x}} \log p_t$ needed for computing $\mathcal{L}_{\text{FP}}$ introduces no additional overhead beyond what is already needed for MD.

**Architecture.** Our choice for the score is conservative, translation invariant, and learns $SO(3)$ equivariance. Similarly to (Arts et al., 2023), we use a graph transformer (Shi et al., 2021), making the score permutation equivariant and achieving translation invariance by using pairwise distances instead of absolute coordinates. For the rotation equivariance, recent high-profile work (Abramson et al., 2024) has shown that the architecture itself does not necessarily need to enforce this property. Hence, we opted to apply random rotations during training so that the network learns rotational equivariance via data augmentation.

The main part of the architecture can be summarized as

$$\begin{aligned}\boldsymbol{e}_{ij} &= \boldsymbol{x}_i - \boldsymbol{x}_j\,, \\ \boldsymbol{n}_i^{(0)} &= [\boldsymbol{a}_i, t]\,, \\ \boldsymbol{n}^{(l+1)} &= \phi^{(l)}(\boldsymbol{n}^{(l)}, \boldsymbol{e}),\end{aligned} \tag{15}$$

where $\boldsymbol{x}$ are the coarse-grained positions, $\boldsymbol{e}$ are the edge features, $\boldsymbol{a}$ are atom features, $t$ is the diffusion time, $\boldsymbol{n}^{(l)}$ are the node embeddings of layer $l$, and $\phi$ is one layer of the graph transformer. When training on a single molecule, we use one-hot atom types; for the transferable model, we use: atom identity, atom number, residue index, and amino acid type following (Klein & Noé, 2024).

Finally, to achieve conservativeness, we map the last node embeddings $\boldsymbol{n}_i^{(L)} \in \mathbb{R}^K$ to scalar energies via $\psi : \mathbb{R}^K \to \mathbb{R}$, and compute the score as $\nabla_{\boldsymbol{x}} \sum_i \psi(\boldsymbol{n}_i^{(L)})$. Overall, this yields a translation-invariant, approximately rotation-equivariant, conservative architecture that also avoids issues caused by mirror symmetries (Trippe et al., 2023; Klein & Noé, 2024).

### 3.3. Mixture of Experts

Using conservative models and Fokker-Planck regularization introduces computational overhead, especially during training. We will show in Section 5, that, particularly at large diffusion times $t$, this precision is unnecessary. To address this, we adopt a time-based mixture of experts (MoE) approach to allocate model capacity more efficiently and further improve model consistency.

Instead of training a single model for all $t \in (0, 1)$, we partition the interval into disjoint subintervals $\mathcal{I}_0, \mathcal{I}_1, \ldots$ with $\bigcup_i \mathcal{I}_i = (0, 1)$, and assign a separate expert $\boldsymbol{s}_i^{\boldsymbol{\theta}}$ to each interval. The overall score is

$$\boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x}, t) = \sum_i w_i(t) \boldsymbol{s}_i^{\boldsymbol{\theta}}(\boldsymbol{x}, t), \tag{16}$$

where $w(t) \in [0, 1]$ is a time-dependent gating function that selects the current active model. Similar ideas have been explored in the image domain to fine-tune models and improve sampling performance (Balaji et al., 2023; Ganjdanesh et al., 2025). In our setup, only one model is active at a given $t$, simplifying memory and compute requirements and allowing all models to be trained in parallel. Although training each model independently from the others induces discontinuities at the boundaries where two models switch, we observe no significant drawbacks in practice.

This scheme has two main advantages: First of all, our experiments reveal that introducing the loss from Equation (10) to the whole model can lead to overregularization at larger timescales, degrading iid sampling performance. This suggests that models for large $t$ do not require Fokker-Planck regularization (and no conservative parameterization) to be accurate. By using simpler unconstrained models for larger timescales, we can improve performance while preventing overregularization. Further, as each expert specializes on a subinterval of $t$, it can focus its capacity accordingly. Experts for small $t$ handle fine-grained structure, while those at large $t$ model coarse features (Ganjdanesh et al., 2025). As this makes the structures each model sees more similar, MoE can further stabilize and improve simulation results, even when keeping the overall number of parameters fixed.

## 4. Related Work

In recent years, a variety of deep learning methods have been proposed to enhance or replace molecular simulation. The work most closely related to ours is that of Arts et al. (2023), who employ an energy-based diffusion model for coarse-grained systems. However, their approach fails to maintain consistency between sampling and simulation. Moreover, they mitigate some of the inconsistencies we describe by evaluating the diffusion model at larger timesteps $t > 0$, which introduces additional noise and reduces structural fidelity. We compare against this model in Section 5 and show that evaluating at a different $t$ is not a suitable way to prevent this mismatch. Several works instead learn coarse-grained force fields via a force-matching objective (Husic et al., 2020; Köhler et al., 2023; Charron et al., 2023; Durumeric et al., 2024). Rather than training a model to represent the data distribution directly, these methods aim to approximate the target forces. However, these models typically require system-specific energy priors, which rely heavily on domain knowledge.

Other approaches bypass MD sampling entirely, generating Boltzmann distributed configurations either sequentially, by conditioning each sample on its predecessor (Dibak et al., 2022; Plainer et al., 2023a; Schreiner et al., 2023; Tamagnone et al., 2024), or completely independently (Noé et al., 2019; Wirnsberger et al., 2020; Köhler et al., 2020; Midgley et al., 2023; Klein et al., 2023b; Abdin & Kim, 2023; Kim et al., 2024; Wu & Noé, 2024; Schebek et al., 2024; Diez et al., 2024; Tan et al., 2025). These methods frequently leverage diffusion- or flow-based architectures. In the latter case, for all-atom systems with known energy functions, they can guarantee asymptotically unbiased sampling via reweighting or integration into an MCMC scheme. Although, this is often a desirable property, extending it to larger systems remains challenging, as CG is not possible.

The inaccurate behavior of the score function has also been studied in low-dimensional settings by (Koehler et al., 2023; Li et al., 2023), who demonstrated inconsistencies and derived error bounds. (Lai et al., 2023) also proposed a Fokker–Planck-inspired regularization; however, unlike our approach, their method applies a higher-order regularization to the score itself to improve iid sample quality, rather than enforcing consistency through the potential. Relatedly, Hu et al. (2024) propose a score-based solver for high-dimensional Fokker–Planck equations, focusing on general SDE forward problems, and (Du et al., 2024) use the Fokker-Planck equation to describe MD as a series of Gaussians.

## 5. Experiments

In this section, we compare models using two modes: classical diffusion sampling (*iid*) and Langevin simulation (*sim*). We consider a model to be consistent when the outputs of these two modes match. We demonstrate that our approach improves the consistency of diffusion models across several settings. We begin with a two-dimensional toy example—the Müller-Brown potential (Müller & Brown, 1979)—followed by alanine dipeptide, and conclude with a model that generalizes across dipeptides. The code is available at `https://github.com/noegroup/ScoreMD`.

**Metrics.** We mainly compare the 2D free energy surfaces across different methods. For molecules, we use the dihedral angles $\varphi, \psi$ to reduce the data to two dimensions. We

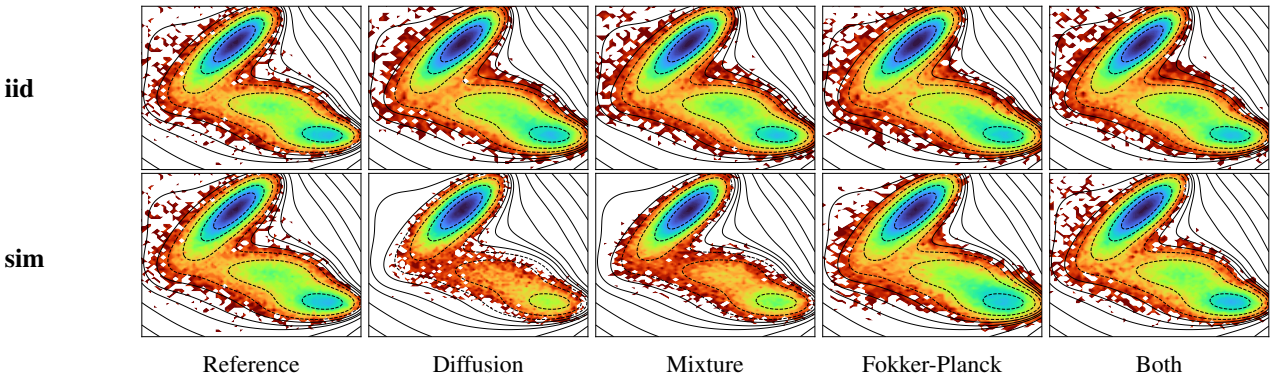


Figure 2: Comparing free energy plots of different models on the Müller-Brown potential for classical diffusion sampling (iid) and Langevin simulation (sim). The energies should align with the training data (Reference).

report an approximated Jensen-Shannon (JS) divergence to describe the similarity between the sampled and reference densities. We also report the potential of mean force (PMF) error, which computes the squared distance between the negative logarithms of the samples and the reference, giving more weight to low-density regions (Durumeric et al., 2024). More details can be found in Appendix B.1.

**Baselines.** We train a conservative *Diffusion* model and have re-implemented the *Two For One* method (Arts et al., 2023) using a continuous-time diffusion process to ensure comparability. Both methods use the same architecture, with the only difference being that *Two For One* evaluates at a non-zero diffusion time for simulation. For transferability, we re-train the *Transferable Boltzmann generator (BG)* model (Klein & Noé, 2024) with coarse-graining and compare it without reweighting.

We compare these baselines with three models introduced in this work: *Mixture* refers to the MoE scheme with three models trained on the intervals $(0, 0.1)$, $[0.1, 0.6)$, and $[0.6, 1.0)$. All models are combined for iid sampling, while only the smallest-timescale model is used for simulation. The models for larger timescales are reduced in size and complexity. *Fokker-Planck* refers to a model where we use the loss from Equation (10) to train a single model. *Both* combines these two approaches with the regularization only applied to the smallest-timescale model.

### 5.1. Müller-Brown Potential

We first evaluate on the Müller-Brown potential using 100k samples drawn from its Boltzmann distribution in Figure 2. All methods produce iid samples that match the *Reference*. However, when using the learned score for Langevin simulation, the standard *Diffusion* model fails to reproduce the correct distribution and undersamples the low-probability state, highlighting the inconsistency between sampling and simulation. Although having roughly the same number of parameters, the *Mixture* model partially improves this, but consistency is only achieved with *Fokker-Planck* regularization. Combining *Both* approaches further improves performance (also compare Appendix C.1).

### 5.2. Alanine Dipeptide

**Dataset.** We use 50k samples from an MD simulation of alanine dipeptide in implicit solvent (Köhler et al., 2021), coarse-grained to five atoms: [C, N, CA, C, N], as shown in Figure 3 (a). When evaluating the models, we perform $1.2\mu s$ of simulations with a $2fs$ timestep, starting from 100 different training conformations and downsample to match the training set size for consistency.

**Inconsistent sampling.** Figure 3 (b) compares the free energies of the sampled dihedral angles for iid sampling and Langevin simulation (sim). While all methods can match the training distribution under iid sampling, simulation quality varies, and existing models show inconsistencies. Standard *Diffusion* fails to recover the low-probability mode (i.e., $\varphi > 0$) completely, even when starting a simulation from these regions. *Mixture* generally improves the results, but still does not find the other mode. This is reflected in the numerical results in Table 1, where *Mixture* achieves lower means and smaller variance, but simulation errors remain noticeable. We attribute this behavior to the smaller time range, which focuses the model's attention, allowing it to learn a better, more stable optimum.

**Noisy simulation.** *Two For One* improves consistency by increasing the diffusion time $t > 0$ where the model is evaluated for simulation. However, this also introduces excessive noise, which degrades structural fidelity. Although iid sampling is not affected by this, the structures produced by Langevin simulation show significant deviations. For example, the W1 distance for the C–N bond is $48.14 \pm 13.03$ larger than for *Both*, as can be seen in Figure 4.

**Consistent models.** *Fokker-Planck* regularization enables the model to recover the missing states without modifying

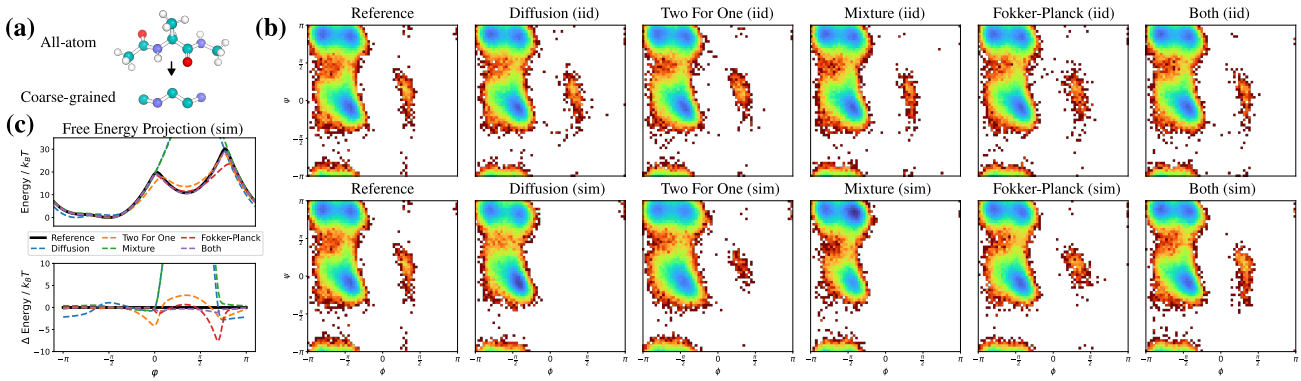


Figure 3: Comparison of methods on alanine dipeptide. **(a)** The coarse-graining scheme. **(b)** Comparison of the Ramachandran plots of different methods for iid sampling and Langevin simulation. **(c)** The projection of the free energy surface and differences along the dihedral angle $\varphi$ for samples generated with simulation.

| Method | iid JS (↓) | sim. JS (↓) | iid PMF (↓) | sim. PMF (↓) |
|---|---|---|---|---|
| Diffusion | **0.0081 ± 0.0003** | 0.0695 ± 0.0517 | 0.095 ± 0.003 | 1.047 ± 0.924 |
| Two For One | **0.0081 ± 0.0003** | 0.0158 ± 0.0002 | 0.098 ± 0.006 | 0.206 ± 0.004 |
| Mixture | **0.0080 ± 0.0004** | 0.0353 ± 0.0117 | **0.092 ± 0.007** | 0.388 ± 0.109 |
| Fokker-Planck | 0.0084 ± 0.0002 | **0.0088 ± 0.0006** | 0.098 ± 0.006 | **0.105 ± 0.011** |
| Both | **0.0079 ± 0.0002** | **0.0086 ± 0.0004** | **0.089 ± 0.005** | **0.099 ± 0.003** |

Table 1: Comparison of alanine dipeptide with JS divergence and PMF error. To compute the mean and the standard deviation, we have trained and evaluated three models with three different seeds. Lower values are better.

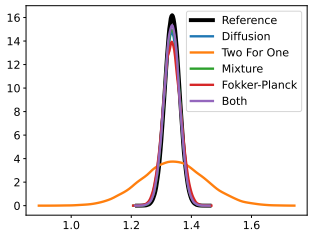


Figure 4: Histogram of the C–N bond length in Å for MD simulation (sim) for various models. We can see that only *Two For One* produces structures with significantly worse accuracy.

the diffusion time, and thus preserving structural accuracy. Table 1 shows that the regularization substantially improves consistency between iid and simulation, although iid performance slightly declines in favor of improved simulation performance. Combining MoE with Fokker-Planck regularization for *Both*, enhances simulation quality further while mitigating the drop in iid performance. The resulting model achieves close alignment between iid and simulation, and captures the free energy landscape in simulations accurately (see Figure 3 (c)). The superior iid performance of *Both* over *Fokker-Planck* suggests that applying regularization at larger diffusion times may introduce too many constraints on the model and hence degrade generative quality.

### 5.3. Transferability Across Dipeptides (two amino acids)

**Dataset.** We use the dataset introduced by (Klein et al., 2023a) that consists of 49k samples from implicit solvent simulation of $1\mu s$ for all 400 dipeptides (i.e., all possible combinations of the standard 20 amino acids). We have coarse-grained the dipeptides and kept the atoms [N, CA, CB, C, O] for each amino acid, which is a common coarse-grained resolution (Charron et al., 2023). With this coarse-graining scheme, up to 10 atoms are retained per molecule, as seen in Figure 5 (a). In total, we simulate $30ns$ for each dipeptide starting from 10 random conformations and a timestep of $0.5fs$.

**Overdispersion.** By inspecting the free energy in Figure 5 (b), we can observe that, again, all models are capable of learning to produce independent samples that resemble the reference distribution. The samples produced by *Transferable BG* contain more noise and also sample some unlikely states. While this does not produce statistically significant differences in Table 2, it produces a higher mean with a similar variance than other methods. Further, note that this model does not support simulation.

When using the score for simulation, *Two For One* produces broader, overdispersed distributions, as visible in Figure 5 (b). This overdispersion also affects structural features such as bond and inter-atom distances, consistent with the behavior observed for alanine dipeptide (see Appendix C.4.1).

**Advantages of mixture.** For this specific dipeptide, *Both* provides no clear improvement in sampling or simulation over *Fokker-Planck*, which becomes apparent in Figure 5 (c). However, we have to consider two things: First, MoE reduces computational cost by applying training regularization only for small timescales and using simpler models for larger timescales, reducing sampling time by over 50% in this specific case (see Appendix C.3). And second, for some dipeptides (e.g., NY or RV), MoE is essential for

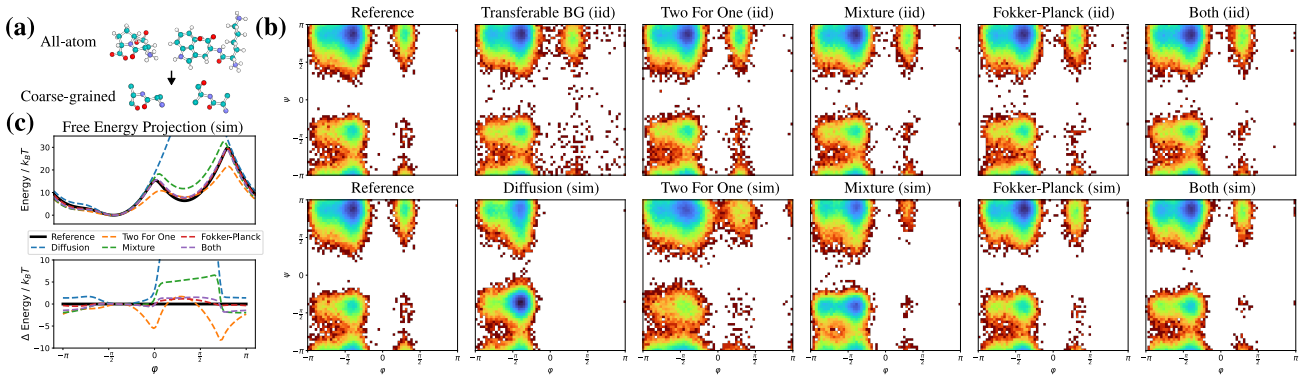


Figure 5: Comparison of methods on testset dipeptide AC. **(a)** The coarse-graining scheme. **(b)** Comparison of the Ramachandran plots of different methods for iid sampling and Langevin simulation. **(c)** The projection of the free energy surface and differences along the dihedral angle $\varphi$ for samples generated with simulation.

| Method | iid JS (↓) | sim. JS (↓) | iid PMF (↓) | sim. PMF (↓) |
|---|---|---|---|---|
| Transferable BG | **0.0183 ± 0.0070** | - | **0.230 ± 0.119** | - |
| Diffusion | **0.0155 ± 0.0083** | 0.2256 ± 0.1304 | **0.206 ± 0.159** | 6.515 ± 3.175 |
| Two For One | **0.0153 ± 0.0080** | 0.0466 ± 0.0114 | **0.203 ± 0.149** | 0.741 ± 0.319 |
| Mixture | **0.0155 ± 0.0078** | 0.0444 ± 0.0237 | **0.200 ± 0.127** | 0.658 ± 0.407 |
| Fokker-Planck | **0.0154 ± 0.0060** | **0.0200 ± 0.0106** | **0.192 ± 0.118** | **0.290 ± 0.222** |
| Both | **0.0158 ± 0.0077** | **0.0158 ± 0.0052** | **0.197 ± 0.124** | **0.183 ± 0.070** |

Table 2: Comparison of dipeptide AC with JS divergence and PMF error. To compute the mean and standard deviation, we have averaged the metrics across the dipeptides from the test set. Lower values are better.

consistency (see Appendix C.5), which also results in a significantly lower JS divergence and PMF error in Table 2. Also in this case, MoE improves the results over *Diffusion*.

**Fokker-Planck error.** Figure 6 shows the deviation from the Fokker-Planck equation, quantified as $\left\|\mathcal{F}_{p^{\boldsymbol{\theta}}}(\boldsymbol{x}, t) - \partial_t \log p_t^{\boldsymbol{\theta}}(\boldsymbol{x})\right\|_2$, plotted on a log scale. For models using MoE, this error is evaluated only up to $t = 0.1$, since only the small-timescale model is conservative. Across all methods, the error is highest near $t = 0$. Applying the Fokker-Planck regularization significantly reduces this error, correlating with the improved sampling-simulation consistency observed earlier.

Interestingly, while *Mixture* improves consistency, its Fokker-Planck error remains comparable to that of unregularized models. This suggests that Fokker-Planck regularization and MoE improve consistency through different mechanisms, which explains why combining them outperforms either approach on its own, making *Both* again clearly the best model (compare Table 2).

## 6. Conclusion, Limitations and Future Work

In this work, we investigated the gap between independent sampling and simulation using diffusion models. We introduced a Fokker–Planck-based regularization on the model's energy and showed that reducing the deviation from the

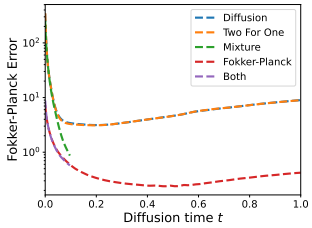


Figure 6: Comparing the Fokker-Planck error for $\log p^{\boldsymbol{\theta}}$ of multiple models. This figure plots the error over the diffusion time $t$. The y-axis is in log scale, and we can see that the largest error for all models is at $t = 0$.

Fokker–Planck equation improves the consistency of the model. Additionally, we demonstrated that restricting the model's focus to smaller diffusion timescales further improves simulation quality. We validated these findings across multiple systems, from toy examples to realistic biomolecular systems. This improved consistency enables the same model to be used for sampling and simulation, giving access to both kinetic and static properties.

Despite the theoretical motivation behind our approach, the results presented are primarily empirical. While our results indicate that reducing the Fokker–Planck deviation improves consistency, we do not claim this to be the only source of error. In fact, due to the fundamental differences between diffusion sampling and Langevin simulation, perfect alignment may not be achievable without limiting model expressivity. Additionally, evaluating the Fokker–Planck residual introduces computational overhead, which we mitigate through a weak residual formulation, although it still requires multiple forward passes of the model in training.

Future work could explore applying this approach to larger molecular systems, including proteins. It may also be promising to fine-tune pre-trained models with the proposed regularization or to train an auxiliary model to correct the identified inconsistencies.

## Acknowledgements

The authors would like to thank Aleksander Durumeric, Yaoyi Chen, Maximilian Schebek, Winfried Ripken, Marcel Kollovieh, Nicholas Gao, Matej Mezera, Yuanqi Du, and Jungyoon Lee for the fruitful discussions and their helpful input. The work of Michael Plainer was supported by the Konrad Zuse School of Excellence in Learning and Intelligent Systems (ELIZA) through the DAAD program Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the Federal Ministry of Education and Research. The work of Hao Wu was supported by the National Natural Science Foundation of China, Grant No. 12171367. Moreover, we gratefully acknowledge support by the Deutsche Forschungsgemeinschaft (SFB1114, Projects No. A04 and No. B08) and the Berlin Mathematics center MATH+ (AA1-10). We gratefully acknowledge the computing time made available to them on the high-performance computer "Lise" at the NHR Center NHR@ZIB. This center is jointly supported by the Federal Ministry of Education and Research and the state governments participating in the NHR (`www.nhr-verein.de/unsere-partner`).

## References

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

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., Bodenstein, S. W., Evans, D. A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A. I., Cowie, A., Figurnov, M., Fuchs, F. B., Gladman, H., Jain, R., Khan, Y. A., Low, C. M. R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E. D., Zielinski, M., Žídek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., and Jumper, J. M. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, May 2024. ISSN 1476-4687.

Arts, M., Garcia Satorras, V., Huang, C.-W., Zügner, D., Federici, M., Clementi, C., Noé, F., Pinsler, R., and van den Berg, R. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *Journal of Chemical Theory and Computation*, 19(18):6151–6159, 2023.

Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Zhang, Q., Kreis, K., Aittala, M., Aila, T., Laine, S., Catanzaro, B., Karras, T., and Liu, M.-Y. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers, 2023.

Boltzmann, L. *Studien über das Gleichgewicht der lebendigen Kraft zwischen bewegten materiellen Punkten: vorgelegt in der Sitzung am 8. October 1868*. k. und k. Hof- und Staatsdr., 1868.

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

Charron, N. E., Musil, F., Guljas, A., Chen, Y., Bonneau, K., Pasos-Trejo, A. S., Venturin, J., Gusew, D., Zaporozhets, I., Krämer, A., et al. Navigating protein landscapes with a machine-learned transferable coarse-grained model. *arXiv preprint arXiv:2310.18278*, 2023.

Clementi, C. Coarse-grained models of protein folding: toy models or predictive tools? *Current opinion in structural biology*, 18(1):10–15, 2008.

Corso, G., Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T. DiffDock: Diffusion steps, twists, and turns for molecular docking. In *International Conference on Learning Representations*, 2023.

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

Diez, J. V., Atance, S. R., Engkvist, O., and Olsson, S. Generation of conformational ensembles of small molecules via surrogate model-assisted molecular dynamics. *Machine Learning: Science and Technology*, 5(2):025010, 2024.

Du, Y., Durkan, C., Strudel, R., Tenenbaum, J. B., Dieleman, S., Fergus, R., Sohl-Dickstein, J., Doucet, A., and Grathwohl, W. S. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8489–8510. PMLR, 23–29 Jul 2023.

Du, Y., Plainer, M., Brekelmans, R., Duan, C., Noé, F., Gomes, C. P., Aspuru-Guzik, A., and Neklyudov, K. Doob's lagrangian: A sample-efficient variational approach to transition path sampling. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 65791–65822. Curran Associates, Inc., 2024.

Durumeric, A. E., Charron, N. E., Templeton, C., Musil, F., Bonneau, K., Pasos-Trejo, A. S., Chen, Y., Kelkar, A., Noé, F., and Clementi, C. Machine learned coarse-grained protein force-fields: Are we there yet? *Current opinion in structural biology*, 79:102533, 2023.

Durumeric, A. E. P., Chen, Y., Noé, F., and Clementi, C. Learning data efficient coarse-grained molecular dynamics from forces and noise, 2024.

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.

Fornberg, B. Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation*, 51 (184):699–706, 1988.

Ganjdanesh, A., Kang, Y., Liu, Y., Zhang, R., Lin, Z., and Huang, H. Mixture of efficient diffusion experts through automatic interval and sub-network selection. In Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., and Varol, G. (eds.), *Computer Vision – ECCV 2024*, pp. 54–71, Cham, 2025. Springer Nature Switzerland.

Guo, L., Wu, H., and Zhou, T. Normalizing field flows: Solving forward and inverse stochastic differential equations using physics-informed flow models. *Journal of Computational Physics*, 461:111202, 2022. ISSN 0021-9991.

Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., and van Zee, M. Flax: A neural network library and ecosystem for JAX, 2024. URL `http://github.com/google/flax`.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020.

Hoffmann, M., Scherer, M. K., Hempel, T., Mardt, A., de Silva, B., Husic, B. E., Klus, S., Wu, H., Kutz, J. N., Brunton, S., and Noé, F. Deeptime: a python library for machine learning dynamical models from time series data. *Machine Learning: Science and Technology*, 2021.

Hu, Z., Zhang, Z., Karniadakis, G. E., and Kawaguchi, K. Score-based physics-informed neural networks for high-dimensional fokker-planck equations, 2024.

Husic, B. E., Charron, N. E., Lemm, D., Wang, J., Pérez, A., Majewski, M., Krämer, A., Chen, Y., Olsson, S., De Fabritiis, G., et al. Coarse graining molecular dynamics with graph neural networks. *The Journal of chemical physics*, 153(19), 2020.

Kim, D., Shin, S., Song, K., Kang, W., and Moon, I.-C. Soft truncation: A universal training technique of score-based diffusion model for high precision score estimation. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11201–11228. PMLR, 17–23 Jul 2022.

Kim, J. C., Bloore, D., Kapoor, K., Feng, J., Hao, M.-H., and Wang, M. Scalable normalizing flows enable boltzmann generators for macromolecules. In *International Conference on Learning Representations (ICLR)*, 2024.

Klein, L. and Noé, F. Transferable Boltzmann generators. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 45281–45314. Curran Associates, Inc., 2024.

Klein, L., Foong, A., Fjelde, T., Mlodozeniec, B., Brockschmidt, M., Nowozin, S., Noe, F., and Tomioka, R. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 52863–52883. Curran Associates, Inc., 2023a.

Klein, L., Krämer, A., and Noé, F. Equivariant flow matching. *Advances in Neural Information Processing Systems*, 36:59886–59910, 2023b.

Koehler, F., Heckett, A., and Risteski, A. Statistical efficiency of score matching: The view from isoperimetry. In *International Conference on Learning Representations*, 2023.

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*, 2020. URL `http://proceedings.mlr.press/v119/kohler20a.html`.

Köhler, J., Krämer, A., and Noé, F. Smooth normalizing flows. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 2796–2809. Curran Associates, Inc., 2021. URL `https://proceedings.`

neurips.cc/paper/2021/file/167434fa6219316417cd4160c0c5e7d2-Paper.pdf.

Köhler, J., Chen, Y., Krämer, A., Clementi, C., and Noé, F. Flow-matching: Efficient coarse-graining of molecular dynamics without forces. *Journal of Chemical Theory and Computation*, 19(3):942–952, 2023.

Krämer, A., Durumeric, A. E., Charron, N. E., Chen, Y., Clementi, C., and Noé, F. Statistically optimal force aggregation for coarse-graining molecular dynamics. *The Journal of Physical Chemistry Letters*, 14(17):3970–3979, 2023.

Lai, C.-H., Takida, Y., Murata, N., Uesaka, T., Mitsufuji, Y., and Ermon, S. FP-Diffusion: Improving score-based diffusion models by enforcing the underlying score Fokker-Planck equation. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 18365–18398. PMLR, 07 2023.

Leimkuhler, B. and Matthews, C. *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*. Springer International Publishing, 2015.

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

Li, P., Li, Z., Zhang, H., and Bian, J. On the generalization properties of diffusion models. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 2097–2127. Curran Associates, Inc., 2023.

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

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *Energy Based Models Workshop*, 2019.

McGibbon, R. T., Beauchamp, K. A., Harrigan, M. P., Klein, C., Swails, J. M., Hernández, C. X., Schwantes, C. R., Wang, L.-P., Lane, T. J., and Pande, V. S. Mdtraj: A modern open library for the analysis of molecular dynamics trajectories. *Biophysical Journal*, 109(8):1528 – 1532, 2015.

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*, 2023. URL https://openreview.net/forum?id=XCTVFJwS9LJ.

Müller, K. and Brown, L. D. Location of saddle points and minimum energy paths by a constrained simplex optimization procedure. *Theoretica Chimica Acta*, 53(1): 75–93, 1979.

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

Noid, W. G. Perspective: Coarse-grained models for biomolecular systems. *The Journal of chemical physics*, 139(9), 2013.

Øksendal, B. *Stochastic Differential Equations*, pp. 65–84. Springer Berlin Heidelberg, 2003.

Plainer, M., Stärk, H., Bunne, C., and Günnemann, S. Transition path sampling with boltzmann generator-based mcmc moves. In *Generative AI and Biology Workshop*, 2023a.

Plainer, M., Toth, M., Dobers, S., Stärk, H., Corso, G., Marquet, C., and Barzilay, R. DiffDock-Pocket: Diffusion for pocket-level docking with sidechain flexibility. In *Machine Learning in Structural Biology*, 2023b.

Salimans, T. and Ho, J. Should EBMs model the energy or the score? In *Energy Based Models Workshop*, 2021.

Särkkä, S. and Solin, A. *Applied Stochastic Differential Equations*. Cambridge University Press, April 2019.

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

Schreiner, M., Winther, O., and Olsson, S. Implicit transfer operator learning: Multiple time-resolution models for molecular dynamics. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=1kZx7JiuA2.

Shi, Y., Huang, Z., Feng, S., Zhong, H., Wang, W., and Sun, Y. Masked label prediction: Unified message passing model for semi-supervised classification. In Zhou, Z.-H. (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 1548–1554. International Joint Conferences on Artificial Intelligence Organization, 8 2021.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In Wallach, H.,

Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Tamagnone, S., Laio, A., and Gabrié, M. Coarse-grained molecular dynamics with normalizing flows. *Journal of Chemical Theory and Computation*, 20(18):7796–7805, 2024.

Tan, C. B., Bose, A. J., Lin, C., Klein, L., Bronstein, M. M., and Tong, A. Scalable equilibrium sampling with sequential boltzmann generators. *arXiv preprint arXiv:2502.18462*, 2025.

Trippe, B. L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R., and Jaakkola, T. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. In *International Conference on Learning Representations*, 2023.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Hanikel, N., Pellock, S. J., Courbet, A., Sheffler, W., Wang, J., Venkatesh, P., Sappington, I., Torres, S. V., Lauko, A., De Bortoli, V., Mathieu, E., Ovchinnikov, S., Barzilay, R., Jaakkola, T. S., DiMaio, F., Baek, M., and Baker, D. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, July 2023.

Wirnsberger, P., Ballard, A. J., Papamakarios, G., Abercrombie, S., Racaniere, S., Pritzel, A., Jimenez Rezende, D., and Blundell, C. Targeted free energy estimation via learned mappings. *The Journal of Chemical Physics*, 153 (14):144112, 2020.

Wolf, S., Lickert, B., Bray, S., and Stock, G. Multisecond ligand dissociation dynamics from atomistic simulations. *Nature Communications*, 11(1), June 2020.

Wu, H. and Noé, F. Reaction coordinate flows for model reduction of molecular kinetics. *The Journal of Chemical Physics*, 160(4), January 2024.

# A. Proofs, Derivations, and Mathematical Details

## A.1. Diffusion

We have opted to use VP diffusion (Song et al., 2021) throughout the paper, as such the drift and diffusion can be written as

$$\boldsymbol{f}(\boldsymbol{x}, t) = -\frac{1}{2}\beta(t)\boldsymbol{x}, \quad g(t) = \sqrt{\beta(t)}, \tag{17}$$

where

$$\beta(t) = \beta_{\text{min}} + t \cdot (\beta_{\text{max}} - \beta_{\text{min}}), \tag{18}$$

with the hyperparameters from (Song et al., 2021) such that $(\beta_{\text{min}}, \beta_{\text{max}}) = (0.1, 20)$. For this noise schedule to be suitable for molecules, it is important that we normalize the data to have unit variance.

With this specific choice for $\boldsymbol{f}$ and $g$, we can write the transition kernel as a Gaussian (Särkkä & Solin, 2019) with a moving mean and standard deviation such that

$$p_t(\boldsymbol{x}(t) \,|\, \boldsymbol{x}(0)) = \mathcal{N}(\boldsymbol{x}(t); \mu(\boldsymbol{x}(0), t), \sigma(t) I) \tag{19}$$

$$= \mathcal{N}(\boldsymbol{x}(t); e^{-\frac{1}{2}\int_0^t \beta(s)ds}\boldsymbol{x}(0), (1 - e^{-\int_0^t \beta(s)ds})I). \tag{20}$$

## A.2. Residual Loss

In this section, we prove Theorem 3.1 and show that $\widetilde{R}(\boldsymbol{x}, t; \boldsymbol{v})$ can be used to get an unbiased estimation of $\widetilde{R}(\boldsymbol{x}, t)$. For simplicity of notation, let us express the log Fokker-Planck equations from Equation (8) as

$$\frac{1}{2}g^2(t)\,\text{div}_{\boldsymbol{x}}\, \boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x}, t) + \gamma_{\boldsymbol{\theta}}(\boldsymbol{x}, t) = 0, \tag{21}$$

where $\boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x}, t) = \nabla_{\boldsymbol{x}} \log p_t^{\boldsymbol{\theta}}(\boldsymbol{x})$, and $\gamma_{\boldsymbol{\theta}}$ involves only the first-order gradient of $\log p_t^{\boldsymbol{\theta}}$. Here we define the "weak" residual (Guo et al., 2022) of the above equations for each $(\boldsymbol{x}, t)$

$$\widetilde{R}(\boldsymbol{x}, t) = \mathbb{E}_{\boldsymbol{v}\sim\mathcal{N}(0,\sigma^2 I)}\left[\frac{1}{2}g^2(t)\,\text{div}_{\boldsymbol{x}}\, \boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{v}, t) + \gamma_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{v}, t)\right], \tag{22}$$

where $\sigma > 0$ is a small number. It can be seen that residuals are zero if the two parts of the equations are exactly equal. We now aim to get the unbiased estimation of the above residual, without calculating high-order derivatives.

As such, we can show that for an arbitrary $t$,

$$\mathbb{E}_{\boldsymbol{v}}\left[\text{div}_{\boldsymbol{x}}\, \boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{v}, t)\right] = \int \frac{\exp\left(-\frac{\boldsymbol{v}^\top \boldsymbol{v}}{2\sigma^2}\right)}{(2\pi\sigma^2)^{\frac{D}{2}}} \cdot \text{div}_{\boldsymbol{x}}\, \boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{v}, t) dv \tag{23}$$

$$= -\frac{1}{(2\pi\sigma^2)^{\frac{D}{2}}}\int \left\langle \nabla_{\boldsymbol{v}} \exp\left(-\frac{\boldsymbol{v}^\top \boldsymbol{v}}{2\sigma^2}\right), \boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{v}, t)\right\rangle d\boldsymbol{v} \tag{24}$$

$$= \frac{1}{(2\pi\sigma^2)^{\frac{D}{2}}}\int \exp\left(-\frac{\boldsymbol{v}^\top \boldsymbol{v}}{2\sigma^2}\right)\frac{\boldsymbol{v}^\top}{\sigma^2}\boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{v}, t) dv \tag{25}$$

$$= \mathbb{E}_{\boldsymbol{v}}\left[\frac{\boldsymbol{v}^\top \boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{v}, t)}{\sigma^2}\right] \tag{26}$$

$$= \frac{1}{2}\mathbb{E}_{\boldsymbol{v}}\left[\frac{\boldsymbol{v}^\top \boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{v}, t)}{\sigma^2} - \frac{\boldsymbol{v}^\top \boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x} - \boldsymbol{v}, t)}{\sigma^2}\right] \tag{27}$$

$$= \mathbb{E}_{\boldsymbol{v}}\left[\left(\frac{\boldsymbol{v}}{\sigma}\right)^\top \frac{\boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{v}, t) - \boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x} - \boldsymbol{v}, t)}{2\sigma}\right], \tag{28}$$

where $\boldsymbol{x} \in \mathbb{R}^D$, $\boldsymbol{v} \in \mathbb{R}^D$ and $t \in \mathbb{R}$.

Based on this, we can obtain

$$\widetilde{R}(\boldsymbol{x}, t) = \mathbb{E}_{\boldsymbol{v}}\left[\widetilde{R}(\boldsymbol{x}, t; \boldsymbol{v})\right] \tag{29}$$

$$= \mathbb{E}_{\boldsymbol{v}}\left[\frac{1}{2} g^2(t) \left(\frac{\boldsymbol{v}}{\sigma}\right)^\top \frac{\boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{v}, t) - \boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x}-\boldsymbol{v}, t)}{2\sigma} + \frac{\gamma_\theta(\boldsymbol{x}+\boldsymbol{v}, t) + \gamma_{\boldsymbol{\theta}}(\boldsymbol{x}-\boldsymbol{v}, t)}{2}\right]. \tag{30}$$

Hence, $\widetilde{R}(\boldsymbol{x}, t; \boldsymbol{v})$ is an unbiased estimation of $\widetilde{R}(\boldsymbol{x}, t)$ by drawing a single sample $\boldsymbol{v} \sim \mathcal{N}(0, \sigma^2 I)$.

In practice, we can further reduce the computational overhead by only using a single approximation for $\gamma_{\boldsymbol{\theta}}$, and defining

$$\widetilde{R}(\boldsymbol{x}, t; \boldsymbol{v}) = \frac{1}{2} g^2(t) \left(\frac{\boldsymbol{v}}{\sigma}\right)^\top \frac{\boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{v}, t) - \boldsymbol{s}^{\boldsymbol{\theta}}(\boldsymbol{x}-\boldsymbol{v}, t)}{2\sigma} + \gamma_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{v}, t). \tag{31}$$

We found $\sigma = 0.0001$ to be an effective choice throughout our experiments.

### A.3. Finite Difference Approximation

To approximate $\partial_t \log p^{\boldsymbol{\theta}}$, we relied on a finite difference approximation (Fornberg, 1988), as stated in Equation (15). For this estimation, we have followed the work of (Lai et al., 2023), and used the hyperparameters that they suggested $(h_s, h_d) = (0.001, 0.0005)$.

## B. Details for Experiments

### B.1. Metrics

To compute the JS divergence and the PMF error, we first discretize the observed free energy into binned histograms. For the JS divergence, we then compute the JS distance between the two probability vectors (we flatten the 2D histograms). To prevent discontinuities, we assume that in each bin there is at least one observation by adding 1.

As for the PMF error, we discretize into 64 bins and compute the proportion of samples in each window. These are then transformed by taking the $\log$ in each bin and then computing the square loss, which is averaged over all bins. Similarly, we have ensured that each bin contains some data and have added $10^{-6}$ as a baseline proportion. The approach and implementation are analogous to (Durumeric et al., 2024).

### B.2. Toy System

**Dataset.** We have used a version of the Müller-Brown potential (Müller & Brown, 1979) to demonstrate the capabilities of our approach in two dimensions. For this, we have used the following potential

$$\begin{aligned} U(x, y) = &-200 \cdot \exp\left(-(x-1)^2 - 10y^2\right) \\ &-100 \cdot \exp\left(-x^2 - 10 \cdot (y-0.5)^2\right) \\ &-170 \cdot \exp\left(-6.5 \cdot (0.5+x)^2 + 11 \cdot (x+0.5) \cdot (y-1.5) - 6.5 \cdot (y-1.5)^2\right) \\ &+\ 15 \cdot \exp\left(0.7 \cdot (1+x)^2 + 0.6 \cdot (x+1) \cdot (y-1) + 0.7 \cdot (y-1)^2\right). \end{aligned} \tag{32}$$

To generate training samples from this potential, we have performed a Langevin simulation (compare Equation (6)). For this, we have performed 5M steps with $k_B T = 23$, $dt = 0.005$, $\boldsymbol{M} = 0.5 \cdot I$, where we only store every 50th sample to generate 100k training samples.

**Architecture and training.** For the toy systems, we have used a simple multi-layer perception with the hyperparameters presented in Table 3. As for the optimizer, we have used AdamW (Loshchilov & Hutter, 2019).

### B.3. Formalization Coarse-Craining

In coarse-graining, we aim to reduce the number of dimensions of our system by combining multiple atoms into individual beads. Given non-CG samples $\boldsymbol{x}$, the Boltzmann distribution of CG samples $\boldsymbol{z}$ can be recovered by $p(\boldsymbol{z}) \propto \int \exp\left(-\frac{U(\boldsymbol{x})}{k_B T}\right) \delta(\Xi(\boldsymbol{x}) - \boldsymbol{z})\, \mathrm{d}\boldsymbol{x}$ which defines the CG potential up to a constant. $\delta$ is the Dirac delta function.

| Parameter | Diffusion | Mixture | Fokker-Planck | Both |
|---|---|---|---|---|
| # Parameters | 17849 | 17263 | 17849 | 17263 |
| BS | 128 | 128 | 128 | 128 |
| Model-Ranges | (0, 1) | (0, 0.1), [0.1, 0.6), [0.6, 1.0) | (0, 1) | (0, 0.1), [0.1, 0.6), [0.6, 1.0) |
| Epochs | 180 | 120, 30, 30 | 180 | 120, 30, 30 |
| Hidden Layers | [92, 92, 92] | [64, 64, 64], [64, 64], [54, 54] | [92, 92, 92] | [64, 64, 64], [64, 64], [54, 54] |
| $\alpha$ | 0 | 0 | 0.0005 | 0.0005, 0, 0 |

Table 3: This table contains the hyperparameters for the different models shown for the Müller-Brown potential.

### B.4. Alanine Dipeptide

**Dataset.** The alanine dipeptide datasets is available as part of the public `bgmol` (MIT licence) repository here: `https://github.com/noegroup/bgmol`. The dataset was generated with an MD simulation, using the classical *Amber ff99SBildn* force-field at 300K for implicit solvent for a duration of 1ms (Köhler et al., 2021) with the `openMM` library (Eastman et al., 2017). For training, we have selected 50k random samples from this simulation.

**Architecture.** For alanine dipeptide we have used quite a small architecture, where the hyperparameters are listed in Table 4. When multiple parameters are listed for the same model, this means that they are used for the corresponding MoE model. Note that when using MoE, we have mostly used the same model architecture, except that only the Fokker-Planck regularized model is conservative.

| Parameter | Diffusion | Mixture | Fokker-Planck | Both |
|---|---|---|---|---|
| Epochs | 10000 | 7000, 2000, 1000 | 10000 | 7000, 2000, 1000 |
| BS | 1024 | 1024 | 1024 | 1024 |
| Attention Heads | 8 | 8 | 8 | 8 |
| Feature Dim | 16 | 16 | 16 | 16 |
| Model-Ranges | (0, 1) | (0, 0.1), [0.1, 0.6), [0.6, 1.0) | (0, 1) | (0, 0.1), [0.1, 0.6), [0.6, 1.0) |
| Conservative | Yes | Yes, No, No | Yes | Yes, No, No |
| $\alpha$ | 0 | 0, 0, 0 | 0.0005 | 0.0001, 0, 0 |
| Hidden Dimension | 96 | 96 | 96 | 96 |
| Layers | 2 | 2 | 2 | 2 |

Table 4: This table contains the hyperparameters for the different models shown for alanine dipeptide.

**Simulation.** To perform Langevin simulation, we have extracted the forces from the model via Equation (7) at $t = 10^{-5}$ for all models except for *Two For One*, where we chose $t = 0.02$, the same hyperparameter as presented in (Arts et al., 2023).

### B.5. Dipeptides (2AA)

**Dataset.** The original dipeptide dataset (2AA) was introduced in (Klein et al., 2023a) (MIT License) and is available here: `https://huggingface.co/datasets/microsoft/timewarp`. As this includes a lot of intermediate saved states and quantities, like energies, there is a smaller version made available by Klein & Noé (2024) (CC BY 4.0): `https://osf.io/n8vz3/?view_only=1052300a21bd43c08f700016728aa96e`. For a comprehensive overview of the simulation details, refer to (Klein et al., 2023a). All dipeptides were simulated in implicit solvent with a classical *amber-14* force-field at $T = 310$K. The simulation of the training and validation peptides were run for 50ns, while the test peptides were simulated for $1\mu$s. All simulation were performed with the `openMM` library (Eastman et al., 2017).

Note that we have removed dipeptides containing Glycine from our dataset to ensure that all dipeptides have the same number of (coarse-grained) atoms. This made it easier to handle it in the code, but it is not a technical limitation of our architecture. It is split into 175 train, 75 validation, and 92 test dipeptides, out of which we have used 15 for the results in the paper (also the metrics) to reduce on inference time.

**Architecture.** The hyperparameters are listed in Table 5. When multiple parameters are listed for the same model, this means that they are used for the corresponding MoE model. Note that when using MoE, we have used smaller networks for larger diffusion times, and only the Fokker-Planck regularized model is conservative.

**Simulation.** To perform Langevin simulation, we have extracted the forces from the model via Equation (7) at $t = 10^{-5}$ for

| Parameter | Diffusion | Mixture | Fokker-Planck | Both |
|---|---|---|---|---|
| Epochs | 120 | 100, 20, 10 | 120 | 100, 20, 10 |
| BS | 1024 | 1024 | 1024 | 1024 |
| Attention Heads | 8 | 8 | 8 | 8 |
| Feature Dim | 16 | 16 | 16 | 16 |
| Model-Ranges | (0, 1) | (0, 0.1), [0.1, 0.6), [0.6, 1.0) | (0, 1) | (0, 0.1), [0.1, 0.6), [0.6, 1.0) |
| Conservative | Yes | Yes, No, No | Yes | Yes, No, No |
| $\alpha$ | 0 | 0, 0, 0 | 0.0005 | 0.0001, 0, 0 |
| Hidden Dimension | 128 | 128, 96, 96 | 128 | 128, 96, 96 |
| Layers | 3 | 3, 2, 2 | 3 | 3, 2, 2 |

Table 5: This table contains the hyperparameters for the different models shown for the minipeptides.

all models except for *Two For One*. As this system has not been tested by (Arts et al., 2023), we opted to use the same $t$ as for Alanine dipeptide, namely $t = 0.02$, which yielded robust results.

### B.6. Compute Infrastructure

We have used a single RTX 3090 GPU for the toy systems, an A100 with 80GB memory for alanine dipeptide, and two A100 80GB GPUs for the dipeptide dataset.

### B.7. Software Licences

In our code, we have used `jax` (Bradbury et al., 2018) (Apache-2.0) and the accompanying machine learning library `flax` (Heek et al., 2024) (Apache-2.0). For the graph transformer architecture, we have extended code from (Arts et al., 2023) (MIT) and have re-implemented the code from `https://github.com/lucidrains/graph-transformer-pytorch` (MIT) in `jax`.

For the free-energy plots of the Müller-Brown potential, we used (Hoffmann et al., 2021) (LGPL-3.0). For trajectories and simulations, we have used `openMM` (Eastman et al., 2017) (MIT) and `mdtraj` (McGibbon et al., 2015) (LGPL-2.1).

## C. Ablation Studies and More Experiments

### C.1. Müller-Brown Potential

We would like to complement Figure 2 from the main paper with numerical values. We have used the same metrics as for the other experiments and illustrate the results in Table 6.

| Method | iid JS (↓) | sim JS (↓) | iid PMF (↓) | sim PMF (↓) |
|---|---|---|---|---|
| Reference | $0.0119 \pm 0.0004$ | | $0.087 \pm 0.002$ | |
| Diffusion | $\mathbf{0.0122 \pm 0.0013}$ | $0.0448 \pm 0.0125$ | $0.111 \pm 0.006$ | $0.504 \pm 0.150$ |
| Mixture | $\mathbf{0.0109 \pm 0.0007}$ | $0.0254 \pm 0.0109$ | $\mathbf{0.097 \pm 0.004}$ | $0.247 \pm 0.113$ |
| Fokker-Planck | $0.0130 \pm 0.0010$ | $0.0166 \pm 0.0009$ | $0.122 \pm 0.006$ | $0.163 \pm 0.008$ |
| Both | $\mathbf{0.0110 \pm 0.0007}$ | $\mathbf{0.0108 \pm 0.0008}$ | $\mathbf{0.098 \pm 0.003}$ | $\mathbf{0.099 \pm 0.004}$ |

Table 6: Comparison of methods based on JS Divergence and the PMF error. Lower values are better. To compute the standard deviation, we have trained ten different models and performed sampling/simulation with them. As for the reference, we have started multiple simulations with a different seed on the same ground-truth potential. This serves as a reference of what could optimally be achieved.

### C.2. Comparing Conservative and Score-based Models

Previous work (Arts et al., 2023) suggested that conservative models improve the quality of the diffusion process. However, this effect was not observed for image data (Salimans & Ho, 2021). In Figure 7, we compare these approaches in practice. For iid sampling, conservative models provide a slight improvement. In contrast, for simulation, we were unable to train stable score-based models without a conservative parameterization. Using a conservative model yields much more stable forces, making simulation feasible. We attribute this to the smoother behavior of the conservative parameterization,

which prevents sudden changes in the score. Since the impact on iid sampling is negligible, we consider the conservative parameterization most relevant for small timescales, where model training is more sensitive. Consequently, in our MoE architecture, we apply the conservative parameterization only to the small-time diffusion model, achieving comparable or even superior iid sampling performance.

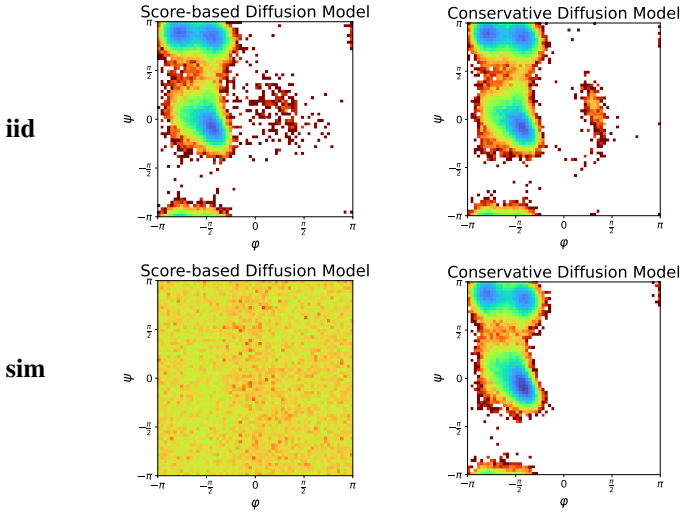


Figure 7: We compare a conservative diffusion model with a score-based model. We can see that around the low-density regions, the conservative parameterization generates better iid samples. As for simulation, a score-based model exhibits stability issues, and the simulation becomes unstable after a few thousand steps.

### C.3. Runtime Comparison

We compare the runtime of different approaches for training and inference in Table 7. Note that the MoE training could be parallelized. However, our current implementation is not optimized and does not distribute any training. In some cases, it even introduces overhead due to unoptimized implementation. Hence, we only see performance speedup for larger systems.

| Dataset | Task | Diffusion | Mixture | Fokker-Planck | Both |
|---|---|---|---|---|---|
| Alanine Dipeptide | Train | 49min | 50min | 4h 39min | 3h 59min |
| Alanine Dipeptide | Inference | 3min | 4min | 3min | 4min |
| Minipeptide | Train | 4h 5min | 3h 50min | 28h 39min | 27h 5min |
| Minipeptide | Inference | 8 min | 4min | 8min | 4min |

Table 7: We report the training and inference time for the different models.

### C.4. Alanine Dipeptide

In this section, we report some further results and plots on alanine dipeptide. In Figure 8 the Fokker-Planck residual error $\left\|\mathcal{F}_{p^{\boldsymbol{\theta}}}(\boldsymbol{x}, t) - \partial_t \log p_t^{\boldsymbol{\theta}}(\boldsymbol{x})\right\|_2$ is reported. Overall, the results are similar to what was reported in Figure 6. However, we can note that the Fokker-Planck error of *Mixture* is lower than *Diffusion*, indicating that MoE can improve the model's consistency.

In Figure 9 we compare the free energies along the dihedral angles $\varphi, \psi$ for iid sampling and simulation. We can see that the results from the main paper persist and that *Both* also shows the best performance for iid sampling.

Figure 4 shows all bond lengths of the coarse-grained molecule for iid sampling and Langevin simulation. Since *Two For One* does not evaluate the model at $t = 0$ it introduces noise across all bonds. We can also see this behavior by looking at the Wasserstein distance of the bond-lengths to the reference data as seen in Table 8.

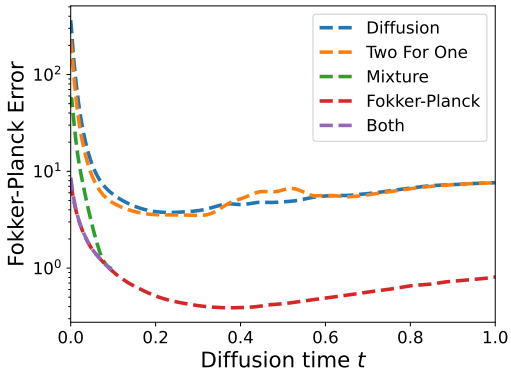


Figure 8: Comparing the Fokker-Planck error for $\log p^{\boldsymbol{\theta}}$ of multiple models. This figure shows the results for alanine dipeptide.

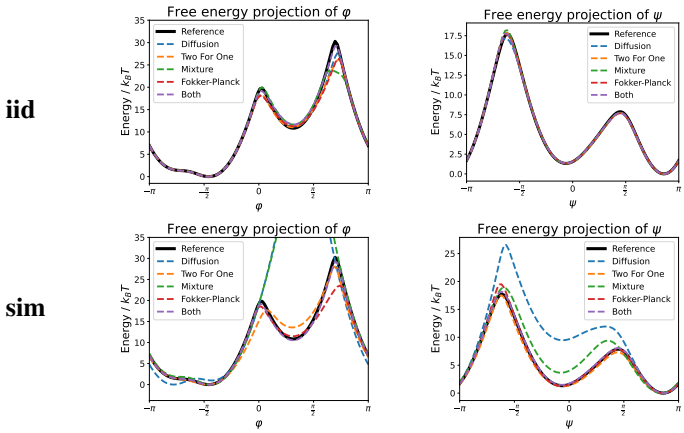


Figure 9: Comparing the free energy of alanine dipeptide along the dihedral angles $\varphi, \psi$ for iid sampling and simulation across different models.

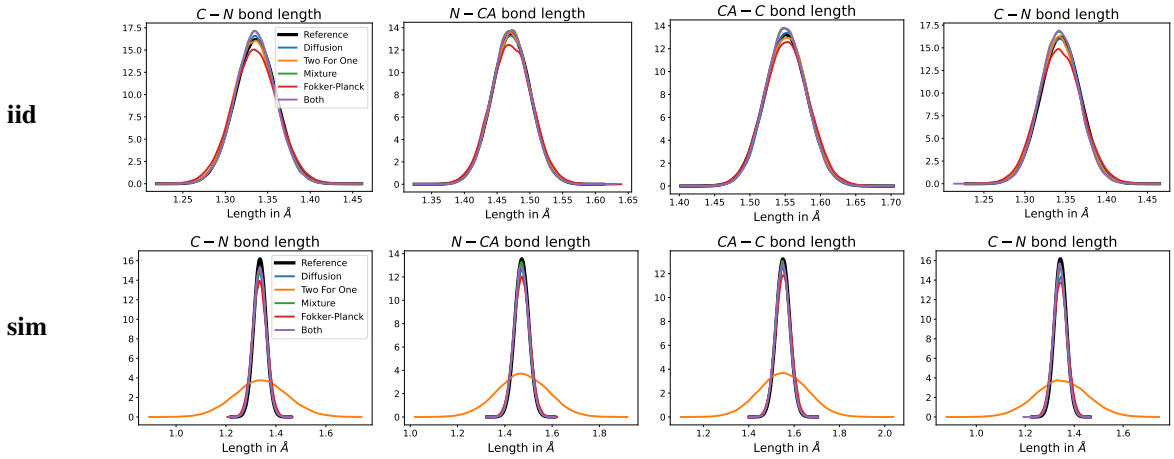


Figure 10: This illustration shows the sampled bond lengths for the molecule alanine dipeptide.

| Method | iid relative W1 (↓) | sim relative W1 (↓) |
|---|---|---|
| Diffusion | **1.51** ± **1.28** | 1.70 ± 0.38 |
| Two For One | **0.96** ± **0.34** | 48.14 ± 13.03 |
| Mixture | 1.36 ± 0.21 | **0.94** ± **0.21** |
| Fokker-Planck | 2.05 ± 0.62 | 2.51 ± 0.59 |
| Both | **1.00** ± **0.00** | **1.00** ± **0.00** |

Table 8: Comparison of methods based on the Wasserstein 1 distance of the C-N bond lengths to the reference data. Lower values are better. We have divided all entries by the Wasserstein 1 distance of *Both* so that the numbers are easier to compare. In other words, numbers larger than 1 mean that the bonds are worse than *Both*.

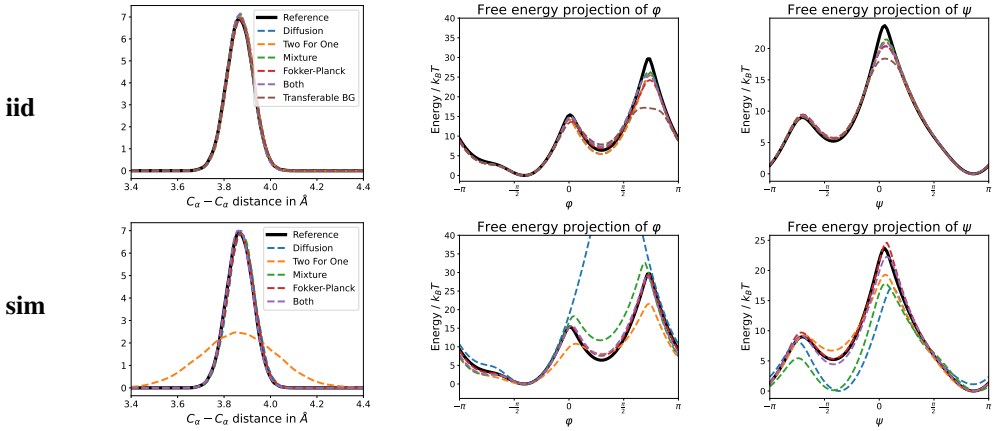


Figure 11: **AC:** We compare further metrics between iid sampling and Langevin simulation. We compare the $C_\alpha$–$C_\alpha$ distance for the dipeptides and also the free energy projections along the dihedral angles $\varphi, \psi$.

#### C.4.1. ALANINE-CYSTEINE (AC)

In Figure 11 we present extended results for the dipeptide investigated in the main part of the paper (AC). We can see that the results are consistent with what we presented and also the free energy surfaces on $\psi$ improve with Fokker-Planck regularization.

### C.5. Transferability: Results on More Dipeptides

In this section, we depict more dipeptides from the test set and demonstrate their performance. While the results are slightly different for each system, the general trends are consistent. We present the following dipeptides: KS Figures 12 and 13, HP Figures 14 and 15, NY Figures 16 and 17, TD Figures 18 and 19, and RV Figures 20 and 21.

## D. Societal Impact

Our work focuses on improving the efficiency of molecular sampling and simulation. We consider this research foundational, with the potential to accelerate applications such as drug and material discovery. While we do not identify any immediate risks, the technology could be misused, for example, in the development of biological weapons. Furthermore, our method currently lacks formal guarantees, which poses a risk of misleading downstream researchers if the method produces incorrect or biased results.

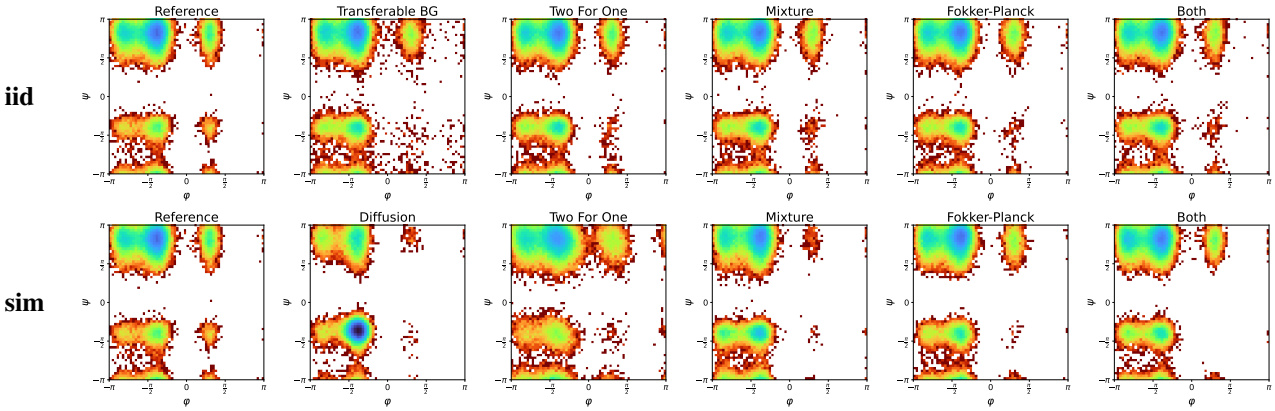


Figure 12: **KS:** We compare the free energy plot on the dihedral angles $\varphi, \psi$ for all presented methods for iid sampling and Langevin simulation.

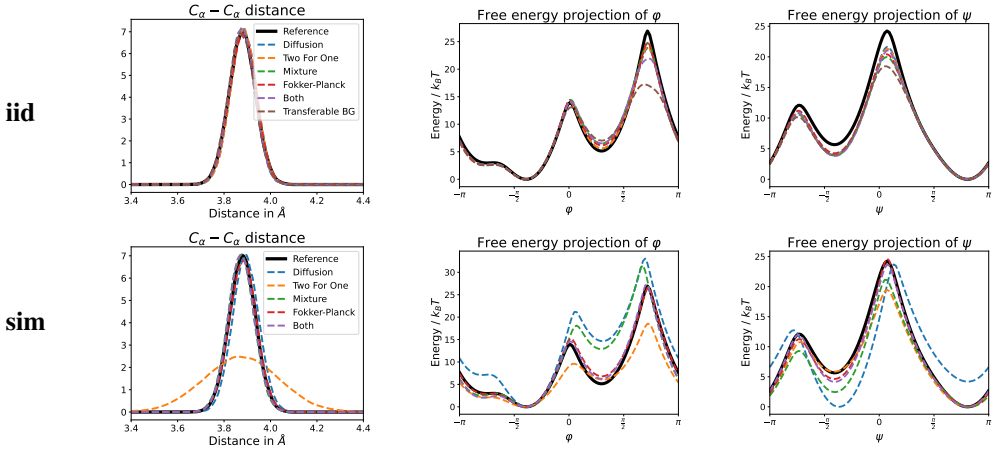


Figure 13: **KS:** We compare further metrics between iid sampling and Langevin simulation. We compare the $C_\alpha$–$C_\alpha$ distance for the dipeptides and also the free energy projections along the dihedral angles $\varphi, \psi$.

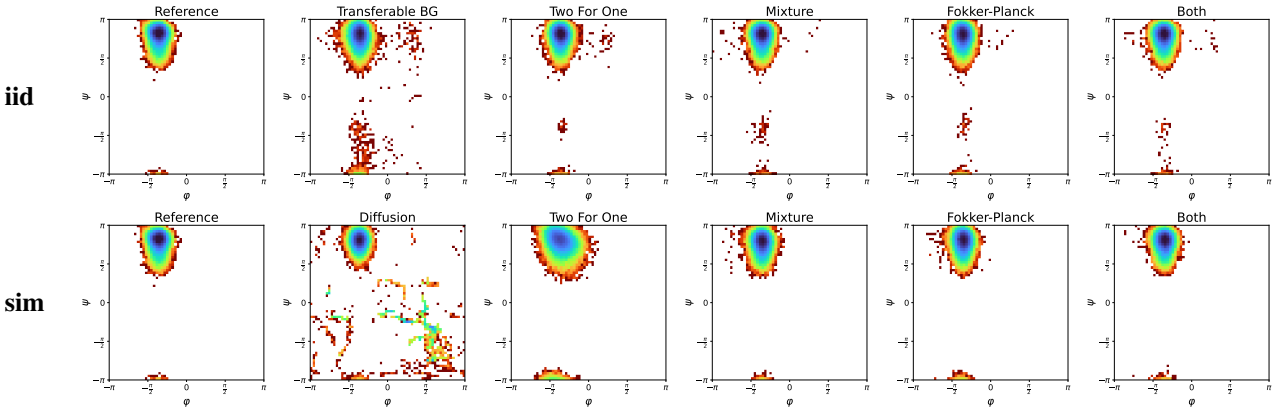


Figure 14: **HP:** We compare the free energy plot on the dihedral angles $\varphi, \psi$ for all presented methods for iid sampling and Langevin simulation.

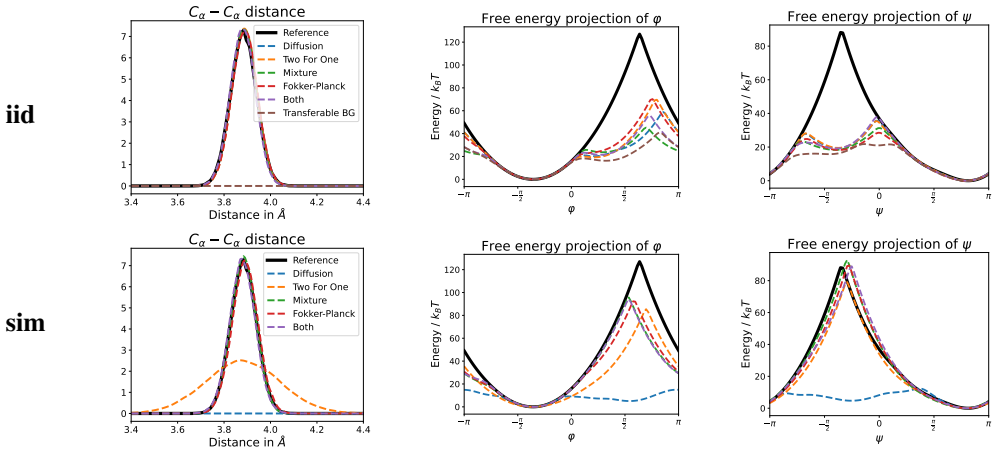


Figure 15: **HP:** We compare further metrics between iid sampling and Langevin simulation. We compare the $C_\alpha$–$C_\alpha$ distance for the dipeptides and also the free energy projections along the dihedral angles $\varphi, \psi$.

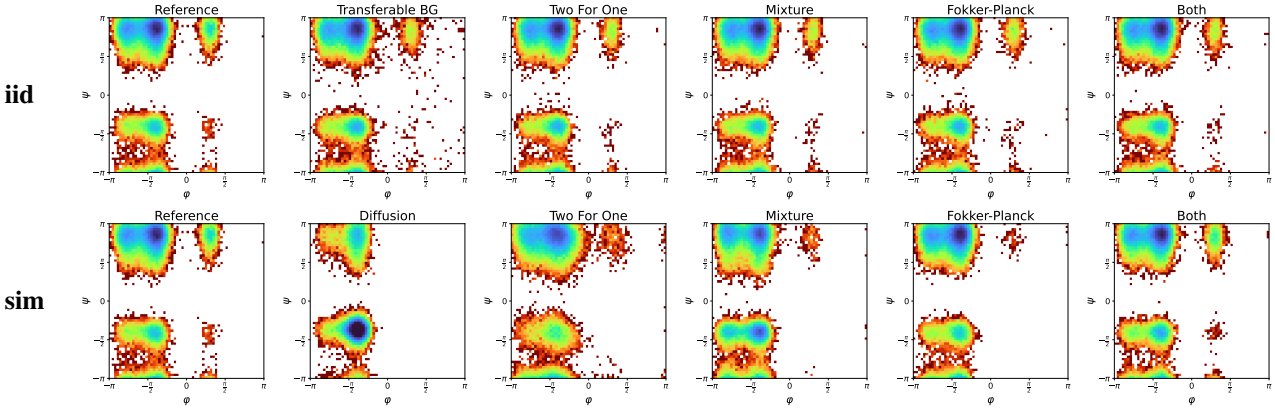


Figure 16: **NY:** We compare the free energy plot on the dihedral angles $\varphi, \psi$ for all presented methods for iid sampling and Langevin simulation.

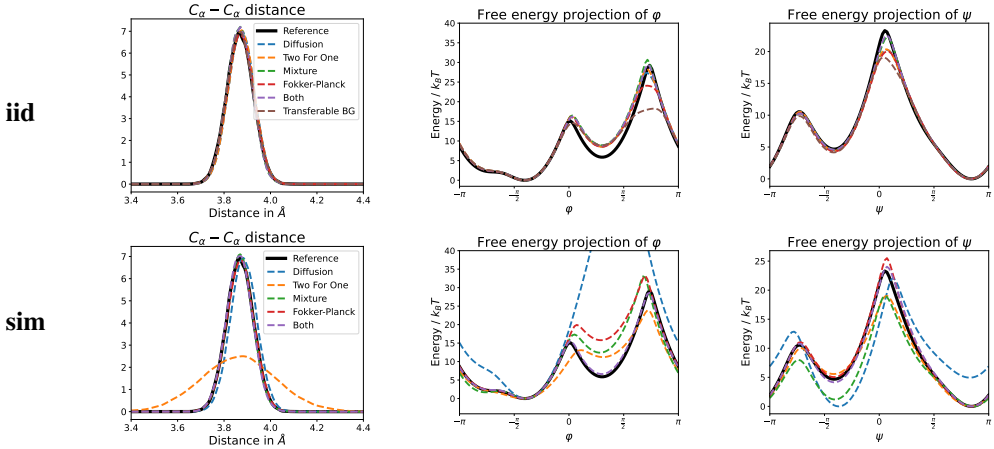


Figure 17: **NY:** We compare further metrics between iid sampling and Langevin simulation. We compare the $C_\alpha$–$C_\alpha$ distance for the dipeptides and also the free energy projections along the dihedral angles $\varphi, \psi$.

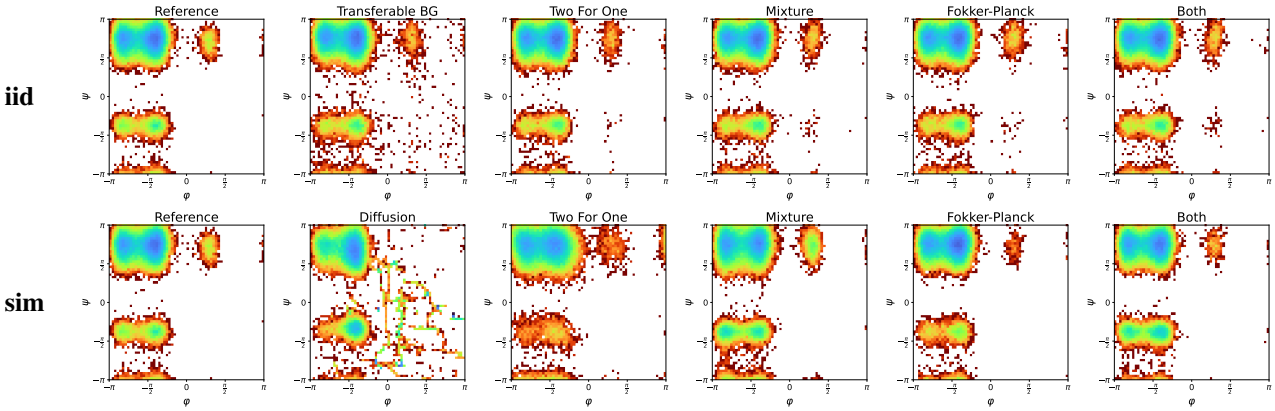


Figure 18: **TD:** We compare the free energy plot on the dihedral angles $\varphi, \psi$ for all presented methods for iid sampling and Langevin simulation.

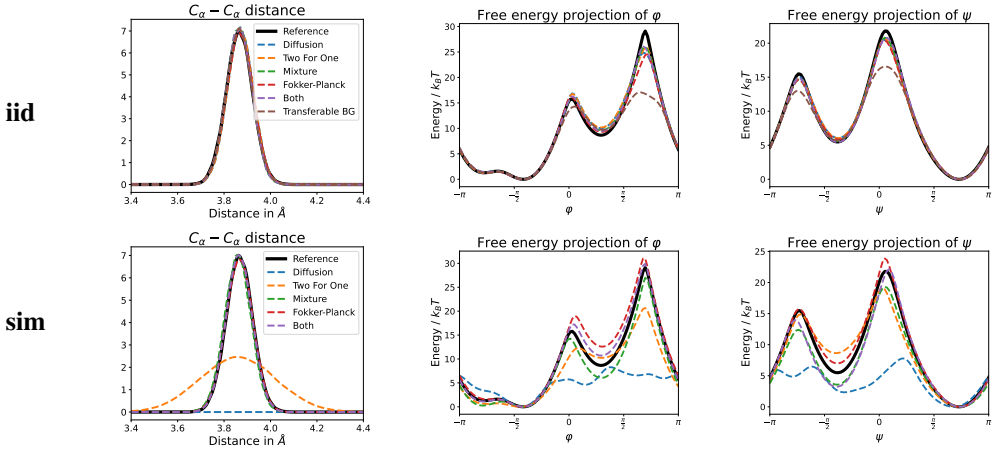


Figure 19: **TD:** We compare further metrics between iid sampling and Langevin simulation. We compare the $C_\alpha$–$C_\alpha$ distance for the dipeptides and also the free energy projections along the dihedral angles $\varphi, \psi$.

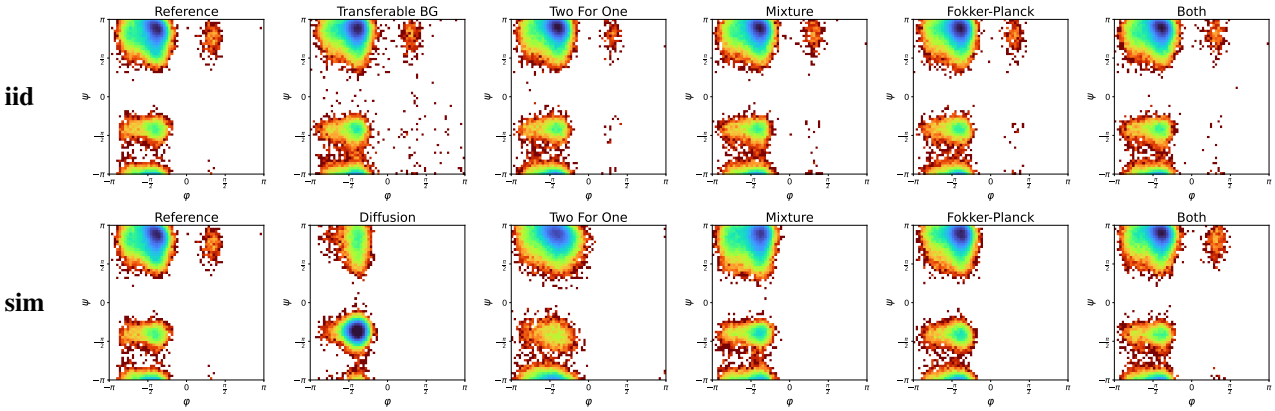


Figure 20: **RV:** We compare the free energy plot on the dihedral angles $\varphi, \psi$ for all presented methods for iid sampling and Langevin simulation.

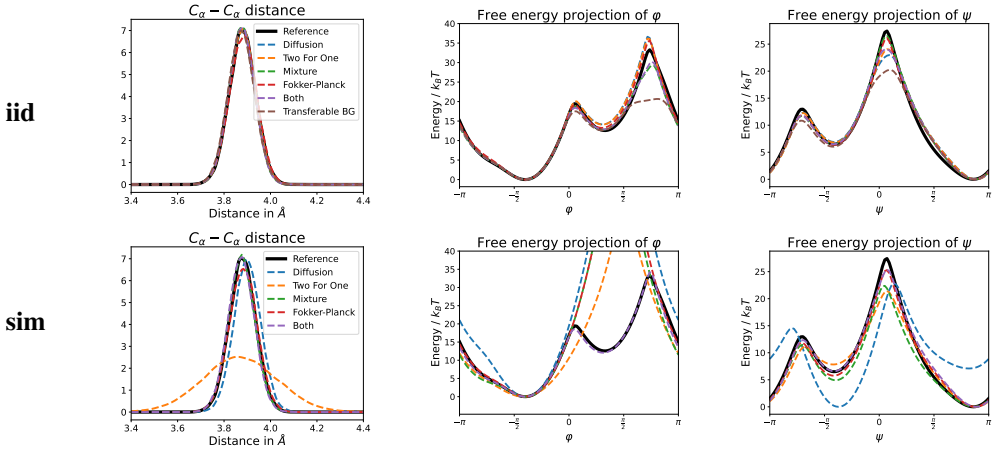


Figure 21: **RV:** We compare further metrics between iid sampling and Langevin simulation. We compare the $C_\alpha$–$C_\alpha$ distance for the dipeptides and also the free energy projections along the dihedral angles $\varphi, \psi$.