

---

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

---

Michael Plainer<sup>†1,2,3,4</sup>

Hao Wu<sup>†5</sup>

Leon Klein<sup>1</sup>

Stephan Günnemann<sup>6</sup>

Frank Noé<sup>†1,7,8</sup>

<sup>1</sup>Freie Universität Berlin   <sup>2</sup>Zuse School ELIZA   <sup>3</sup>Technische Universität Berlin  
<sup>4</sup>Berlin Institute for the Foundations of Learning and Data   <sup>5</sup>Shanghai Jiao Tong University  
<sup>6</sup>Technische Universität München   <sup>7</sup>Rice University   <sup>8</sup>Microsoft Research AI4Science

## Abstract

In recent years, diffusion models trained on equilibrium molecular distributions have proven effective for sampling biomolecules. Beyond direct sampling, the score of such a model can also be used to derive the forces that act on molecular systems. However, while classical diffusion sampling usually recovers the training distribution, the corresponding energy-based interpretation of the learned score is often inconsistent with this distribution, even for low-dimensional toy systems. We trace this inconsistency to inaccuracies of the learned score at very small diffusion timesteps, where the model must capture the correct evolution of the data distribution. In this regime, diffusion models fail to satisfy the Fokker–Planck equation, which governs the evolution of the score. We interpret this deviation as one source of the observed inconsistencies and propose an energy-based diffusion model with a Fokker–Planck-derived regularization term to enforce consistency. We demonstrate our approach by sampling and simulating multiple biomolecular systems, including fast-folding proteins, and by introducing a state-of-the-art transferable Boltzmann emulator for dipeptides that supports simulation and achieves improved consistency and efficient sampling. Our code, model weights, and self-contained JAX and PyTorch notebooks are available at <https://github.com/noegroup/ScoreMD>.

## 1 Introduction

Understanding biochemical systems requires modeling not only static molecular structures but also their temporal evolution and interactions. Molecular dynamics (MD) simulations offer a principled framework to obtain such information from atomistic force fields and, with recent methodological and computational advances, can reach biologically relevant timescales (Lindorff-Larsen et al., 2011; Wolf et al., 2020). Nonetheless, fully atomistic simulations of large systems or slow conformational transitions remain computationally challenging (Kmieciak et al., 2016; Plattner et al., 2017). Coarse-graining (CG) methods address this limitation by grouping atoms into beads to accelerate simulations at the cost of physical resolution. In this reduced space, interactions and forces cannot be described accurately with traditional MD methods, requiring learning-based approaches to approximate forces and recover correct dynamics (Clementi, 2008; Noid, 2013; Husic et al., 2020; Charron et al., 2025).

Diffusion models (Ho et al., 2020; Song et al., 2021) have emerged as powerful generative tools capable of learning high-dimensional molecular distributions (Abramson et al., 2024; Watson et al., 2023; Corso et al., 2023; Plainer et al., 2023b; Lewis et al., 2025). They are trained to reverse

---

<sup>†</sup>Corresponding Authors:

michael.plainer@fu-berlin.de, hwu81@sjtu.edu.cn, franknoe@microsoft.com



Figure 1: Training diffusion models on a 2D toy example reveals inconsistencies. While classical iid diffusion sampling (i.e., denoising) correctly reproduces both modes, evaluating the score at  $t = 0$  to estimate the unnormalized density yields a third mode and an incorrect mass distribution. Such a diffusion model would produce incorrect dynamics while producing correct samples. A model trained with Fokker–Planck regularization is self-consistent, and aligns the learned score at  $t = 0$  with the distribution recovered by diffusion sampling.

a stochastic diffusion process in which data is progressively perturbed with Gaussian noise over diffusion time  $t$ , starting from clean samples at  $t = 0$  and approaching pure noise at  $t = 1$ . Generation then proceeds in reverse: random noise is iteratively denoised using the learned time-dependent *score* function  $\nabla_x \log p(x, t)$ , producing molecular configurations that approximate the equilibrium training distribution. While such models can efficiently generate independent equilibrium structures (Lewis et al., 2025), they lack temporal structure and cannot capture kinetic properties (Wang and Hou, 2011a,b) or transition mechanisms (Henkelman et al., 2000; Bolhuis et al., 2002). In contrast, MD yields both, thermodynamic and kinetic properties but has a high computational cost.

In principle, the score function at  $t = 0$  of a diffusion model trained on Boltzmann-distributed molecular samples can serve as a surrogate for the physical forces. In an ideal energy-based diffusion model, this correspondence would ensure thermodynamic consistency between the learned energy landscape and the equilibrium distribution. In practice, however, diffusion models rarely behave as proper energy-based models, especially as system complexity increases (Arts et al., 2023). In this work, we address these limitations and propose a consistent energy-based diffusion model that can generate realistic equilibrium samples while providing physically consistent forces for simulation.

To address this, we leverage the Fokker–Planck equation, which governs the evolution of probability densities under diffusion processes (Särkkä and Solin, 2019), but is generally violated by existing diffusion models (Lai et al., 2023). By introducing a loss term during training that penalizes deviations from this equation, we improve model alignment and empirically enhance consistency between generated samples and simulated dynamics, as shown in Figure 1. In practice, we parameterize the score as the gradient of an energy function, providing access to a potential energy and conservative forces, which is the prerequisite to enable the use of established physical sampling methods (Torrie and Valleau, 1977; Laio and Parrinello, 2002) with coarse-grained diffusion models. Furthermore, by decomposing the diffusion timeline into smaller intervals and assigning a separate model to each, this regularization can be selectively applied to small diffusion timesteps (i.e., where it is needed most).

Our main contributions in this work are as follows:

1. We include a Fokker–Planck-based regularization when training energy-based diffusion models, which substantially improves simulation stability and enforces consistency between diffusion sampling and molecular dynamics simulations recovered from the learned forces. This enables accurate sampling and simulation of fast-folding proteins such as Chignolin and BBA.
2. We show that training on a small sub-interval of the diffusion time suffices for stable simulation. Combining this with smaller models trained on complementary intervals can yield more efficient training and inference while preserving high sampling quality.
3. We develop a state-of-the-art transferable Boltzmann emulator for dipeptides capable of fast high-quality independent sampling, and consistent long-time simulation.

## 2 Background

### 2.1 Generative Score-based Modeling

**Diffusion models** (Song and 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 defined by a stochastic differential equation (SDE)

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t) dt + g(t) d\mathbf{w}, \quad (1)$$

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

$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})] dt + g(t) d\bar{\mathbf{w}}. \quad (2)$$

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

As for  $\mathbf{f}$  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) (compare Appendix A.1).

**Denosing score matching** (Vincent, 2011; Song et al., 2021) is a common way to train diffusion models by minimizing the squared error between a time-dependent learned score function  $\mathbf{s}^\theta(\mathbf{x}(t), t)$  and the true score of the transition kernel  $p_{0t}(\mathbf{x}(t) | \mathbf{x}(0))$  conditioned on the training data  $\mathbf{x}(0)$ :

$$\theta^* = \operatorname{argmin}_{\theta} \mathbb{E}_{t \sim \mathcal{U}(0,1)} \mathbb{E}_{\mathbf{x}(0)} \mathbb{E}_{\mathbf{x}(t) | \mathbf{x}(0)} \left[ \lambda(t) \left\| \mathbf{s}^\theta(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t) | \mathbf{x}(0)) \right\|_2^2 \right], \quad (3)$$

where  $\lambda(t)$  is a time-dependent weighting function. For brevity, we will denote the denosing diffusion loss as  $\mathcal{L}_{\text{DSM}}[\mathbf{s}^\theta](\mathbf{x}, t) = \lambda(t) \left\| \mathbf{s}^\theta(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t) | \mathbf{x}(0)) \right\|_2^2$ .

**Parameterization and instabilities.** With an affine drift  $\mathbf{f}$ , we can write the closed-form solution of  $p_{0t}$  as a Gaussian (Särkkä and Solin, 2019), and can efficiently train diffusion models such that

$$\theta^* = \operatorname{argmin}_{\theta} \mathbb{E}_{t \sim \mathcal{U}(0,1)} \mathbb{E}_{\mathbf{x}(0)} \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} \left[ \lambda(t) \left\| \mathbf{s}^\theta(\mu(\mathbf{x}(0), t) + \sigma(t)\epsilon, t) + \frac{\epsilon}{\sigma(t)} \right\|_2^2 \right], \quad (4)$$

where  $\mu(\mathbf{x}(0), t)$ ,  $\sigma(t)$  depend on the concrete choices for  $\mathbf{f}$  and  $g$ . By construction,  $\sigma(0) = 0$ , which ensures correct interpolation between data and noise. Minimizing the denosing loss yields a neural network with parameters  $\theta$  that can approximate the unconditional score  $\mathbf{s}^\theta(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ .

As  $t \rightarrow 0$ , 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

**Molecular dynamics simulations** numerically integrate Newton’s equations of motion to describe the time evolution of a molecular system. A widely used formulation is Langevin dynamics (Leimkuhler and Matthews, 2015), which corresponds to integrating the following set of SDEs

$$d\mathbf{x} = \mathbf{v} dt, \quad \mathbf{M} d\mathbf{v} = -\nabla_{\mathbf{x}} U(\mathbf{x}) dt - \gamma \mathbf{M} \mathbf{v} dt + \sqrt{2\gamma k_B T} d\mathbf{w}_t. \quad (5)$$

$\mathbf{M}$  denotes the particle masses,  $\mathbf{v}$  the velocities,  $\gamma$  is a friction constant,  $k_B T$  a constant, and  $\mathbf{w}_t$  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  $-\nabla_{\mathbf{x}} U$ . 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  $\mathbf{s}^\theta(\mathbf{x}, t = 0)$  for this purpose, as we describe next.

**Extracting forces.** After a long, equilibrated MD simulation, samples follow the *Boltzmann distribution* (Boltzmann, 1868) such that  $p(\mathbf{x}) = \exp(-\frac{U(\mathbf{x})}{k_B T})/Z$ , where  $U$  is the potential energy and  $Z$  the normalization constant. Training a diffusion model on data from such a distribution yields at  $t = 0$

$$\begin{aligned} \mathbf{s}^\theta(\mathbf{x}, t = 0) &\approx \nabla_{\mathbf{x}} \log p_{t=0}(\mathbf{x}) \\ &= \nabla_{\mathbf{x}} \log \exp\left(-\frac{U(\mathbf{x})}{k_B T}\right) - \nabla_{\mathbf{x}} \log Z \\ &= -\nabla_{\mathbf{x}} \frac{U(\mathbf{x})}{k_B T} - 0. \end{aligned} \quad (6)$$

Hence, the score at diffusion time  $t = 0$  is proportional to  $-\nabla_{\mathbf{x}}U(\mathbf{x})$ , the forces acting on the system. This equivalence implies that any diffusion model trained on Boltzmann-distributed data can not only be used for independent sampling but also for molecular simulation by using the learned score as a force estimator with the SDEs of Equation (5). Unlike previous methods that rely on explicit force labels for training (Husic et al., 2020; Durumeric et al., 2023; Charron et al., 2025; Krämer et al., 2023), this relation allows us to learn a model directly from equilibrium samples. This is particularly useful for systems where potential energy and force labels are unavailable, as in CG modeling, which is the setting we consider. The central question we will answer next is whether and how this identity can be realized in practice and if the resulting model is sufficiently accurate.

### 3 Method

Diffusion models trained on Boltzmann-distributed molecular data provide a bridge between generative sampling and physical energy landscapes. At the diffusion endpoint  $t = 0$ , the score represents the forces acting on the system (Equation (6)), suggesting that a single model can capture the sampling process and the underlying potential energy. These two perspectives correspond to reconstructing the data distribution through denoising,  $p(\mathbf{x})$ , or evaluating the energy-based form at  $t = 0$ ,  $p_0(\mathbf{x})$ :

$$\begin{aligned} p(\mathbf{x}) &\sim \text{Denoising}[\mathbf{s}^\theta](\mathbf{x}, t) \\ p_0(\mathbf{x}) &\sim \exp\left(-\frac{U(\mathbf{x})}{k_B T}\right) \quad \text{with } \mathbf{s}^\theta(\mathbf{x}, 0) = -\nabla_{\mathbf{x}} \frac{U(\mathbf{x})}{k_B T}. \end{aligned} \quad (7)$$

Ideally, these two formulations should agree such that  $p(\mathbf{x}) = p_0(\mathbf{x})$ . In practice, however, diffusion models often display inconsistencies (Koehler et al., 2023; Li et al., 2023; Bortoli et al., 2024). Prior attempts to use the score–force relation beyond static sampling can reproduce realistic ensembles under classical diffusion sampling  $p(\mathbf{x})$  but fail to recover these distributions in simulation  $p_0(\mathbf{x})$  (Arts et al., 2023), indicating that the learned model does not behave as a proper energy-based model.

We show that consistency can be recovered directly from unlabeled equilibrium data by enforcing the Fokker–Planck equation. This equation links energy functions across different diffusion times, effectively transferring accuracy from stable, large- $t$  regions to the small- $t$  regime. The result is a self-consistent diffusion model that learns physically meaningful forces without ever observing them.

#### 3.1 Improving Consistency with the Fokker–Planck Equation

The Fokker–Planck equation (Øksendal, 2003; Särkkä and Solin, 2019) is a partial differential equation that describes how probability densities evolve 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., 2025) can be written as

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

where  $\text{div}_{\mathbf{x}}$  denotes the divergence operator  $\text{div}_{\mathbf{x}} \mathbf{F} = \text{tr}(\partial_{\mathbf{x}} \mathbf{F})$ . This equation is a fundamental property of the diffusion process, and any well-trained model should satisfy Equation (8). However, as shown in Section 5, we observe that diffusion models violate this equation, particularly at small  $t$ , which is consistent with findings in prior work (Lai et al., 2023). Since the Fokker–Planck equation links the evolution of the score to that of the underlying density, such violations imply that the learned score does not evolve consistently with the density over diffusion time. Consequently, small deviations can accumulate, possibly leading to inconsistencies between the distribution  $p(\mathbf{x})$  recovered through denoising and the instantaneous density  $p_0(\mathbf{x})$  we want to recover.

**Fokker–Planck regularization.** To correct this, we introduce a regularization term that enforces compliance with Equation (8) during training, similar to Lai et al. (2023). Alongside the standard diffusion objective, we minimize the residual error

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

and define the corresponding loss as  $\mathcal{L}_{\text{FP}}[\log p^\theta](\mathbf{x}, t) = \lambda_{\text{FP}}(t) D^{-2} \|R(\mathbf{x}, t)\|_2^2$ , where  $\mathbf{x} \in \mathbb{R}^D$  and  $\lambda_{\text{FP}}$  is a time-dependent weighting function, which we set to be the same as  $\lambda$ . As the regularization can be evaluated for all  $\mathbf{x}, t$ , we combine it with the score-matching objective

$$\underset{\theta}{\text{argmin}} \mathbb{E}_{t \sim \mathcal{U}(0,1)} \mathbb{E}_{\mathbf{x}(0)} \mathbb{E}_{\mathbf{x}(t) | \mathbf{x}(0)} \left[ \mathcal{L}_{\text{DSM}}[\nabla_{\mathbf{x}} \log p^\theta](\mathbf{x}(t), t) + \alpha \cdot \mathcal{L}_{\text{FP}}[\log p^\theta](\mathbf{x}(t), t) \right], \quad (10)$$

where  $\alpha$  is a hyperparameter that determines the regularization strength. The regularization  $\mathcal{L}_{\text{FP}}$  term provides an additional training signal that stabilizes learning in the small-time regime, where the denoising loss becomes ill-conditioned and gradients are dominated by noise. By explicitly linking the temporal evolution of densities, the regularization encourages the model to remain consistent with the Fokker–Planck dynamics throughout diffusion time.

**Weak residual formulation.** The exact computation of the residual  $R$  (and thus the loss  $\mathcal{L}_{\text{FP}}$ ) involves costly higher-order derivatives, especially the divergence term  $\text{div}_{\mathbf{x}}(\nabla_{\mathbf{x}} \log p^{\theta})$  can be challenging to compute for high-dimensional data. To reduce this overhead, we introduce a series of approximations and begin by using a residual in the weak formulation similar to the one used by [Guo et al. \(2022\)](#)

$$\tilde{R}(\mathbf{x}, t) = E_{\mathbf{v}}[R(\mathbf{x} + \mathbf{v}, t)] \quad (11)$$

with  $\mathbf{v} \sim \mathcal{N}(0, \sigma^2 I)$  and a small  $\sigma > 0$ . As  $\sigma$  approaches 0,  $\tilde{R}(\mathbf{x}, t)$  will be equal to  $R(\mathbf{x}, t)$ . Using the weak residual formulation,  $\tilde{R}(\mathbf{x}, t)$  can be estimated by the following unbiased estimator, which only requires the computation of first-order derivatives (see [Appendix A.2](#) for derivation)

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

where  $\mathbf{s}^{\theta} = \nabla_{\mathbf{x}} \log p^{\theta}$ . This yields an unbiased estimator of the squared residual

$$\mathcal{L}_{\text{FP}}(\mathbf{x}, t) \approx \left\| \tilde{R}(\mathbf{x}, t) \right\|_2^2 \approx \tilde{R}(\mathbf{x}, t; \mathbf{v}) \cdot \tilde{R}(\mathbf{x}, t; \mathbf{v}'), \quad (13)$$

with  $\mathbf{v}, \mathbf{v}' \sim \mathcal{N}(0, \sigma^2 I)$ , yielding a computationally manageable estimate of  $\mathcal{L}_{\text{FP}}$ .

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

$$\partial_t \log p_t^{\theta} \approx \frac{h_s^2 \log p_t^{\theta}(\mathbf{x}, t + h_d) + (h_d^2 - h_s^2) \log p_t^{\theta}(\mathbf{x}, t) - h_d^2 \log p_t^{\theta}(\mathbf{x}, t - h_s)}{h_s h_d (h_s + h_d)}. \quad (14)$$

Since both  $t$  and  $\log p_t^{\theta}$  are one-dimensional, finite differences serve as a robust estimate. This allows for computationally feasible approximation of the loss  $\mathcal{L}_{\text{FP}}$ , using multiple forward passes.

### 3.2 Physically Consistent Model Design

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

**Conservative model parameterization.** In diffusion models, it is common to parameterize the score directly as  $\mathbf{s}^{\theta} = \nabla_{\mathbf{x}} \log p_t^{\theta} = \text{NET}(\mathbf{x}, t)$  rather than defining an explicit energy function and taking its gradient,  $\mathbf{s}^{\theta} = \nabla_{\mathbf{x}} \text{NET}'(\mathbf{x}, t)$ . While energy-based formulations have been explored previously for diffusion models ([Song and Ermon, 2019](#)), they are less common in practice ([Du et al., 2023](#)), since most applications require only the score and report no difference in sampling quality ([Salimans and Ho, 2021](#)). In our setting, however, we explicitly aim to construct a consistent energy-based diffusion model, ensuring the correspondence between the learned score and the underlying energy landscape as formalized in [Equation \(7\)](#). This gradient-based formulation provides access to the energy  $\log p_t$ , which is essential for enforcing physical consistency through the Fokker–Planck equation. Moreover, for molecular simulations, a conservative formulation, where the forces are derived from a well-defined energy, is critical for numerical stability and accurate force estimation ([Schütt et al., 2017](#); [Batzner et al., 2022](#); [Arts et al., 2023](#)), as further demonstrated in [Appendix C.1](#).

**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 work ([Abramson et al., 2024](#)) has shown that the architecture itself does not need to enforce this property. Hence, we apply random rotations during training so that the network learns rotational equivariance via data augmentation.

The main part of the architecture can be summarized by describing the nodes  $\mathbf{n}$  and edges  $e$  such that

$$e_{ij} = \mathbf{x}_i - \mathbf{x}_j, \quad \mathbf{n}_i^{(0)} = [\mathbf{a}_i, t], \quad \mathbf{n}^{(l+1)} = \phi^{(l)}(\mathbf{n}^{(l)}, e), \quad (15)$$

where  $\mathbf{x}$  are the coarse-grained positions,  $\mathbf{e}$  are the edge features,  $\mathbf{a}$  are atom features,  $t$  is the diffusion time,  $\mathbf{n}^{(l)}$  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 the atom identity, atom number, residue index, and amino acid type, following Klein and Noé (2024).

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

### 3.3 Mixture of Experts

Classical diffusion models are trained across the full diffusion timeline to enable generative sampling. In some applications, learning the score only at  $t = 0$  can be sufficient, as it gives access to the potential. To support this, we introduce a time-based mixture of experts (MoE) approach, inspired by ideas from the image domain to improve efficiency (Balaji et al., 2023; Ganjdanesh et al., 2025). Rather than training a single model across the entire range  $t \in (0, 1)$ , we partition the interval into disjoint subintervals  $\mathcal{I}_0, \mathcal{I}_1, \dots$  with  $\bigcup_i \mathcal{I}_i = (0, 1)$ , and assign a separate expert  $s_i^\theta$  to each interval. The overall score can then be written by evaluating the correct model

$$s^\theta(\mathbf{x}, t) = s_i^\theta(\mathbf{x}, t), \quad \text{for } t \in \mathcal{I}_i. \quad (16)$$

Thus, only one expert is active at any given  $t$ , which simplifies memory management and allows independent and parallel training of all experts. For simulation, only the expert corresponding to  $t = 0 \in \mathcal{I}_i$  is needed, while iterative diffusion sampling sequentially loads every expert as  $t$  decreases.

This design provides several advantages. For simulation, training only on the relevant small- $t$  range (e.g.,  $(0, 0.1)$ ) avoids wasting resources on unnecessary diffusion times that are never used. For generative sampling, dividing the timeline still offers clear benefits. Applying the Fokker–Planck loss from Equation (10) across all diffusion times can lead to overregularization at large  $t$ , degrading iid sampling performance. Since large- $t$  models do not require Fokker–Planck regularization or a conservative parameterization, using simpler unconstrained models improves generative quality and prevents unnecessary constraints. Moreover, the experts specialize in distinct temporal regions: small- $t$  experts capture fine structural details, while large- $t$  experts focus on coarse features (Ganjdanesh et al., 2025). This specialization stabilizes training and enhances the consistency between sampling and simulation beyond what could be achieved by simply increasing model capacity (see Appendix C.2). Finally, because only one expert is loaded into memory at a time, we can overall train more parameters without exceeding GPU limits. We will explore these benefits in Section 5.

## 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 the learned score. 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. Raja et al. (2025) also use the score–force relation but for transition path sampling and could benefit from our approach to have more accurate forces. Daigavane et al. (2025) use the inherent noise of the diffusion process to converge to an equilibrium distribution faster, but they do not learn accurate scores either. Several works instead learn coarse-grained force fields via a force-matching objective (Husic et al., 2020; Köhler et al., 2023; Charron et al., 2025; Durumeric et al., 2024). Rather than training a model to represent the data distribution, these methods approximate the target forces. However, they typically need system-specific energy priors, relying 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; Abidin and Kim, 2024; Kim et al., 2024; Wu and 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 directly 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. Bortoli et al. (2024) propose to improve the score at small diffusion timesteps, when the probability distribution of the samples is known. However, this is not directly compatible with coarse-graining, as the potential and the derived forces are not known. Similarly to us, Lai et al. (2023) also proposed a Fokker–Planck-inspired regularization; however, their method applies a higher-order regularization to the score itself to improve iid sample quality, rather than enforcing consistency through the potential to improve the learned score. Daras et al. (2023) propose a regularization to ensure average consistency of the score, whereas our objective aims to ensure consistency of the energy over complete trajectories through the Fokker–Planck equation. Relatedly, Hu et al. (2025) propose a score-based solver for high-dimensional Fokker–Planck equations, focusing on solving general SDE forward problems, and Du et al. (2024) use the Fokker–Planck equation to describe MD as a series of Gaussians that can be integrated but do not recover the full unconditional Boltzmann distribution.

## 5 Experiments

In this section, we evaluate the consistency of diffusion models by comparing samples obtained through classical denoising (*iid*) with those generated by sequential Langevin simulations (*sim*) using forces derived with the score–force relation from Equation (6). In other words, we verify the alignment of Equation (7) for energy-based models and show superior performance. We demonstrate our approach (*ScoreMD*) on three biomolecular systems—alanine dipeptide, Chignolin, and BBA—and introduce a transferable model that generalizes across dipeptides, improving over existing state-of-the-art Boltzmann generators (Klein and Noé, 2024). The code, model weights, and self-contained notebooks in JAX and PyTorch are publicly available at <https://github.com/noegroup/ScoreMD>.

**Metrics.** As the main metric of interest, we compare the 2D free energy surfaces of the equilibrium distributions obtained by different methods. For dipeptides, we project the data onto the dihedral angles  $\varphi$  and  $\psi$ , while for proteins we perform time-lagged independent component analysis (TICA) (Pérez-Hernández et al., 2013) on bond distances and dihedral angles to recover two representative coordinates. To quantify differences between free energy surfaces, we report the potential of mean force (PMF) error (Durumeric et al., 2024), which measures the squared distance between the negative logarithms of the sampled and reference densities in the projected space. This metric places higher weight on low-density regions compared to alternatives such as the Jensen-Shannon (JS) divergence. Additional details are provided in Appendix B.2.

**Baselines.** We train a conservative *Diffusion* model and re-implement the *Two For One* method (Arts et al., 2023) within the same continuous-time diffusion framework to ensure a fair comparison. Both models use identical architectures and training procedures and hence produce the same iid samples (up to numerical errors); their only difference lies in the diffusion time used during evaluation. Specifically, *Two For One* performs simulation at a finite diffusion time  $t \gg 0$  to enhance stability, whereas *Diffusion* is evaluated at  $t = 10^{-5}$ . For transferability, we re-train the *Transferable Boltzmann generator (BG)* model (Klein and Noé, 2024) with coarse-graining and use it without reweighting.

### 5.1 Alanine Dipeptide and Fast-Folding Proteins

**Dataset.** For alanine dipeptide, we use 50k samples from an MD simulation in implicit solvent (Köhler et al., 2021), coarse-grained to five atoms: [C, N, CA, C, N]. For Chignolin and BBA, we use the dataset from Lindorff-Larsen et al. (2011), coarse-grained to one bead per amino acid (10 and 28 residues, respectively), and use 80% of the samples for training. For the *iid* setting, we generate the same number of samples as in the training set. For *sim*, we initialize 100 parallel simulations from random training conformations, and simulate with a timestep of 2 fs for a total length of 1.2  $\mu$ s for alanine dipeptide, 855.6 ns for Chignolin, and 200 ns for BBA. All trajectories are downsampled to match the number of training samples for consistency with the *iid* setting.

**(In-)consistent sampling.** Figure 2 compares the free energy landscapes in the projected space for *iid* sampling and Langevin simulation (*sim*). While all methods reproduce the training distribution under *iid* sampling, simulation quality varies and reveals inconsistencies. For the small system alanine dipeptide, the equilibrium distributions appear similar across methods. However, as system size

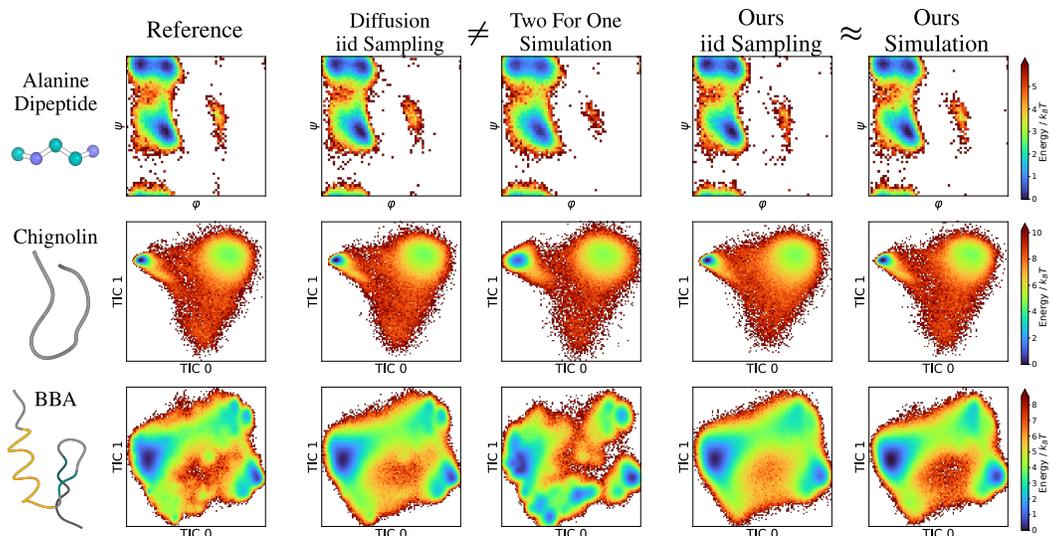


Figure 2: Comparison of equilibrium distributions obtained by *iid* sampling and Langevin simulation (*sim*) across different systems and methods. While classical *iid* sampling recovers the reference equilibrium distribution, performing simulation with the learned score reveals inconsistencies when models are not trained with Fokker-Planck regularization, i.e.,  $p(\mathbf{x}) \neq p_0(\mathbf{x})$ . Regularized models achieve consistent behavior across systems.

Method		ALDP PMF ( $\downarrow$ )	Chignolin PMF ( $\downarrow$ )	BBA PMF ( $\downarrow$ )
Diffusion	<i>iid</i>	$0.098 \pm 0.006$	$0.027 \pm 0.000$	$0.034 \pm 0.000$
Ours	<i>iid</i>	$0.097 \pm 0.008$	$0.035 \pm 0.001$	$0.234 \pm 0.001$
Two For One	<i>sim</i>	$0.206 \pm 0.004$	$1.438 \pm 0.019$	$1.624 \pm 0.107$
Ours	<i>sim</i>	$0.091 \pm 0.004$	$0.038 \pm 0.006$	$0.254 \pm 0.005$

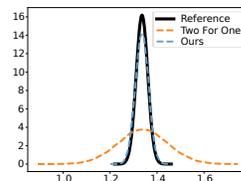


Table 1: Quantitative comparison of sampling and simulation across different methods for alanine dipeptide (ALDP), Chignolin, and BBA. *Ours* yields significantly more consistent results between *iid* and *sim*.

Figure 3: Density of C-N bond length in Å for simulation of alanine dipeptide using different models.

increases, artifacts become more pronounced: for Chignolin, the *Two For One* simulation exhibits strong noise, and for BBA, the system separates into unphysical modes, yielding an incorrect equilibrium distribution. In contrast, our Fokker-Planck regularization enforces consistency, producing matching ensembles for *iid* sampling and Langevin simulation (*sim*), i.e.,  $p(\mathbf{x}) \approx p_0(\mathbf{x})$ .

These trends can be seen in Table 1, which shows that *Ours* substantially improves simulation quality. We also observe a modest reduction in *iid* sampling performance, reflecting a tradeoff between generative fidelity and consistency, governed by the regularization strength  $\alpha$ . Larger values promote stronger alignment between sampling and simulation but may slightly degrade *iid* sampling accuracy.

**Noisy simulation.** *Two For One* improves consistency by evaluating the model at a larger diffusion time  $t \gg 0$  during simulation, which introduces substantial noise that degrades structural fidelity. While *iid* sampling remains unaffected, simulations exhibit pronounced deviations, as seen for Chignolin in Figure 2. For alanine dipeptide, the free energy surface obtained with *Two For One* appears reasonable, but because the method relies on “noisy” forces, the observables are also affected by this noise. For instance, bond-length distributions will be incorrect, as depicted in Figure 3.

**Dynamics.** A key advantage of a *consistent* energy-based diffusion model is that it provides access not only to generative diffusion sampling but also to a physically meaningful energy function usable for downstream tasks such as MD simulation. As such, we can recover kinetic information and temporal behavior, as studied in Figure 4 for BBA, which is not accessible with classical diffusion sampling. Additional analysis of the dynamics, such as transition probabilities between states and the corresponding results for Chignolin, is provided in Appendix C.6.4.

**Force matching.** CG models are typically trained using force matching rather than diffusion-based objectives. In Appendix C.5.4, we benchmark our method against recent force-matching approaches for alanine dipeptide. Although these baselines often rely on explicit physical priors and force labels, our diffusion-based model achieves ensembles that more closely match the reference distribution.

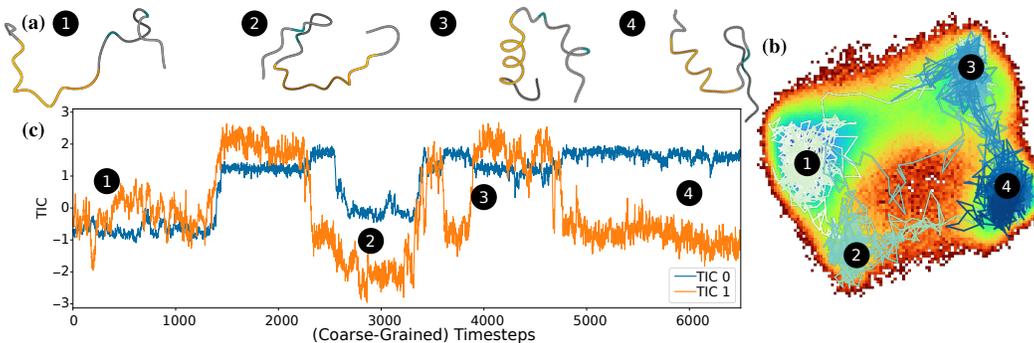


Figure 4: Recovering and analyzing the dynamic behavior of BBA. (a) 3D molecular structures from a single MD trajectory generated by our model. (b) Trajectory projected onto the first two TIC coordinates. (c) Illustrating the TIC coordinates over the coarse-grained (and subsampled) timesteps.

## 5.2 Transferability Across Dipeptides (Two Amino Acids)

**Dataset.** We use the dataset introduced by Klein et al. (2023a), which contains 49k samples from implicit-solvent simulations of  $1 \mu\text{s}$  covering all possible combinations of the 20 standard amino acids. From these, we use 175 dipeptides for training. Each dipeptide is coarse-grained by retaining the atoms [N, CA, CB, C, O] for each amino acid, a common CG resolution (Charron et al., 2025). With this, up to 10 atoms are retained per molecule, as seen in Figure 5 (a). For *iid* sampling, we draw 49k independent samples, while for *sim* we perform 30 ns of Langevin simulation per dipeptide, initialized from 10 random conformations with a timestep of 0.5 fs downsampled to 49k frames.

**Models.** We ablate the individual contributions of the Fokker–Planck regularization, the impact of MoE, and the effect of evaluating models at larger diffusion timesteps  $t \gg 0$ . Similar ablation studies for previous results are provided in Appendices C.5 and C.6. Specifically, we compare the *Diffusion* and *Two For One* baselines with three variants: *Mixture* denotes the MoE scheme, consisting of three experts trained on the intervals  $(0, 0.1)$ ,  $[0.1, 0.6)$ , and  $[0.6, 1.0)$ . The models were robust to moderate changes in these ranges, provided the smallest-time expert was trained on a sufficiently large subinterval. All three experts are combined for *iid* sampling, while only the smallest-timescale model is used for simulation. The experts corresponding to larger timescales are reduced in size and complexity. *Fokker–Planck* refers to a single model trained with the loss from Equation (10). *Both* combines these two approaches, where the regularization is applied only to the smallest-timescale expert within the MoE setup. This corresponds to the model referred to previously as *Ours*.

**Overdispersion.** As shown in Figure 5 (b), all models can generate independent samples that closely match the reference distribution. However, samples from the *Transferable BG* model exhibit increased noise and over-explore low-probability states. In the original work, this behavior is corrected through sample reweighting, which we omit here and instead treat the model simply as a Boltzmann emulator. While the resulting differences are within standard deviations, the model yields a higher mean with comparable variance (see Table 2). Moreover, note that this model does not support simulation. When using the score for simulation, *Two For One* produces broader, overdispersed distributions, as shown in Figure 5 (b). This overdispersion also propagates to structural features such as inter-atomic distances, consistent with the behavior observed in previous systems (see Appendix C.8.2). More metrics and evaluation for more dipeptides can be found in Appendix C.8.

**Advantages of mixture.** The MoE architecture can significantly improve simulation quality compared to *Diffusion*. For the specific dipeptide shown in Figure 5 (c), *Both* further enhances performance over *Fokker–Planck*, improving both *iid* and simulated distributions. While the degree of improvement varies across dipeptides—with some cases showing comparable results (see Appendix C.8.3)—MoE generally increases simulation stability, as reflected in the lower PMF errors reported in Table 2.

Since *Mixture* and *Both* contain more parameters (see Appendix B.5), part of the improvement can be attributed to increased capacity. However, MoE also offers clear computational advantages: by applying regularization only at small diffusion times and using smaller, unconstrained models for larger timescales, it reduces training and inference cost, cutting sampling time by more than 50% in this case (see Appendix C.3) and prevents overregularization at large diffusion timesteps. Moreover, additional ablation studies in Appendix C.2 show that the benefits of MoE extend beyond parameter scaling, demonstrating that training on smaller temporal subranges improves robustness and that independently trained experts can be seamlessly combined into a coherent model.

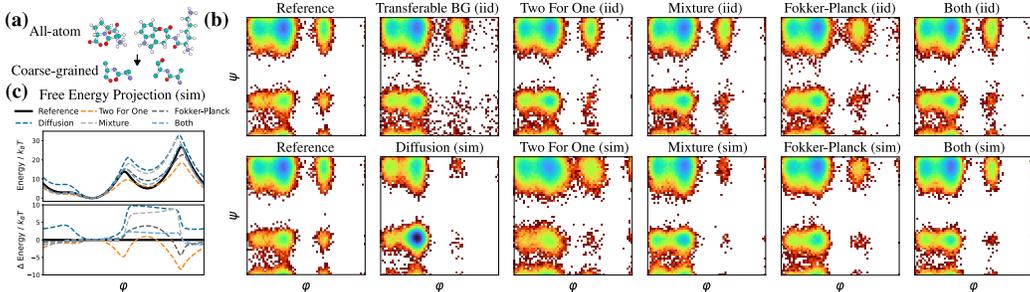


Figure 5: Comparison of methods on testset dipeptide KS. (a) The coarse-graining scheme. (b) Comparison of the Ramachandran plots of different methods for iid sampling and Langevin simulation. (c) The projection of the free energy surface and differences along the dihedral angle  $\varphi$  for samples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

Method	iid PMF ( $\downarrow$ )	sim. PMF ( $\downarrow$ )
Transferable BG	<b>0.230</b> $\pm$ <b>0.119</b>	-
Diffusion	<b>0.206</b> $\pm$ <b>0.159</b>	6.515 $\pm$ 3.175
Two For One	<b>0.203</b> $\pm$ <b>0.149</b>	0.741 $\pm$ 0.319
Mixture	<b>0.200</b> $\pm$ <b>0.127</b>	0.658 $\pm$ 0.407
Fokker-Planck	<b>0.241</b> $\pm$ <b>0.105</b>	<b>0.368</b> $\pm$ <b>0.267</b>
Both	<b>0.199</b> $\pm$ <b>0.127</b>	<b>0.203</b> $\pm$ <b>0.104</b>

Table 2: Comparison of metrics across all testset dipeptides with PMF error. To compute the mean and standard deviation, we have averaged the results across all evaluated dipeptides. Lower values are better.

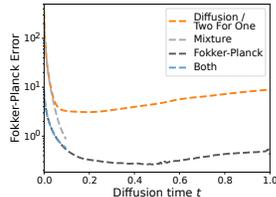


Figure 6: Comparing the Fokker-Planck error for  $\log p^\theta$  of multiple models.

**Fokker-Planck error.** Figure 6 shows the deviation from the Fokker-Planck equation, quantified as  $\|\mathcal{F}_{p^\theta}(\mathbf{x}, t) - \partial_t \log p_t^\theta(\mathbf{x})\|_2$ , 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 energy-based. Across all methods, the error is largest 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 at  $t = 0$  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 energy-based diffusion models and analyzed the discrepancy between the density recovered through denoising and the learned energy at diffusion time  $t = 0$ . We showed that diffusion models are generally inconsistent, and these two densities do not coincide. To address this, we introduced a Fokker-Planck-based regularization on the model’s energy, demonstrating that reducing deviations from the Fokker-Planck equation substantially improves model consistency. Furthermore, we found that focusing training on smaller diffusion times further enhances simulation quality. With these improvements, we obtain a physically consistent energy-based model that can generate independent equilibrium samples and recover realistic molecular dynamics from the same learned potential, providing access to both static and kinetic properties. This is particularly useful in coarse-grained settings where direct force information is not available. We validated these findings across multiple systems, trained on fast-folding proteins, and generalized across dipeptides.

Despite the theoretical motivation behind our approach, the results presented are primarily empirical. While our findings indicate that reducing the Fokker-Planck deviation improves consistency, this is unlikely to be the only source of simulation error. In fact, due to the fundamental differences between diffusion sampling and Langevin simulation, perfect alignment may not be achievable without limiting model expressivity. Moreover, evaluating the Fokker-Planck residual introduces computational overhead, which we mitigate through a weak residual formulation, though it still requires multiple forward passes during training and thus remains computationally demanding. Future work could extend this framework to larger molecular systems and explore transfer learning across biomolecular families. Another promising direction is to fine-tune existing diffusion models with the proposed regularization to explicitly correct Fokker-Planck deviations and enable simulation.

## **Acknowledgments and Disclosure of Funding**

The authors would like to thank Aleksander Durumeric, Yaoyi Chen, Maximilian Schebek, Klara Bonneau, Winfried Ripken, Marcel Kollovich, Nicholas Gao, Matej Mezera, Yuanqi Du, Jiajun He, 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](http://www.nhr-verein.de/unsere-partner)).

## References

- Abdin, O. and Kim, P. M. (2024). Direct conformational sampling from peptide energy landscapes through hypernetwork-conditioned diffusion. *Nature Machine Intelligence*, 6(7):775–786.
- 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., Židek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., and Jumper, J. M. (2024). Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500.
- Albergo, M. S. and Vanden-Eijnden, E. (2025). NETS: A non-equilibrium transport sampler. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J., editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 1026–1055. PMLR.
- Arts, M., Garcia Satorras, V., Huang, C.-W., Zügner, D., Federici, M., Clementi, C., Noé, F., Pinsler, R., and van den Berg, R. (2023). Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *Journal of Chemical Theory and Computation*, 19(18):6151–6159.
- 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. (2023). ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. arXiv:2211.01324.
- Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., Molinari, N., Smidt, T. E., and Kozinsky, B. (2022). E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453.
- Bolhuis, P. G., Chandler, D., Dellago, C., and Geissler, P. L. (2002). Transition path sampling: Throwing ropes over rough mountain passes, in the dark. *Annual review of physical chemistry*, 53(1):291–318.
- Boltzmann, L. (1868). *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.
- Bortoli, V. D., Hutchinson, M., Wirnsberger, P., and Doucet, A. (2024). Target score matching. arXiv:2402.08667.
- 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. (2018). JAX: composable transformations of Python+NumPy programs. <http://github.com/google/jax>.
- Charron, N. E., Bonneau, K., Pados-Trejo, A. S., Guljas, A., Chen, Y., Musil, F., 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. (2025). Navigating protein landscapes with a machine-learned transferable coarse-grained model. *Nature chemistry*, pages 1–9.
- Clementi, C. (2008). Coarse-grained models of protein folding: toy models or predictive tools? *Current opinion in structural biology*, 18(1):10–15.
- Corso, G., Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T. (2023). DiffDock: Diffusion steps, twists, and turns for molecular docking. In *International Conference on Learning Representations*.
- Daigavane, A., Vani, B. P., Davidson, D., Saremi, S., Rackers, J. A., and Kleinhenz, J. (2025). Jamun: Bridging smoothed molecular dynamics and score-based learning for conformational ensemble generation. In *ICML 2025 Generative AI and Biology (GenBio) Workshop*.
- Daras, G., Dagan, Y., Dimakis, A., and Daskalakis, C. (2023). Consistent diffusion models: Mitigating sampling drift by learning to be consistent. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 42038–42063. Curran Associates, Inc.

- Dibak, M., Klein, L., Krämer, A., and Noé, F. (2022). Temperature steerable flows and Boltzmann generators. *Phys. Rev. Res.*, 4:L042005.
- Diez, J. V., Atance, S. R., Engkvist, O., and Olsson, S. (2024). Generation of conformational ensembles of small molecules via surrogate model-assisted molecular dynamics. *Machine Learning: Science and Technology*, 5(2):025010.
- Du, Y., Durkan, C., Strudel, R., Tenenbaum, J. B., Dieleman, S., Fergus, R., Sohl-Dickstein, J., Doucet, A., and Grathwohl, W. S. (2023). 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., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 8489–8510. PMLR.
- Du, Y., Plainer, M., Brekelmans, R., Duan, C., Noé, F., Gomes, C. P., Aspuru-Guzik, A., and Neklyudov, K. (2024). 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., editors, *Advances in Neural Information Processing Systems*, volume 37, pages 65791–65822. Curran Associates, Inc.
- 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. (2023). Machine learned coarse-grained protein force-fields: Are we there yet? *Current opinion in structural biology*, 79:102533.
- Durumeric, A. E. P., Chen, Y., Noé, F., and Clementi, C. (2024). Learning data efficient coarse-grained molecular dynamics from forces and noise. arXiv:2407.01286.
- Eastman, P., Swails, J., Chodera, J. D., McGibbon, R. T., Zhao, Y., Beauchamp, K. A., Wang, L.-P., Simonnet, A. C., Harrigan, M. P., Stern, C. D., Wiewiora, R. P., Brooks, B. R., and Pande, V. S. (2017). OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLOS Computational Biology*, 13(7):e1005659.
- Fornberg, B. (1988). Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation*, 51(184):699–706.
- Ganjanesh, A., Kang, Y., Liu, Y., Zhang, R., Lin, Z., and Huang, H. (2025). 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., editors, *Computer Vision – ECCV 2024*, pages 54–71, Cham. Springer Nature Switzerland.
- Guo, L., Wu, H., and Zhou, T. (2022). Normalizing field flows: Solving forward and inverse stochastic differential equations using physics-informed flow models. *Journal of Computational Physics*, 461:111202.
- Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., and van Zee, M. (2024). Flax: A neural network library and ecosystem for JAX. <http://github.com/google/flax>.
- Henkelman, G., Uberuaga, B. P., and Jónsson, H. (2000). A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *The Journal of chemical physics*, 113(22):9901–9904.
- Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.
- 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. (2021). Deeptime: a python library for machine learning dynamical models from time series data. *Machine Learning: Science and Technology*.
- Hu, Z., Zhang, Z., Karniadakis, G. E., and Kawaguchi, K. (2025). Score-based physics-informed neural networks for high-dimensional fokker–planck equations. *SIAM Journal on Scientific Computing*, 47(3):C680–C705.

- 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. (2020). Coarse graining molecular dynamics with graph neural networks. *The Journal of Chemical Physics*, 153(19):194101.
- Kim, D., Shin, S., Song, K., Kang, W., and Moon, I.-C. (2022). 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., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 11201–11228. PMLR.
- Kim, J. C., Bloore, D., Kapoor, K., Feng, J., Hao, M.-H., and Wang, M. (2024). Scalable normalizing flows enable boltzmann generators for macromolecules. In *International Conference on Learning Representations (ICLR)*.
- Klein, L., Foong, A., Fjelde, T., Mlodozieniec, B., Brockschmidt, M., Nowozin, S., Noe, F., and Tomioka, R. (2023a). 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., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 52863–52883. Curran Associates, Inc.
- Klein, L., Kelkar, A., Durumeric, A., Chen, Y., Clementi, C., and Noé, F. (2025). Operator forces for coarse-grained molecular dynamics. *The Journal of Chemical Physics*, 163(10).
- Klein, L., Krämer, A., and Noé, F. (2023b). Equivariant flow matching. *Advances in Neural Information Processing Systems*, 36:59886–59910.
- Klein, L. and Noé, F. (2024). Transferable Boltzmann generators. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C., editors, *Advances in Neural Information Processing Systems*, volume 37, pages 45281–45314. Curran Associates, Inc.
- Kmiecik, S., Gront, D., Kolinski, M., Wieteska, L., Dawid, A. E., and Kolinski, A. (2016). Coarse-grained protein models and their applications. *Chemical reviews*, 116(14):7898–7936.
- Koehler, F., Heckett, A., and Risteski, A. (2023). Statistical efficiency of score matching: The view from isoperimetry. In *International Conference on Learning Representations*.
- Köhler, J., Chen, Y., Krämer, A., Clementi, C., and Noé, F. (2023). Flow-matching: Efficient coarse-graining of molecular dynamics without forces. *Journal of Chemical Theory and Computation*, 19(3):942–952.
- Köhler, J., Klein, L., and Noé, F. (2020). Equivariant Flows: Exact Likelihood Generative Learning for Symmetric Densities. In *Proceedings of the 37th International Conference on Machine Learning*.
- Köhler, J., Krämer, A., and Noé, F. (2021). Smooth normalizing flows. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2796–2809. Curran Associates, Inc.
- Krämer, A., Durumeric, A. E., Charron, N. E., Chen, Y., Clementi, C., and Noé, F. (2023). Statistically optimal force aggregation for coarse-graining molecular dynamics. *The Journal of Physical Chemistry Letters*, 14(17):3970–3979.
- Lai, C.-H., Takida, Y., Murata, N., Uesaka, T., Mitsufuji, Y., and Ermon, S. (2023). 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., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 18365–18398. PMLR.
- Laio, A. and Parrinello, M. (2002). Escaping free-energy minima. *Proceedings of the national academy of sciences*, 99(20):12562–12566.
- Leimkuhler, B. and Matthews, C. (2015). *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*. Springer International Publishing.

- Lewis, S., Hempel, T., Jiménez-Luna, J., Gastegger, M., Xie, Y., Foong, A. Y., Satorras, V. G., Abdin, O., Veeling, B. S., Zaporozhets, I., Chen, Y., Yang, S., Foster, A. E., Schneuing, A., Nigam, J., Barbero, F., Vincent, S., Campbell, A., Yim, J., Lienen, M., Shi, Y., Zheng, S., Schulz, H., Munir, U., Sordillo, R., Tomioka, R., Clementi, C., and Noé, F. (2025). Scalable emulation of protein equilibrium ensembles with generative deep learning. *Science*, page eadv9817.
- Li, P., Li, Z., Zhang, H., and Bian, J. (2023). On the generalization properties of diffusion models. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 2097–2127. Curran Associates, Inc.
- Lindorff-Larsen, K., Piana, S., Dror, R. O., and Shaw, D. E. (2011). How fast-folding proteins fold. *Science*, 334(6055):517–520.
- Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. (2023). Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*.
- Lipman, Y., Havasi, M., Holderrieth, P., Shaul, N., Le, M., Karrer, B., Chen, R. T. Q., Lopez-Paz, D., Ben-Hamu, H., and Gat, I. (2024). Flow matching guide and code. arXiv:2412.06264.
- Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *Energy Based Models Workshop*.
- 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. (2015). Mdtraj: A modern open library for the analysis of molecular dynamics trajectories. *Biophysical Journal*, 109(8):1528 – 1532.
- Midgley, L. I., Stimper, V., Simm, G. N. C., Schölkopf, B., and Hernández-Lobato, J. M. (2023). Flow annealed importance sampling bootstrap. In *The Eleventh International Conference on Learning Representations*.
- Müller, K. and Brown, L. D. (1979). Location of saddle points and minimum energy paths by a constrained simplex optimization procedure. *Theoretica Chimica Acta*, 53(1):75–93.
- Noé, F., Olsson, S., Köhler, J., and Wu, H. (2019). Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147.
- Noid, W. G. (2013). Perspective: Coarse-grained models for biomolecular systems. *The Journal of chemical physics*, 139(9).
- Øksendal, B. (2003). *Stochastic Differential Equations*, pages 65–84. Springer Berlin Heidelberg.
- Pérez-Hernández, G., Paul, F., Giorgino, T., De Fabritiis, G., and Noé, F. (2013). Identification of slow molecular order parameters for markov model construction. *The Journal of chemical physics*, 139(1).
- Plainer, M., Stärk, H., Bunne, C., and Günemann, S. (2023a). Transition path sampling with boltzmann generator-based mcmc moves. In *Generative AI and Biology Workshop*.
- Plainer, M., Toth, M., Dobers, S., Stärk, H., Corso, G., Marquet, C., and Barzilay, R. (2023b). DiffDock-Pocket: Diffusion for pocket-level docking with sidechain flexibility. In *Machine Learning in Structural Biology*.
- Plattner, N., Doerr, S., De Fabritiis, G., and Noé, F. (2017). Complete protein–protein association kinetics in atomic detail revealed by molecular dynamics simulations and markov modelling. *Nature chemistry*, 9(10):1005–1011.
- Raja, S., Sipka, M., Psenka, M., Kreiman, T., Pavelka, M., and Krishnapriyan, A. S. (2025). Action-minimization meets generative modeling: Efficient transition path sampling with the onsager-machlup functional. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J., editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 50972–51008. PMLR.
- Salimans, T. and Ho, J. (2021). Should EBMs model the energy or the score? In *Energy Based Models Workshop*.

- Särkkä, S. and Solin, A. (2019). *Applied Stochastic Differential Equations*. Cambridge University Press.
- Schebek, M., Invernizzi, M., Noé, F., and Rogal, J. (2024). Efficient mapping of phase diagrams with conditional boltzmann generators. *Machine Learning: Science and Technology*.
- Schreiner, M., Winther, O., and Olsson, S. (2023). Implicit transfer operator learning: Multiple time-resolution models for molecular dynamics. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Schütt, K., Kindermans, P.-J., Saucedo Felix, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. (2017). Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Shi, Y., Huang, Z., Feng, S., Zhong, H., Wang, W., and Sun, Y. (2021). Masked label prediction: Unified message passing model for semi-supervised classification. In Zhou, Z.-H., editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1548–1554. International Joint Conferences on Artificial Intelligence Organization.
- Song, Y. and Ermon, S. (2019). 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., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.
- Tamagnone, S., Laio, A., and Gabrié, M. (2024). Coarse-grained molecular dynamics with normalizing flows. *Journal of Chemical Theory and Computation*, 20(18):7796–7805.
- Tan, C. B., Bose, A. J., Lin, C., Klein, L., Bronstein, M. M., and Tong, A. (2025). Scalable equilibrium sampling with sequential boltzmann generators. arXiv:2502.18462.
- Torrie, G. M. and Valleau, J. P. (1977). Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of computational physics*, 23(2):187–199.
- Trippe, B. L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R., and Jaakkola, T. (2023). Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. In *International Conference on Learning Representations*.
- Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674.
- Wang, J. and Hou, T. (2011a). Application of molecular dynamics simulations in molecular property prediction. 1. density and heat of vaporization. *Journal of chemical theory and computation*, 7(7):2151–2165.
- Wang, J. and Hou, T. (2011b). Application of molecular dynamics simulations in molecular property prediction ii: diffusion coefficient. *Journal of computational chemistry*, 32(16):3505–3519.
- Wang, J., Olsson, S., Wehmeyer, C., Pérez, A., Charron, N. E., Fabritiis, G. D., Noé, F., and Clementi, C. (2019). Machine learning of coarse-grained molecular dynamics force fields. *ACS Cent. Sci.*, 5:755–767.
- 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. (2023). De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100.
- Wirnsberger, P., Ballard, A. J., Papamakarios, G., Abercrombie, S., Racaniere, S., Pritzel, A., Jimenez Rezende, D., and Blundell, C. (2020). Targeted free energy estimation via learned mappings. *The Journal of Chemical Physics*, 153(14):144112.

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

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

## A Proofs, Derivations, and Formalisms

### A.1 Diffusion

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

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

where

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

with the hyperparameters from Song et al. (2021) such that  $(\beta_{\min}, \beta_{\max}) = (0.1, 20)$ . 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  $\mathbf{f}$  and  $g$ , we can write the transition kernel as a Gaussian (Särkkä and Solin, 2019) with a moving mean and standard deviation such that

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

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

### A.2 Residual Loss

In this section, we prove Equation (12) and show that  $\tilde{R}(\mathbf{x}, t; \mathbf{v})$  can be used to get an unbiased estimation of  $\tilde{R}(\mathbf{x}, t)$ . For simplicity of notation, let us express the log Fokker–Planck equations from Equation (8) as

$$\frac{1}{2}g^2(t) \operatorname{div}_{\mathbf{x}} \mathbf{s}^{\theta}(\mathbf{x}, t) + \gamma_{\theta}(\mathbf{x}, t) = 0, \quad (21)$$

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

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

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  $t$ ,

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

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

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

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

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

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

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

Based on this, we can obtain

$$\begin{aligned}\tilde{R}(\mathbf{x}, t) &= \mathbb{E}_{\mathbf{v}} \left[ \tilde{R}(\mathbf{x}, t; \mathbf{v}) \right] \\ &= \mathbb{E}_{\mathbf{v}} \left[ \frac{1}{2} g^2(t) \left( \frac{\mathbf{v}}{\sigma} \right)^\top \frac{\mathbf{s}^\theta(\mathbf{x} + \mathbf{v}, t) - \mathbf{s}^\theta(\mathbf{x} - \mathbf{v}, t)}{2\sigma} + \frac{\gamma_\theta(\mathbf{x} + \mathbf{v}, t) + \gamma_\theta(\mathbf{x} - \mathbf{v}, t)}{2} \right].\end{aligned}\tag{29}$$

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

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

$$\tilde{R}(\mathbf{x}, t; \mathbf{v}) = \frac{1}{2} g^2(t) \left( \frac{\mathbf{v}}{\sigma} \right)^\top \frac{\mathbf{s}^\theta(\mathbf{x} + \mathbf{v}, t) - \mathbf{s}^\theta(\mathbf{x} - \mathbf{v}, t)}{2\sigma} + \gamma_\theta(\mathbf{x} + \mathbf{v}, t).\tag{30}$$

We found  $\sigma = 10^{-4}$  to be an effective choice throughout our experiments.

### A.2.1 Relation with Hutchinson Trace Estimation

In fact, our treatment of the divergence term in the weak residual formulation is closely related to the Hutchinson’s trace estimator used by [Albergo and Vanden-Eijnden \(2025\)](#) for computing the PINN objective. Both approaches utilize Gaussian perturbations of  $\mathbf{x}$  to obtain unbiased estimates of the divergence. However, it is important to note a key difference in how the residual is defined. As described in [Equation \(29\)](#), in our weak residual  $\tilde{R}(\mathbf{x}, t)$ , we apply the same Gaussian perturbation not only to the divergence term but also to other terms in the residual. This means that when the strong residual  $R(\mathbf{x}, t)$  is exactly zero, the weak residual  $\tilde{R}(\mathbf{x}, t) = \mathbb{E}[R(\mathbf{x} + \mathbf{v}, t)]$  also remains exactly zero, regardless of the perturbation variance  $\sigma^2$ , avoiding truncation errors. This distinction allows our weak residual formulation to maintain consistency even under large perturbations.

### A.3 Finite Difference Approximation

To approximate  $\partial_t \log p^\theta$ , we relied on a finite difference approximation [Fornberg \(1988\)](#), as stated in [Equation \(14\)](#). For this estimation, we have followed the work of [Lai et al. \(2023\)](#), and used the hyperparameters that they suggested  $(h_s, h_d) = (0.001, 0.0005)$ .

### A.4 Connection with Flow Matching

The score–force relation introduced in [Equation \(6\)](#) is not specific to diffusion models. It also applies to other generative models, such as flow matching ([Lipman et al., 2023](#)), and in general to any model that can estimate  $\nabla_{\mathbf{x}} \log p_{t=0}(\mathbf{x})$ . In standard Gaussian flow matching, the learned vector field  $v^\theta$  relates to the score through

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) = \frac{1}{1-t} (t, v^\theta(\mathbf{x}, t) - \mathbf{x}),\tag{31}$$

as discussed by [Lipman et al. \(2024\)](#). Thus, forces can also be obtained from flow-matching models via reparameterization. However, in our experiments, we found that flow-based models perform worse near  $t \approx 0$ , likely because the stochasticity inherent to diffusion models improves generalization in this regime.

### A.5 Formalization Coarse-Graining

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  $\mathbf{x}$ , the Boltzmann distribution of CG samples  $\mathbf{z}$  can be recovered by

$$p(\mathbf{z}) \propto \int \exp\left(-\frac{U(\mathbf{x})}{k_B T}\right) \delta(\Xi(\mathbf{x}) - \mathbf{z}) d\mathbf{x},\tag{32}$$

which defines the CG potential up to a constant.  $\delta$  is the Dirac delta function.

## B Implementation Details and Experimental Setup

In this section, we will discuss additional details for the main experiments presented in the paper.

## B.1 Langevin Integrator

We perform NVT dynamics with the Langevin integrator as implemented in openMM (Eastman et al., 2017) version 8.2.0. The update reads

$$\begin{aligned} \mathbf{v}^{t+\Delta t} &= \alpha \mathbf{v}^t - \frac{1-\alpha}{\gamma} \nabla_{\mathbf{x}} U(\mathbf{x}) \mathbf{M}^{-1} + \sqrt{k_B T (1-\alpha)^2 \mathbf{M}^{-1}} \mathbf{R}, \\ \mathbf{x}^{t+\Delta t} &= \mathbf{x}^t + \Delta t \cdot \mathbf{v}^{t+\Delta t}, \end{aligned} \tag{33}$$

with  $\alpha = \exp(-\gamma \Delta t)$ . Here,  $t$  and  $\Delta t$  index the simulation timesteps (not the diffusion time),  $\gamma$  is the friction coefficient,  $\mathbf{M}$  the mass matrix,  $\mathbf{R} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$  a standard normal random vector,  $k_B$  the Boltzmann constant, and  $T$  the temperature.

## B.2 Metrics

To compute the JS divergence and the PMF error, we first project our data onto either the dihedral angles or the first two TIC coordinates (Pérez-Hernández et al., 2013) and then 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 for the dipeptides and into 100 bins for the proteins and compute the proportion of samples in each bin. 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.3 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 1 ms 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 3. 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. As for the optimizer, we have used AdamW (Loshchilov and Hutter, 2019).

Parameter	Diffusion	Mixture	Fokker–Planck	Both
Epochs	10000	7000, 2000, 1000	10000	7000, 2000, 1000
Max Learning Rate	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$
Min Learning Rate	$10^{-5}$	$10^{-5}$	$10^{-5}$	$10^{-5}$
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	$5 \cdot 10^{-4}$	$10^{-3}$ , 0, 0
Hidden Dimension	96	96, 96, 96	96	96, 96, 96
Layers	2	2, 2, 2	2	2, 2, 2

Table 3: **Alanine dipeptide.** This table contains the hyperparameters for the different models shown.

**Simulation.** To perform Langevin simulation, we have extracted the forces from the model via Equation (6) at  $t = 10^{-5}$  for all models except for *Two For One*, where we chose  $t = 0.02$ , which is the same value as presented in Arts et al. (2023) for the same number of training samples.

**Evaluation.** To compute the standard deviation in the experiments, we have trained the same model with three different seeds.

## B.4 Fast-Folding Proteins

**Dataset.** We evaluate our model on the proteins Chignolin (10 amino acids) and BBA (28 amino acids) using the data from Lindorff-Larsen et al. (2011). The reference data was simulated in explicit solvent with a step size of 2 fs at 340K for Chignolin and 325K for BBA. In total 106  $\mu$ s and 223  $\mu$ s were simulated for Chignolin and BBA. For training, we have randomly selected 80% of the samples which accounts to 427, 794 and 1, 300, 156 samples for Chignolin and BBA respectively.

**Architecture.** We have used the same architecture as for alanine dipeptide, with the only difference being that we use larger networks. Compare the hyperparameters in Table 4 for Chignolin and Table 5 for BBA. Note that for *Two For One* we used the same model as *Diffusion* but evaluated it at a different  $t$  for simulation. The hyperparameters slightly deviate from the choices in (Arts et al., 2023) to produce consistent results with our method. As optimizer, we used AdamW (Loshchilov and Hutter, 2019).

Parameter	Diffusion	Mixture	Fokker-Planck	Both
Epochs	2000	1700, 200, 100	2000	1700, 200, 100
Max Learning Rate	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$10^{-4}$	$3 \cdot 10^{-4}$
Min Learning Rate	$10^{-5}$	$10^{-5}$	$10^{-5}$	$10^{-5}$
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	$2 \cdot 10^{-3}$	$2 \cdot 10^{-3}$ , 0, 0
Hidden Dimension	128	128, 96, 96	128	128, 96, 96
Layers	3	3, 2, 2	2	3, 2, 2

Table 4: **Chignolin.** This table contains the hyperparameters for the different models shown.

Parameter	Diffusion	Mixture	Fokker-Planck	Both
Epochs	1500	1200, 400, 200	1500	1200, 400, 200
Max Learning Rate	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$10^{-4}$	$10^{-4}$
Min Learning Rate	$10^{-5}$	$10^{-5}$	$10^{-5}$	$10^{-5}$
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	$10^{-2}$	$2 \cdot 10^{-2}$ , 0, 0
Hidden Dimension	128	128, 128, 128	128	128, 128, 128
Layers	3	3, 3, 3	3	3, 3, 3

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

**Simulation.** To perform Langevin simulation, we have extracted the forces from the model via Equation (6) at  $t = 10^{-5}$  for all models except for *Two For One*, where we chose  $t = 0.02$  and  $t = 0.005$  for Chignolin and BBA respectively, which is the same value as presented in Arts et al. (2023) for the same number of training samples.

**Evaluation.** To compute the standard deviation in the experiments, we have used the same models but three different seeds for inference.

## 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 and Noé (2024) (CC BY 4.0): [https://osf.io/n8vz3/?view\\_only=1052300a21bd43c08f700016728aa96e](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  $1\mu\text{s}$ . 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 presented in the paper (also the metrics) to reduce inference time. The fifteen test dipeptides are: AC, AP, AT, ET, HP, HT, IM, KQ, KS, LW, NF, NY, RL, RV, TD.

**Architecture.** The hyperparameters are listed in Table 6. 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. As optimizer, we used AdamW (Loshchilov and Hutter, 2019).

Parameter	Diffusion	Mixture	Fokker–Planck	Both
Epochs	120	100, 20, 10	120	100, 20, 10
Max Learning Rate	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$
Min Learning Rate	$10^{-5}$	$10^{-5}$	$10^{-5}$	$10^{-5}$
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	$5 \cdot 10^{-4}$	$10^{-4}, 0, 0$
Hidden Dimension	128	128, 96, 96	128	128, 96, 96
Layers	3	3, 2, 2	3	3, 2, 2

Table 6: **Dipeptides.** This table contains the hyperparameters for the different models shown.

**Simulation.** To perform Langevin simulation, we have extracted the forces from the model via Equation (6) at  $t = 10^{-5}$  for 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  $t = 0.02$ .

**Evaluation.** To compute the standard deviation in the experiments, we compute the values across all testset dipeptides.

## B.6 Compute Infrastructure

We have used a single RTX 3090 GPU for the toy systems, an A100 with 80GB memory for alanine dipeptide, two A100 80GB GPUs for the dipeptide dataset and Chignolin, and four for BBA. Depending on availability, we have also used H100 for some experiments. For inference, we use a single GPU.

## 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 Further Studies and More Experiments

## C.1 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 and 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, aligning with knowledge from the force-field literature (Schütt et al., 2017; Batzner et al., 2022). 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.

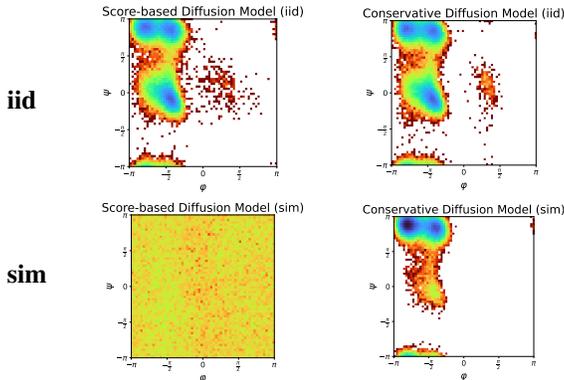


Figure 7: **Alanine dipeptide**. We compare a conservative diffusion model with a score-based model. We can see that around the low-density regions, the conservative parameterization generates better iid samples. As for simulation, a score-based model exhibits stability issues, and the simulation becomes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

### C.2 Mixture vs. More Parameters

In our experiments, using the MoE scheme naturally increases the total number of parameters. To assess whether the observed improvements stem from this additional capacity or from the mixture design itself, we perform ablation studies on alanine dipeptide. Specifically, we compare the original results from the paper against three variants: one with wider layers (*Wide*), one with more layers (*Deep*), and one using three instances of the conservative diffusion model combined into a *Mixture* of experts. The results in Table 7 show that *Mixture* outperforms both larger single models and achieves more stable simulation behavior, and requires the least time for training and inference (see also Appendix C.3). This demonstrates that the advantages of the MoE design extend beyond simple parameter scaling and can yield runtime improvements.

Method	Parameters	Training (↓)	iid JS (↓)	sim. JS (↓)	iid PMF (↓)	sim. PMF (↓)
Diffusion	647585	<b>50min</b>	<b>0.0081 ± 0.0003</b>	0.0695 ± 0.0517	0.095 ± 0.003	1.047 ± 0.924
Wide	1964321	67min	0.0082 ± 0.0003	0.0406 ± 0.0236	0.096 ± 0.003	0.467 ± 0.245
Deep	1968562	111min	<b>0.0078 ± 0.0003</b>	0.0376 ± 0.0087	<b>0.091 ± 0.002</b>	0.478 ± 0.038
Mixture	1957567	<b>50min</b>	<b>0.0079 ± 0.0003</b>	<b>0.0264 ± 0.0085</b>	<b>0.093 ± 0.007</b>	<b>0.325 ± 0.113</b>

Table 7: **Alanine dipeptide**. Ablation study comparing the MoE scheme with parameter scaling. Reported are the JS divergence and PMF error for *iid* sampling and simulation (*sim*). Lower values indicate better performance. The MoE achieves the most stable and accurate simulations while maintaining the same training cost as the baseline.

### C.3 Runtime Comparison

We compare the training and inference runtimes of different approaches in Table 8. While the MoE setup could, in principle, be trained in parallel, our current implementation does not distribute the experts across devices and is therefore not fully optimized. For the low-dimensional systems, this

can even introduce a small additional overhead. As a result, runtime improvements are only visible for larger systems. In contrast, models trained with Fokker–Planck regularization show a clear computational overhead due to the additional residual evaluation, which requires multiple forward passes per training step. During inference, however, they perform the same as their counterparts.

Dataset	Task	Diffusion	Mixture	Fokker–Planck	Both
Alanine Dipeptide	Training	50min	50min	4h 39min	3h 59min
	iid Sampling	3min	4min	3min	4min
Chignolin	Training	4h 58min*	3h 0min	22h 36min	21h 33min
	iid Sampling	43min	14min	43min	14min
BBA	Training	1d 20h*	1d 19h*	7d 2h	6d 15h
	iid Sampling	12h 57min	5h 56min	12h 57min	5h 56min
Dipeptides	Training	4h 5min	3h 50min	28h 39min	27h 5min
	iid Sampling	8min	4min	8min	4min

Table 8: Training and inference runtimes for all models. Methods marked with \* were trained using half the number of GPUs (one for Chignolin and two for BBA). The MoE model reduces inference time substantially due to its modular design, while the Fokker–Planck regularization introduces additional computational overhead from multiple forward evaluations during training.

In terms of simulation, the main computational advantage of our approach arises from its ability to perform simulations in parallel and to leverage coarse-graining. In classical molecular dynamics, simulating a protein typically requires modeling the solvent explicitly, which adds thousands of water molecules and substantially increases computational cost. Coarse-graining eliminates these degrees of freedom, thereby accelerating computation and improving parallel scalability.

For instance, in the dipeptide dataset, where each molecule contains only a few atoms and no solvent to coarse-grain, our model achieves 150 parallel simulations at 830 steps/s each, corresponding to roughly 125k steps/s on a single NVIDIA A100. In comparison, a conventional force-field simulation on an NVIDIA V40 reaches around 10k steps/s. Even larger gains can be expected for systems with explicit solvent, where coarse-graining removes many more particles. As an example, our BBA protein simulation (100 parallel runs of 1M steps each) completes in approximately one hour.

#### C.4 Toy System

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

$$\begin{aligned}
 U(x, y) = & -200 \cdot \exp(-(x-1)^2 - 10y^2) \\
 & -100 \cdot \exp(-x^2 - 10 \cdot (y-0.5)^2) \\
 & -170 \cdot \exp(-6.5 \cdot (0.5+x)^2 + 11 \cdot (x+0.5) \cdot (y-1.5) - 6.5 \cdot (y-1.5)^2) \\
 & + 15 \cdot \exp(0.7 \cdot (1+x)^2 + 0.6 \cdot (x+1) \cdot (y-1) + 0.7 \cdot (y-1)^2).
 \end{aligned} \tag{34}$$

To generate training samples from this potential, we have performed a Langevin simulation (compare Equation (5)). For this, we have performed 5M steps with  $k_B T = 23$ ,  $dt = 0.005$ ,  $M = 0.5 \cdot I$ , 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 9.

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 Dimension	92	64, 64, 54	92	64, 64, 54
Layers	3	3, 2, 2	3	3, 2, 2
$\alpha$	0	0	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}, 0, 0$

Table 9: Müller-Brown. This table contains the hyperparameters for the different models shown.

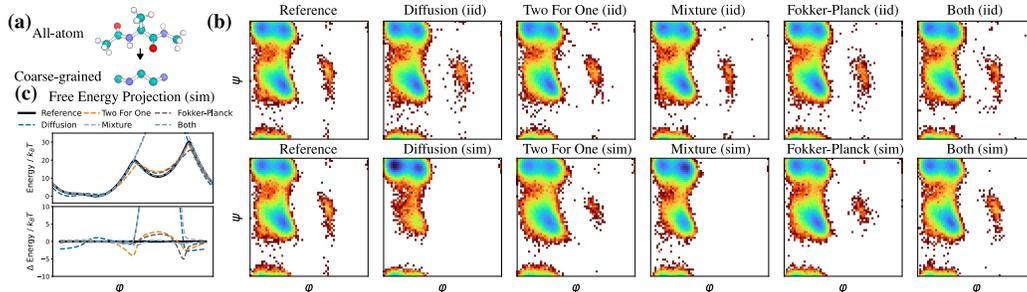


Figure 9: **Alanine dipeptide.** (a) The coarse-graining scheme. (b) Comparison of the Ramachandran plots of different methods for iid sampling and Langevin simulation. (c) The projection of the free energy surface and differences along the dihedral angle  $\varphi$  for samples generated with simulation.

**Evaluation.** In Figure 8, we can see the free energy plots of the different methods. 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. Numerical results can be seen in Table 10.

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

Table 10: **Müller-Brown.** 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.

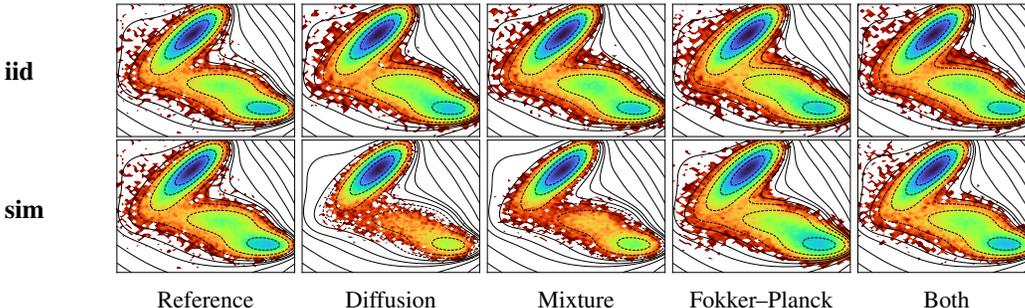


Figure 8: **Müller-Brown.** Comparing free energy plots of different models for classical diffusion sampling (iid) and Langevin simulation (sim). The energies should align with the training data (Reference).

## C.5 Alanine Dipeptide

In this section, we report some further results and plots on alanine dipeptide.

### C.5.1 Free Energies

**Inconsistent sampling.** Figure 9 (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

Method	iid JS ( $\downarrow$ )	sim. JS ( $\downarrow$ )	iid PMF ( $\downarrow$ )	sim. PMF ( $\downarrow$ )
Diffusion	<b>0.0081</b> $\pm$ <b>0.0003</b>	0.0695 $\pm$ 0.0517	<b>0.095</b> $\pm$ <b>0.003</b>	1.047 $\pm$ 0.924
Two For One	<b>0.0081</b> $\pm$ <b>0.0003</b>	0.0158 $\pm$ 0.0002	<b>0.098</b> $\pm$ <b>0.006</b>	0.206 $\pm$ 0.004
Mixture	<b>0.0080</b> $\pm$ <b>0.0004</b>	0.0353 $\pm$ 0.0117	<b>0.092</b> $\pm$ <b>0.007</b>	0.388 $\pm$ 0.109
Fokker–Planck	<b>0.0082</b> $\pm$ <b>0.0002</b>	0.0090 $\pm$ 0.0006	0.098 $\pm$ 0.003	0.104 $\pm$ 0.004
Both	<b>0.0082</b> $\pm$ <b>0.0004</b>	<b>0.0080</b> $\pm$ <b>0.0002</b>	<b>0.097</b> $\pm$ <b>0.008</b>	<b>0.091</b> $\pm$ <b>0.004</b>

Table 11: **Alanine dipeptide**. Comparison of JS divergence and PMF error. To compute the mean and the standard deviation, we have trained and evaluated three models with three different seeds.

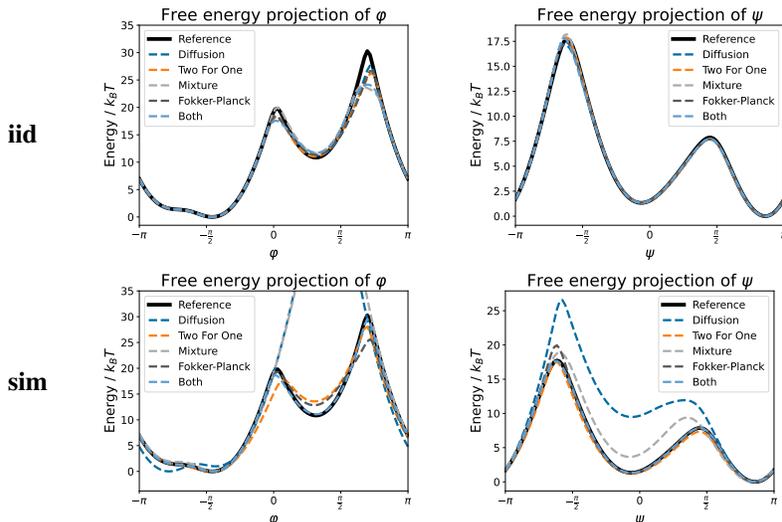


Figure 10: **Alanine dipeptide**. Comparing the free energy of along the dihedral angles  $\varphi, \psi$  for iid sampling and simulation across different models.

mode. This is reflected in the numerical results in Table 11, 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.

**Consistent models.** *Fokker–Planck* regularization enables the model to recover the missing states without modifying the diffusion time, and thus preserving structural accuracy. Table 11 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 barely 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 9 (c)). The similar iid performance of *Both* and *Fokker–Planck* suggests that applying regularization introduces additional constraints and restrictions on the model and hence can degrade generative quality.

**Detailed free energy.** In Figure 10 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.

### C.5.2 Fokker–Planck Residual

In Figure 11 the Fokker–Planck residual error  $\|\mathcal{F}_{p^\theta}(\mathbf{x}, t) - \partial_t \log p_t^\theta(\mathbf{x})\|_2$  is reported. Overall, the results are similar to what was reported in Figure 6. We can note that the Fokker–Planck error of *Mixture* is lower than *Diffusion*, indicating that MoE can improve the model’s consistency.

### C.5.3 Bonds

Figure 12 shows the immediate 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 12. In this table, *Mixture* provides the best bonds during simulation, but does not recover all modes.

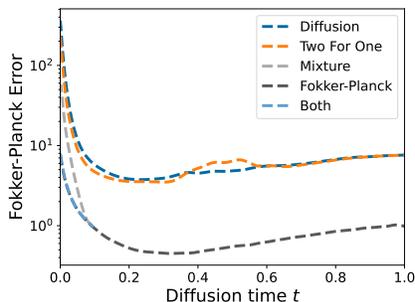


Figure 11: **Alanine dipeptide**. Comparing the Fokker–Planck error for  $\log p^{\theta}$  of multiple models.

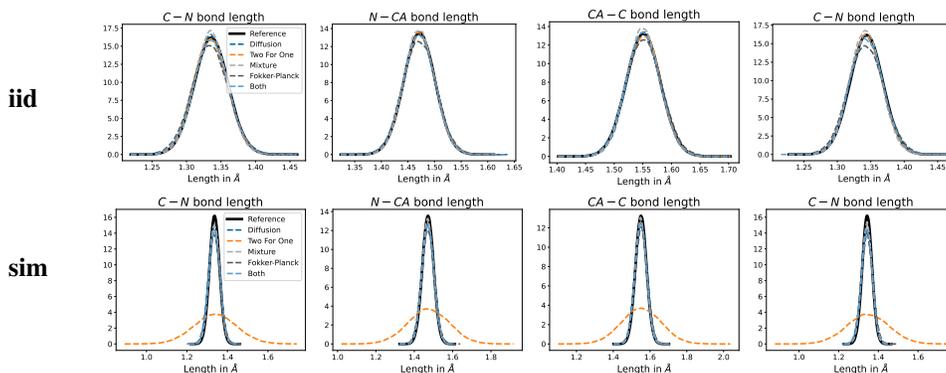


Figure 12: **Alanine dipeptide**. This illustration shows all direct bond lengths sampled by the different methods.

Method	iid relative W1 ( $\downarrow$ )	sim relative W1 ( $\downarrow$ )
Diffusion	<b>0.88 <math>\pm</math> 0.70</b>	0.83 $\pm$ 0.15
Two For One	<b>0.88 <math>\pm</math> 0.58</b>	23.51 $\pm$ 1.91
Mixture	<b>1.13 <math>\pm</math> 0.46</b>	<b>0.47 <math>\pm</math> 0.12</b>
Fokker–Planck	1.65 $\pm$ 0.37	1.23 $\pm$ 0.16
Both	<b>1.00 <math>\pm</math> 0.00</b>	1.00 $\pm$ 0.00

Table 12: Comparison of methods on alanine dipeptide based on the Wasserstein 1 distance of the C–N bond lengths to the reference data. 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*.

### C.5.4 Force Matching

We further compare our model against prior work on force matching, which either employs explicit forces from the training set (Husic et al., 2020) or infers forces implicitly from noise in the data (Durumeric et al., 2024; Klein et al., 2025). All these approaches share the same underlying architecture, the CGSchNet model (Husic et al., 2020), which combines a trainable SchNet component (Schütt et al., 2017) with fixed energy terms accounting for bonded interactions such as bonds, angles, and dihedrals. In contrast, our models operate without any such energy priors, and thus require significantly less prior knowledge about the investigated systems. As these previous studies commonly evaluated their models on a six-bead representation of alanine dipeptide (Wang et al., 2019; Husic et al., 2020; Klein et al., 2025), we retrain our models as well as the *Two for One* model on the same coarse-grained system. The corresponding training and evaluation trajectory was generated via MD for 1  $\mu$ s in explicit solvent (Wang et al., 2019), and the training set for all methods consists of 100k samples. For the force-matching baselines, we report the results from Klein et al. (2025). The comparison, shown in Table 13, evaluates again the PMF between the simulated and target trajectories. We only evaluate the simulations, as the force-matching methods do not permit independent sampling. Overall, our models achieve superior performance compared to all force matching baselines on this dataset, despite not using any force information, unlike Husic et al. (2020).

Method	sim. JS ( $\downarrow$ )	sim PMF ( $\downarrow$ )
Force Matching	0.0243 $\pm$ 0.0015	0.322 $\pm$ 0.005
Noise and Force Matching	0.0402 $\pm$ 0.0066	0.864 $\pm$ 0.146
Operator Forces	0.0214 $\pm$ 0.0041	0.301 $\pm$ 0.043
Operator Forces	0.0166 $\pm$ 0.0009	0.282 $\pm$ 0.031
Diffusion	0.0928 $\pm$ 0.0165	1.135 $\pm$ 0.205
Two for One	0.0151 $\pm$ 0.0006	0.168 $\pm$ 0.009
Mixture	0.0506 $\pm$ 0.0370	0.545 $\pm$ 0.429
Fokker-Planck	<b>0.0092 <math>\pm</math> 0.0017</b>	<b>0.107 <math>\pm</math> 0.028</b>
Both	<b>0.0075 <math>\pm</math> 0.0024</b>	<b>0.090 <math>\pm</math> 0.049</b>

Table 13: **Six-bead alanine dipeptide**. Comparison of different models for learning forces. The table reports the JS divergence and the PMF error between simulated and target distributions (lower is better). Our Fokker-Planck-regularized models outperform all force-matching methods despite being trained without any force labels.

## C.6 Proteins

We now provide additional comparisons and ablation studies for the independent contributions of MoE and Fokker-Planck regularization for Chignolin and BBA as well as analysis of dynamics.

### C.6.1 Free Energy Surfaces

We present the same free energy surfaces as in the main text, but now distinguish between *Mixture*, *Fokker-Planck*, and *Both* (referred to as *Ours* in the main text). In Figure 13, we show the free energy along the first two TIC coordinates by projecting the data into 100 bins, and in Figure 14, we show the corresponding kernel density estimates along each axis. Beyond the results discussed in the main text, we observe that simulations using *Diffusion* (i.e., evaluating the model at  $t = 0$ ) are unstable. The MoE variant in *Mixture* improves stability but does not fully resolve these issues. In contrast, introducing the Fokker-Planck regularization (with or without mixture) yields stable simulations, and the resulting free energy surfaces from diffusion sampling and simulation become consistent. In Figure 14, the models appear nearly identical for *iid* sampling but differ for *sim*. The qualitative metrics are similar between Fokker-Planck with and without MoE, showing only minor differences.

The same trends are observed for BBA in Figures 15 and 16. The *Diffusion* model fails to recover a meaningful free energy landscape, while *Mixture* using MoE substantially improves the results but remains insufficient. Once the Fokker-Planck regularization is applied, the simulations become consistent with diffusion sampling. As shown in Figure 16, the *iid* results of several models are nearly indistinguishable in the one-dimensional projections, making it difficult to determine which performs best. For *sim*, it is also unclear whether *Fokker-Planck* or *Both* yields better performance. A quantitative comparison is provided in Appendix C.6.3.

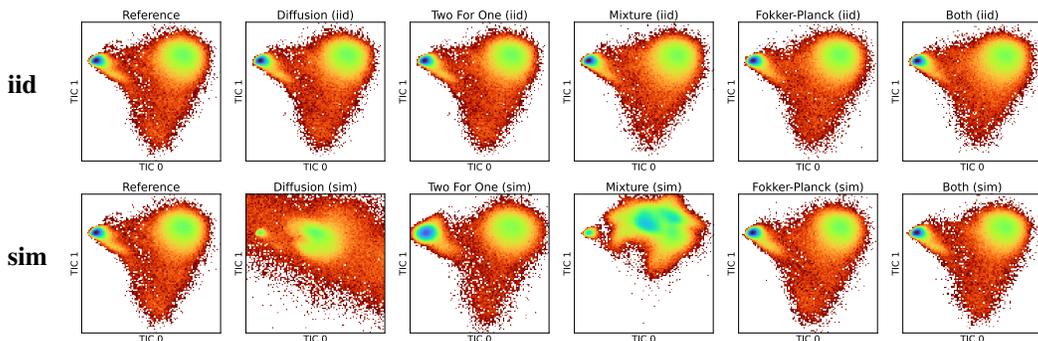


Figure 13: **Chignolin**. We compare the free energy projected onto the first two TIC axes for all presented methods for *iid* sampling and Langevin simulation (*sim*). The aim is to match the *Reference* (left) and have a self-consistent model where *iid* and *sim* closely aligns.

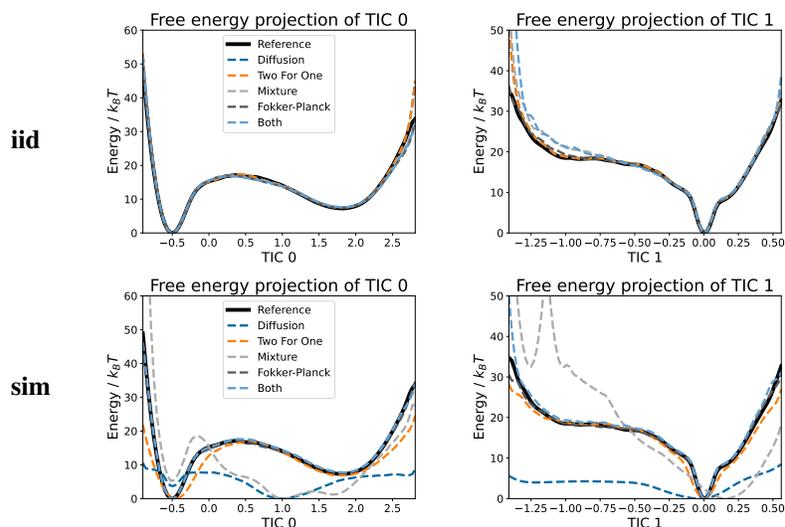


Figure 14: **Chignolin**. Comparing the free energy along the first two TIC axes for iid sampling and simulation across different models with the reference in black.

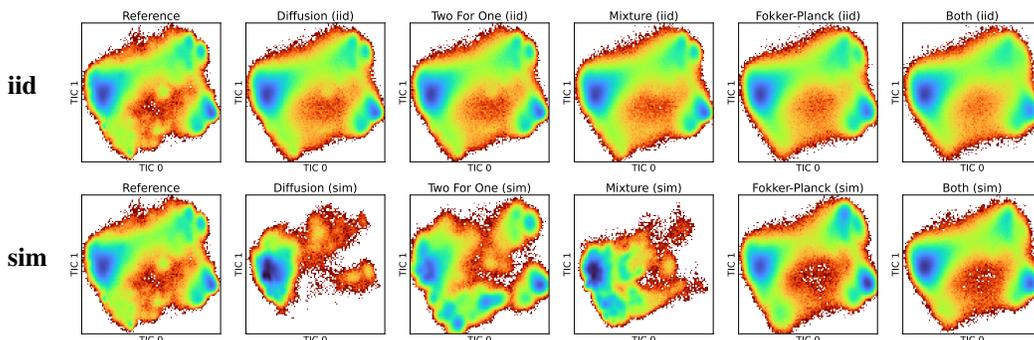


Figure 15: **BBA**. We compare the free energy projected onto the first two TIC axes for all presented methods for *iid* sampling and Langevin simulation (*sim*). The aim is to match the *Reference* (left) and have a self-consistent model where *iid* and *sim* closely aligns.

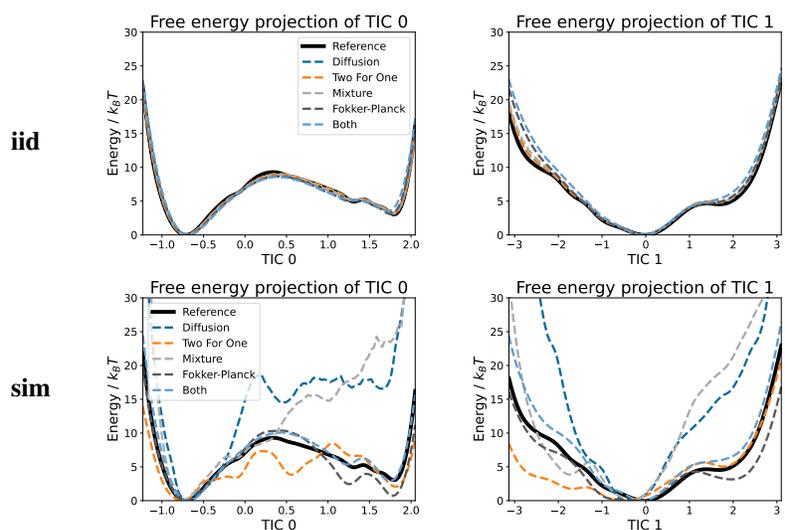


Figure 16: **BBA**. Comparing the free energy along the first two TIC axes for iid sampling and simulation across different models with the reference in black.

### C.6.2 Contact Maps

To analyze structural consistency beyond local distances, we compute pairwise contact maps between all atoms. As such, they provide a compact way of representing molecular structure and indicate which atom pairs are in spatial proximity. Specifically, for each generated ensemble, we compute the distances between all atom pairs and record whether they fall below a threshold of 10 Å. Averaging these contacts over all generated samples yields a probability map indicating how frequently each atom pair interacts. We then take the log of these to smooth out the range and improve visualization.

This is visualized for Chignolin and BBA in Figures 17 and 18. This representation captures the overall spatial organization of a molecule, such as which regions are folded together or form stable secondary structures. Comparing contact maps from different models provides an intuitive way to assess whether the generated ensembles reproduce realistic atomic arrangements.

These figures align with the overall trends in the paper showing that: *iid* works well across all models, *Diffusion* and *Mixture* is unstable for simulation, and *Fokker-Planck/Both* match the reference and the sampling methods are consistent. However, when examining these specific metrics, the additional noise introduced by *Two For One* is less apparent. We attribute this to the binary cutoff used to determine whether two atoms are in contact, which reduces sensitivity to small structural perturbations. This effect is not present in the pairwise distance matrix analysis in Tables 14 and 15.

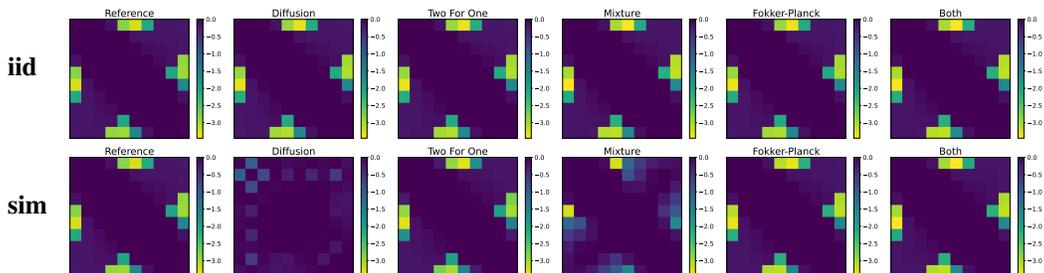


Figure 17: **Chignolin**. Pairwise contact maps comparing different models and the two sampling methods. Two atoms are considered in contact if their distance is below 10 Å. Bright colors (corresponding to small values) indicate a low probability of close proximity. The maps are symmetric by construction.

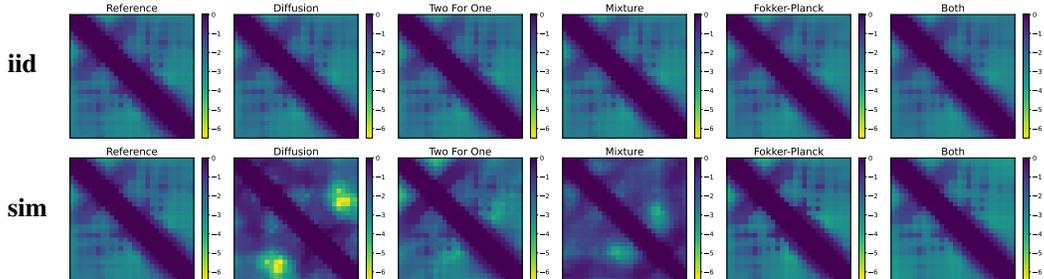


Figure 18: **BBA**. Pairwise contact maps comparing different models and the two sampling methods. Two atoms are considered in contact if their distance is below 10 Å. Bright colors (corresponding to small values) indicate a low probability of close proximity. The maps are symmetric by construction.

### C.6.3 Quantitative Results

In addition to the results presented in the main paper, we provide further quantitative analyses in this section. Alongside the PMF error, we also report the JS divergence. While the PMF error evaluates energy differences, the JS divergence compares probability densities, which can make model differences less pronounced. Both metrics are computed in 2D TICA space.

We additionally report the JS divergence computed on the pairwise distance (PWD) matrix. For this, we first calculate all interatomic distances and extract the upper triangular part of the full PWD matrix to avoid redundancy. We then remove the diagonal elements and exclude pairs separated by fewer than three bonds to focus on nonlocal structural correlations. In other words, we compare distances between atoms that are not close neighbors along the chain. For each atom pair, we then construct normalized histograms of their distance distributions for both the generated and reference ensembles and compute the JS divergence between them. Averaging these divergences across all pairs yields a

single scalar value that quantifies how well the generated ensemble reproduces the overall spatial organization of the molecule. Lower values correspond to higher structural fidelity.

The resulting metrics are summarized in [Tables 14](#) and [15](#) for Chignolin and BBA, respectively. The trends are consistent with those discussed in the main text: for Chignolin, we observe only a slight decrease in *iid* sampling quality when introducing Fokker–Planck regularization or the MoE model. However, for the larger and more complex BBA system, it becomes evident that achieving a consistent energy-based model requires trade-offs in *iid* sampling quality. In contrast, our models excel in the *sim* setting, achieving the best results in both energetic and structural metrics, particularly for the PWD-based measure of structural fidelity.

Method	<i>iid</i> JS ( $\downarrow$ )	<i>sim.</i> JS ( $\downarrow$ )	<i>iid</i> PMF ( $\downarrow$ )	<i>sim.</i> PMF ( $\downarrow$ )	<i>iid</i> PWD JS ( $\downarrow$ )	<i>sim.</i> PWD JS ( $\downarrow$ )
Diffusion	<b>0.0036</b> $\pm$ <b>0.0001</b>	0.4351 $\pm$ 0.0141	<b>0.027</b> $\pm$ <b>0.000</b>	63.804 $\pm$ 0.372	<b>0.0001</b> $\pm$ <b>0.0000</b>	0.3817 $\pm$ 0.0009
Two For One	<b>0.0036</b> $\pm$ <b>0.0001</b>	0.1023 $\pm$ 0.0008	<b>0.027</b> $\pm$ <b>0.000</b>	1.438 $\pm$ 0.019	<b>0.0001</b> $\pm$ <b>0.0000</b>	0.0082 $\pm$ 0.0000
Mixture	0.0042 $\pm$ 0.0001	0.4336 $\pm$ 0.0075	0.033 $\pm$ 0.000	11.185 $\pm$ 0.430	0.0003 $\pm$ 0.0000	0.2045 $\pm$ 0.0004
Fokker–Planck	0.0048 $\pm$ 0.0001	<b>0.0050</b> $\pm$ <b>0.0001</b>	0.037 $\pm$ 0.000	<b>0.039</b> $\pm$ <b>0.001</b>	0.0004 $\pm$ 0.0000	<b>0.0008</b> $\pm$ <b>0.0000</b>
Both	0.0045 $\pm$ 0.0001	<b>0.0050</b> $\pm$ <b>0.0008</b>	0.035 $\pm$ 0.001	<b>0.038</b> $\pm$ <b>0.006</b>	0.0003 $\pm$ 0.0000	<b>0.0012</b> $\pm$ <b>0.0005</b>

Table 14: **Chignolin**. Quantitative comparison of all methods based on JS divergence, PMF error (both computed in TICA space), and the mean JS divergence of the pairwise distance (PWD) distributions of atoms. Lower values indicate better agreement with the reference data. The *iid* metrics assess equilibrium sampling quality, while the *sim* metrics evaluate the consistency of the simulated dynamics.

Method	<i>iid</i> JS ( $\downarrow$ )	<i>sim.</i> JS ( $\downarrow$ )	<i>iid</i> PMF ( $\downarrow$ )	<i>sim.</i> PMF ( $\downarrow$ )	<i>iid</i> PWD JS ( $\downarrow$ )	<i>sim.</i> PWD JS ( $\downarrow$ )
Diffusion	<b>0.0043</b> $\pm$ <b>0.0000</b>	0.2014 $\pm$ 0.0043	<b>0.034</b> $\pm$ <b>0.000</b>	5.387 $\pm$ 0.144	<b>0.0006</b> $\pm$ <b>0.0000</b>	0.1003 $\pm$ 0.0050
Two For One	<b>0.0043</b> $\pm$ <b>0.0000</b>	0.1162 $\pm$ 0.0021	<b>0.034</b> $\pm$ <b>0.000</b>	1.624 $\pm$ 0.107	<b>0.0006</b> $\pm$ <b>0.0000</b>	0.0240 $\pm$ 0.0008
Mixture	0.0091 $\pm$ 0.0000	0.1964 $\pm$ 0.0113	0.081 $\pm$ 0.000	5.970 $\pm$ 0.278	0.0012 $\pm$ 0.0000	0.1016 $\pm$ 0.0020
Fokker–Planck	0.0130 $\pm$ 0.0000	0.0292 $\pm$ 0.0035	0.117 $\pm$ 0.001	<b>0.275</b> $\pm$ <b>0.033</b>	0.0018 $\pm$ 0.0000	0.0078 $\pm$ 0.0014
Both	0.0238 $\pm$ 0.0001	<b>0.0238</b> $\pm$ <b>0.0001</b>	0.234 $\pm$ 0.001	<b>0.254</b> $\pm$ <b>0.005</b>	0.0023 $\pm$ 0.0000	<b>0.0042</b> $\pm$ <b>0.0001</b>

Table 15: **BBA**. Quantitative comparison of all methods based on JS divergence, PMF error (both computed in TICA space), and the mean JS divergence of the pairwise distance (PWD) distributions of atoms. Lower values indicate better agreement with the reference data. The *iid* metrics assess equilibrium sampling quality, while the *sim* metrics evaluate the consistency of the simulated dynamics.

#### C.6.4 Dynamics

The main advantage of our method is that we have access to a *consistent* and physically meaningful energy function that allows us to perform simulation and recover dynamic properties. To illustrate this benefit, we analyze the recovered dynamics and provide further evaluations.

As in the main text for BBA, we visualize in [Figure 19](#) a single simulated trajectory for Chignolin. The system exhibits folding and unfolding transitions between metastable states, showing realistic conformational changes over time. For this, we employ the *Both* model, which combines Fokker–Planck regularization with the MoE scheme.

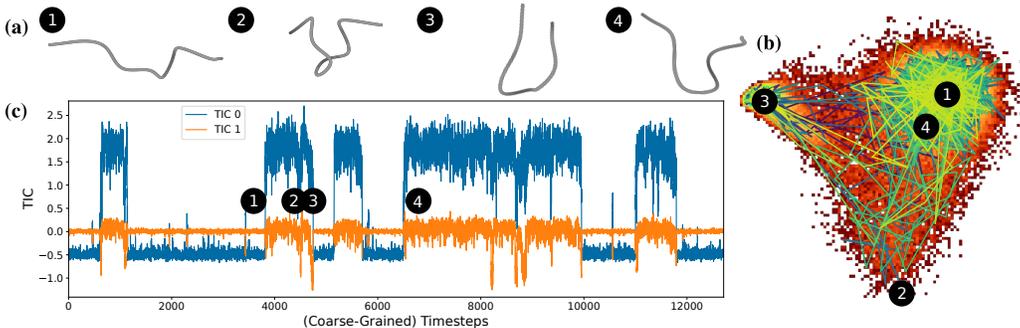


Figure 19: **Chignolin**. Recovering and analyzing the dynamic behavior. (a) 3D molecular structures from a single MD trajectory generated by our model. (b) Trajectory projected onto the first two TIC coordinates. (c) Illustrating the TIC coordinates over the coarse-grained (and subsampled) timesteps.

To further assess the long-time MD simulations, we follow the work of [Arts et al. \(2023\)](#) and cluster the conformations into three metastable states for Chignolin and four for BBA. State assignments are obtained via  $k$ -means clustering on the TIC coordinates, from which we estimate a Markov model and compute the transition probabilities between states. The state definitions and transition matrices are visualized in [Figures 20 and 21](#). Again, this analysis is only applicable to models that recover continuous dynamics rather than independent samples.

Although the visual differences between transition matrices are subtle, the Fokker–Planck–regularized models clearly provide the most accurate estimates of transition probabilities when compared to the reference simulation. To quantify this, we compute the mean JS divergence between the transition probabilities of each model and the reference, reported in [Table 16](#). These results confirm that Fokker–Planck regularization improves kinetic consistency, while the MoE scheme leads to slightly less accurate transitions; both when used independently and in combination with Fokker–Planck.

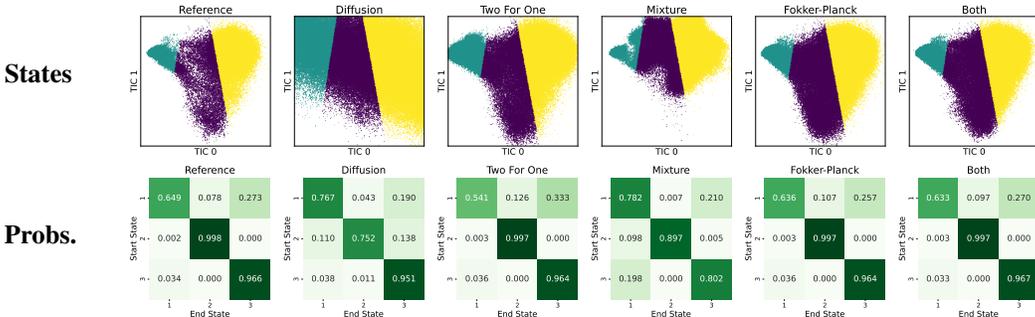


Figure 20: **Chignolin** Each conformation is assigned to one of three states based on the TIC coordinates (top row), shown as State 1 (purple), State 2 (green), and State 3 (yellow). Using these assignments, we estimate the corresponding transition probabilities between states.

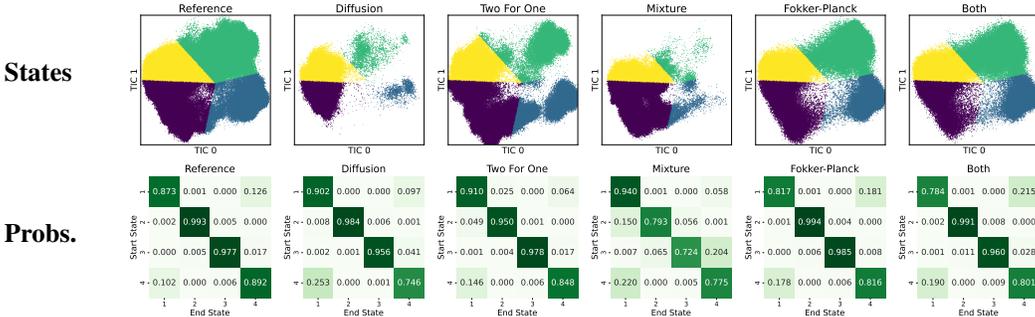


Figure 21: **BBA**. Each conformation is assigned to one of four states based on the TIC coordinates (top row), shown as State 1 (purple), State 2 (blue), State 3 (green), State 4 (yellow). Using these assignments, we estimate the corresponding transition probabilities between states.

Method	Chignolin JS ( $\downarrow$ )	BBA JS ( $\downarrow$ )
Diffusion	$3.4 \cdot 10^{-2}$	$6.8 \cdot 10^{-3}$
Two for One	$2.3 \cdot 10^{-3}$	$7.1 \cdot 10^{-3}$
Mixture	$3.0 \cdot 10^{-2}$	$4.0 \cdot 10^{-2}$
Fokker–Planck	$4.6 \cdot 10^{-4}$	$2.5 \cdot 10^{-3}$
Both	$2.1 \cdot 10^{-4}$	$4.2 \cdot 10^{-3}$

Table 16: Mean JS divergence between the transition probabilities estimated from simulation and those of the reference trajectories (compare [Figures 20 and 21](#)). Lower values indicate better agreement.

### C.7 Umbrella Sampling

Similarly to [Appendix C.6.4](#), access to an energy-based diffusion model enables further applications such as molecular simulation and *umbrella sampling* ([Torrie and Valleau, 1977](#)). In umbrella sampling, several simulations are carried out with restraints applied to specific regions of configuration space (e.g., along selected dihedral angles), and the resulting biased ensembles are reweighted by the bias potential to recover unbiased free energy estimates.

While such enhanced sampling techniques lie beyond the main scope of this work, we conducted preliminary experiments with our model. We found that, in its current form, umbrella sampling remains challenging. The model tends to perform poorly when forced into high-energy or previously unseen regions of configuration space. For example, in Chignolin, umbrella sampling along the first TIC coordinate failed to adequately explore the extremes of the coordinate, leading to unreliable free-energy estimates in those regions. Nonetheless, the overall shape of the free energy profile could still be recovered with significantly fewer simulation steps. We believe that the proposed Fokker–Planck regularization could help address this limitation. Since the loss can be evaluated for arbitrary configurations  $\mathbf{x}$ , it may encourage the model to generalize better to out-of-distribution or noisier samples, thereby improving its robustness in biased or enhanced sampling settings.

## C.8 Transferability Across Dipeptides (two amino acids)

### C.8.1 Complete Metrics

In the main text, we reported the PMF error across the test set, and in Table 17, we additionally include the JS divergence for the same experiments. Both metrics show consistent trends, although JS divergence is less sensitive to small differences, as it measures discrepancies between densities rather than energies.

Method	iid JS ( $\downarrow$ )	sim. JS ( $\downarrow$ )	iid PMF ( $\downarrow$ )	sim. PMF ( $\downarrow$ )
Transferable BG	<b>0.0183 <math>\pm</math> 0.0070</b>	-	<b>0.230 <math>\pm</math> 0.119</b>	-
Diffusion	<b>0.0155 <math>\pm</math> 0.0083</b>	0.2256 $\pm$ 0.1304	<b>0.206 <math>\pm</math> 0.159</b>	6.515 $\pm$ 3.175
Two For One	<b>0.0153 <math>\pm</math> 0.0080</b>	0.0466 $\pm$ 0.0114	<b>0.203 <math>\pm</math> 0.149</b>	0.741 $\pm$ 0.319
Mixture	<b>0.0155 <math>\pm</math> 0.0078</b>	0.0444 $\pm$ 0.0237	<b>0.200 <math>\pm</math> 0.127</b>	0.658 $\pm$ 0.407
Fokker–Planck	<b>0.0200 <math>\pm</math> 0.0071</b>	<b>0.0254 <math>\pm</math> 0.0119</b>	<b>0.241 <math>\pm</math> 0.105</b>	<b>0.368 <math>\pm</math> 0.267</b>
Both	<b>0.0157 <math>\pm</math> 0.0078</b>	<b>0.0177 <math>\pm</math> 0.0084</b>	<b>0.199 <math>\pm</math> 0.127</b>	<b>0.203 <math>\pm</math> 0.104</b>

Table 17: **Dipeptides**. Comparison of metrics across all testset dipeptides 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.

### C.8.2 Lysine-Serine (KS)

In Figure 22, we present extended results for the dipeptide analyzed in the main text (KS). The trends remain consistent with previous findings: using *Two For One* for simulation leads to incorrect observables and distorted equilibrium distributions. In addition, we show the free energy surfaces along the dihedral angles  $\varphi$  and  $\psi$  separately. The Fokker–Planck regularization improves these free energy profiles, and the *Both* model produces particularly accurate and well-aligned surfaces.

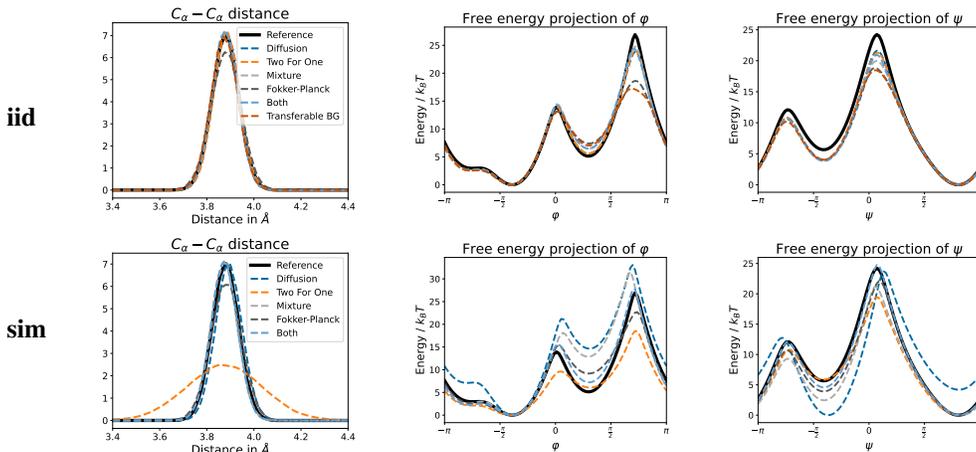


Figure 22: **KS**: We compare further metrics between iid sampling and Langevin simulation. We compare the  $C_\alpha - C_\alpha$  distance for the dipeptides and also the free energy projections along the dihedral angles  $\varphi, \psi$ .

### C.8.3 Transferability: Results on More Dipeptides

In this section, we show additional dipeptides from the test set to evaluate the performance of all models and their combinations. While individual systems exhibit minor variations, the overall trends remain consistent. We present the following dipeptides: AC Figures 23 and 24, AP Figures 25 and 26, ET Figures 27 and 28, HP Figures 29 and 30, NY Figures 31 and 32, RV Figures 33 and 34, and TD Figures 35 and 36.

Notably, in Figures 25, 29 and 35, the *Diffusion* model fails to produce stable simulations, resulting in divergence and visible artifacts. This instability appears across several systems, not only those containing proline. While we found in our experiments that introducing dropout can help to stabilize the simulation, we chose not to include it since all other methods remain stable without it. In particular, these cases highlight that restricting training to a smaller diffusion-time subrange, as is done with *Mixture*, improves simulation stability.

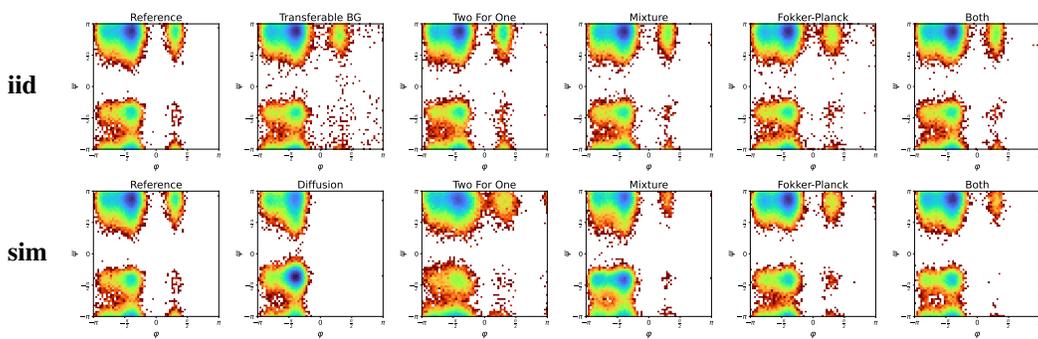


Figure 23: AC: We compare the free energy plot on the dihedral angles  $\varphi, \psi$  for all presented methods for iid sampling and Langevin simulation.

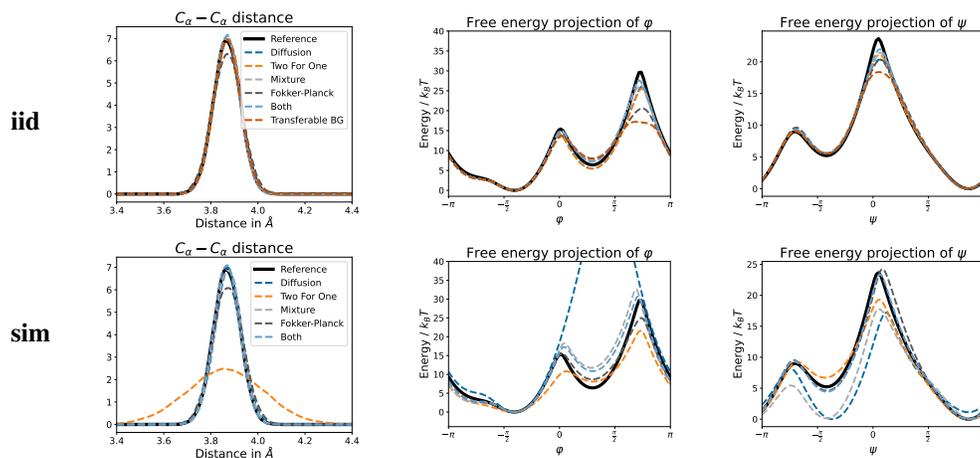


Figure 24: AC: We compare further metrics between iid sampling and Langevin simulation. We compare the  $C_\alpha - C_\alpha$  distance for the dipeptides and also the free energy projections along the dihedral angles  $\varphi, \psi$ .

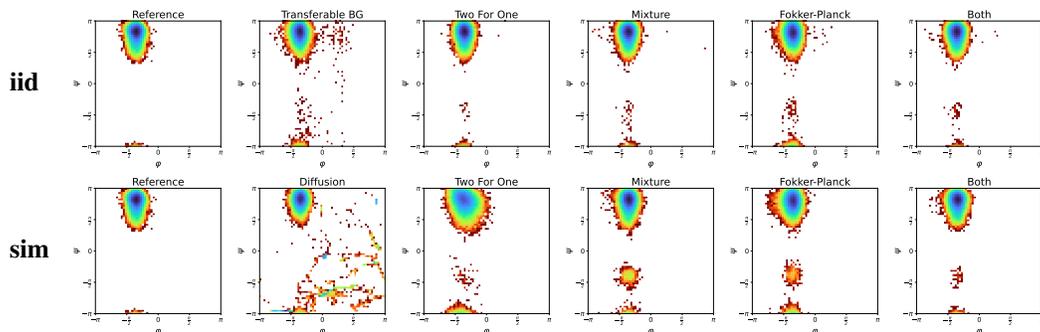


Figure 25: **AP**: We compare the free energy plot on the dihedral angles  $\varphi, \psi$  for all presented methods for iid sampling and Langevin simulation.

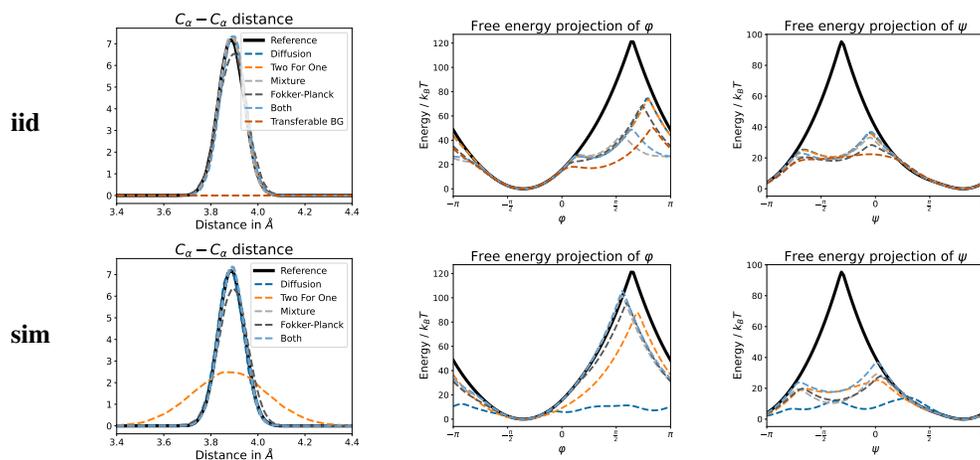


Figure 26: **AP**: We compare further metrics between iid sampling and Langevin simulation. We compare the  $C_{\alpha}-C_{\alpha}$  distance for the dipeptides and also the free energy projections along the dihedral angles  $\varphi, \psi$ .

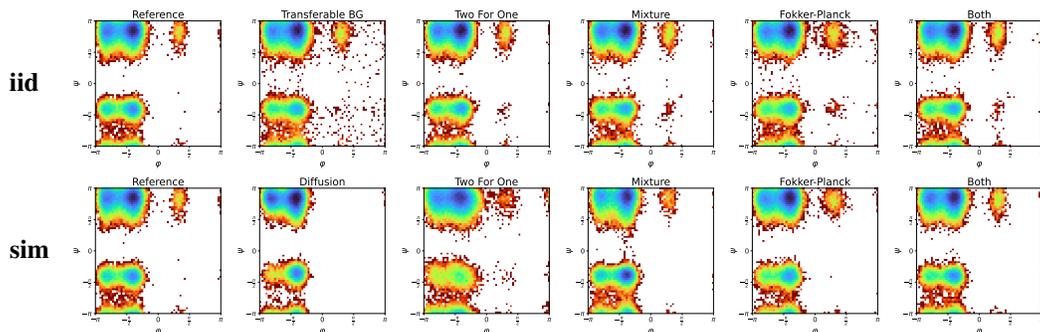


Figure 27: **ET**: We compare the free energy plot on the dihedral angles  $\varphi, \psi$  for all presented methods for iid sampling and Langevin simulation.

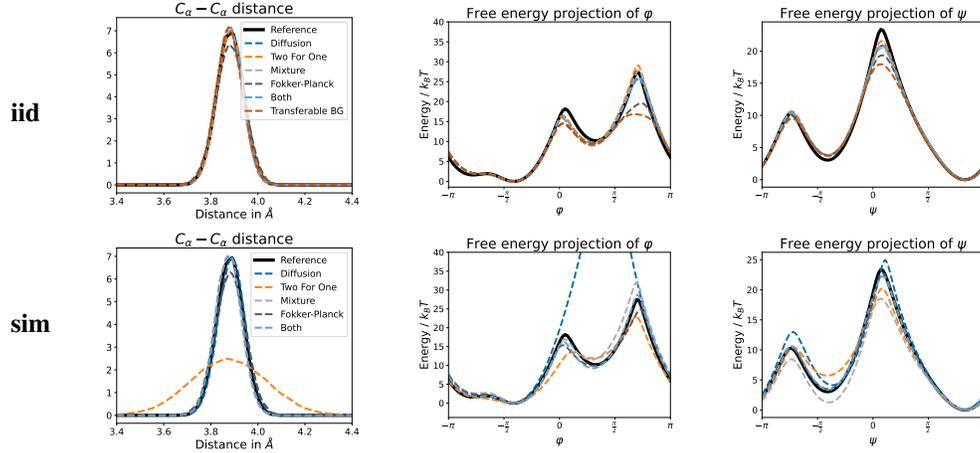


Figure 28: **ET**: We compare further metrics between iid sampling and Langevin simulation. We compare the  $C_{\alpha}-C_{\alpha}$  distance for the dipeptides and also the free energy projections along the dihedral angles  $\varphi, \psi$ .

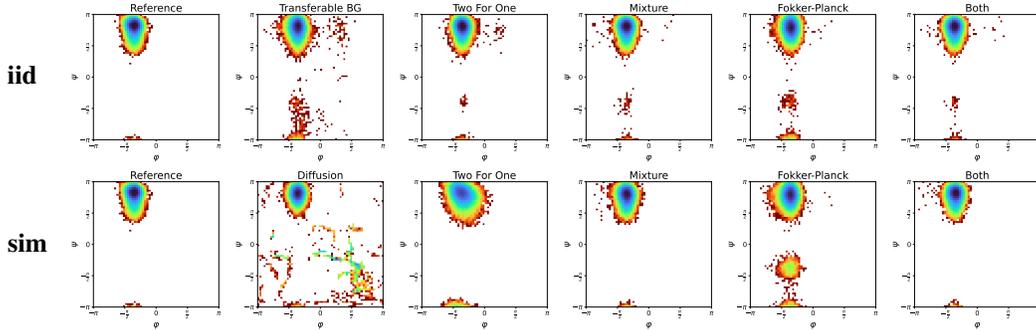


Figure 29: **HP**: We compare the free energy plot on the dihedral angles  $\varphi, \psi$  for all presented methods for iid sampling and Langevin simulation.

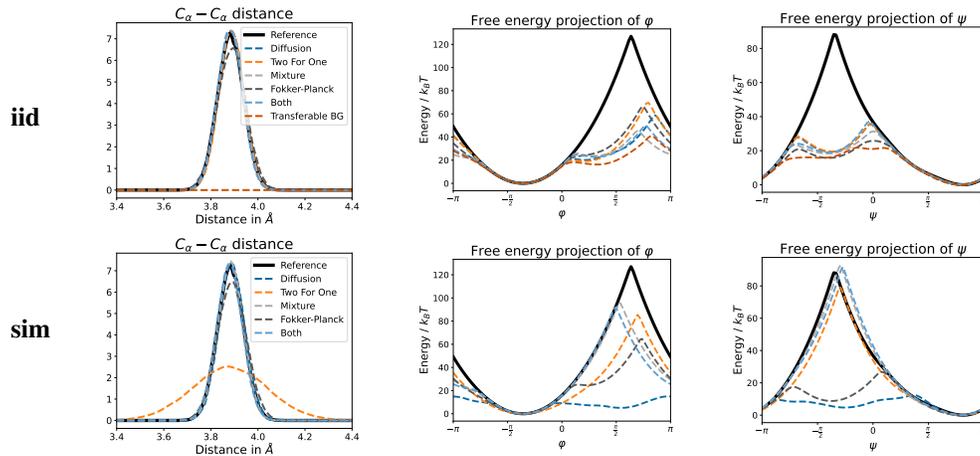


Figure 30: **HP**: We compare further metrics between iid sampling and Langevin simulation. We compare the  $C_{\alpha}-C_{\alpha}$  distance for the dipeptides and also the free energy projections along the dihedral angles  $\varphi, \psi$ .

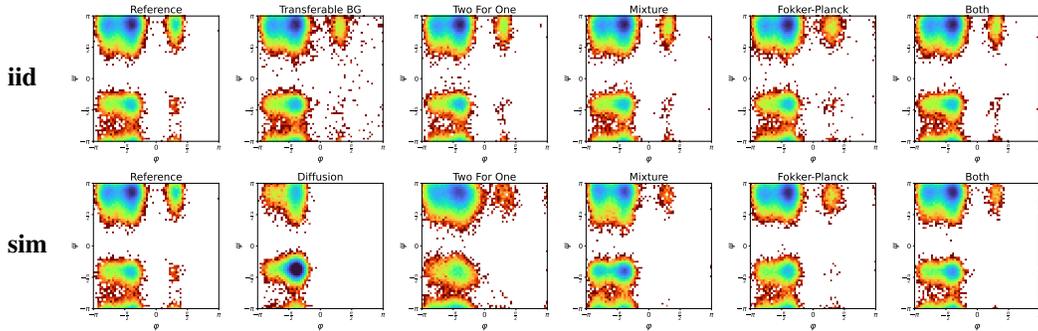


Figure 31: NY: We compare the free energy plot on the dihedral angles  $\varphi, \psi$  for all presented methods for iid sampling and Langevin simulation.

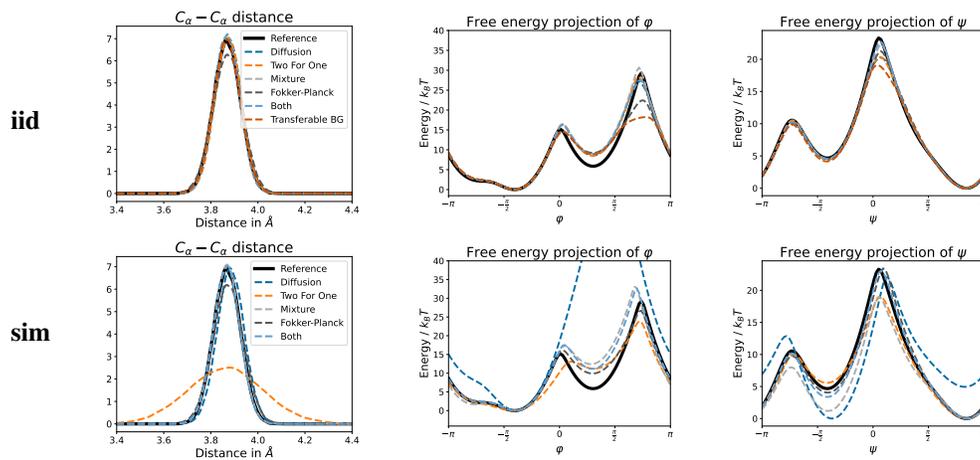


Figure 32: NY: We compare further metrics between iid sampling and Langevin simulation. We compare the  $C_\alpha - C_\alpha$  distance for the dipeptides and also the free energy projections along the dihedral angles  $\varphi, \psi$ .

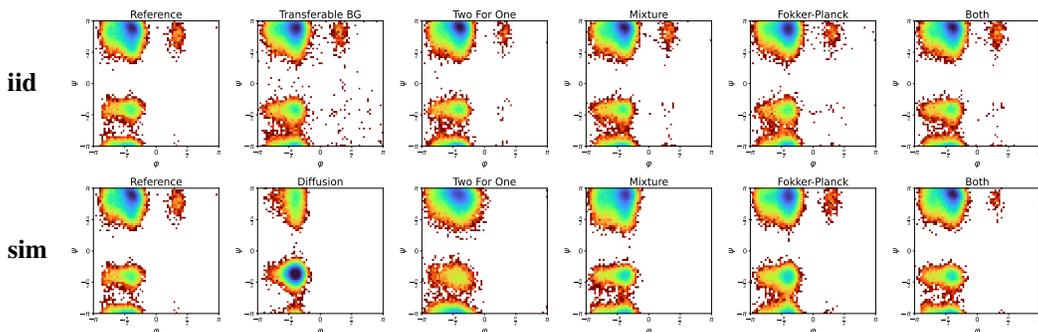


Figure 33: RV: We compare the free energy plot on the dihedral angles  $\varphi, \psi$  for all presented methods for iid sampling and Langevin simulation.

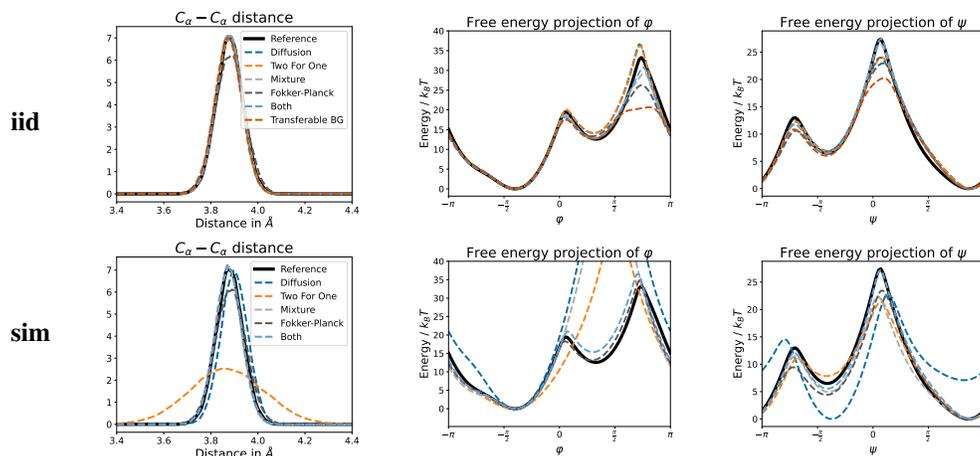


Figure 34: **RV**: We compare further metrics between iid sampling and Langevin simulation. We compare the  $C_\alpha - C_\alpha$  distance for the dipeptides and also the free energy projections along the dihedral angles  $\varphi, \psi$ .

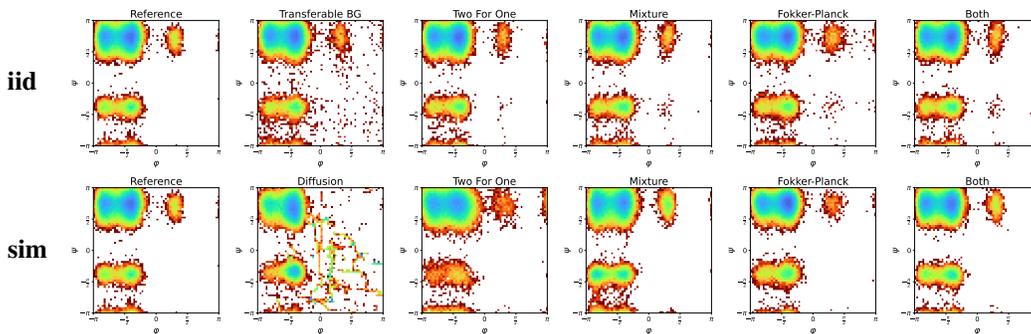


Figure 35: **TD**: We compare the free energy plot on the dihedral angles  $\varphi, \psi$  for all presented methods for iid sampling and Langevin simulation.

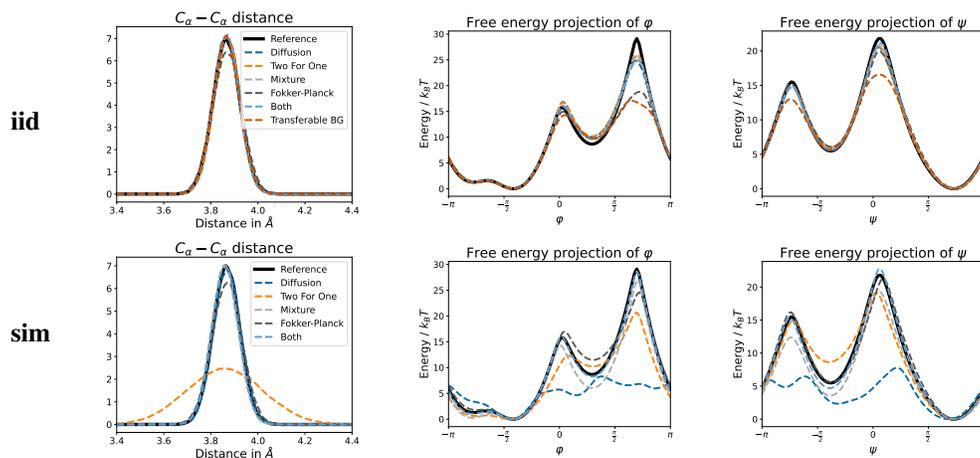


Figure 36: **TD**: We compare further metrics between iid sampling and Langevin simulation. We compare the  $C_\alpha - C_\alpha$  distance for the dipeptides and also the free energy projections along the dihedral angles  $\varphi, \psi$ .

## **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 does not provide formal guarantees, which poses a risk of misleading downstream research if the method produces incorrect or biased results.

## NeurIPS Paper Checklist

### 1. Claims

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

Answer: [Yes]

Justification: The major claims made in the abstract and introduction are backed up by the theoretical and empirical results in the paper. Refer to [Section 3](#) and [Section 5](#).

Guidelines:

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

### 2. Limitations

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

Answer: [Yes]

Justification: We have discussed potential limitations of our work in [Section 6](#).

Guidelines:

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

### 3. Theory assumptions and proofs

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

Answer: [Yes]

Justification: We have discussed all assumptions of our method in [Section 3](#) and have added proofs and mathematical details in [Appendix A](#).

Guidelines:

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

#### 4. Experimental result reproducibility

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

Answer: [\[Yes\]](#)

Justification: To the best of our knowledge, we have disclosed all information necessary to reproduce the results in the main paper. Additionally, we make our code available at <https://github.com/noegroup/ScoreMD>, which can be used to infer more details and reproduce results.

Guidelines:

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

## 5. Open access to data and code

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

Answer: [Yes]

Justification: We have used synthetically generated data that we explain in detail how to produce (compare [Appendix C.4](#)), publicly available datasets (compare [Appendices B.3](#) and [B.5](#)), publicly available libraries (compare [Appendix B.7](#)) and make our code available at <https://github.com/noegroup/ScoreMD>.

Guidelines:

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

## 6. Experimental setting/details

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

Answer: [Yes]

Justification: We provide all technical details for the experiments either in the main text in [Section 5](#) or in [Appendix B](#). Further, we make our code available at <https://github.com/noegroup/ScoreMD> which contains all settings and hyperparameters.

Guidelines:

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

## 7. Experiment statistical significance

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

Answer: [Yes]

Justification: We report the mean and standard deviation in the experiments where suitable (in the main text compare [Table 1](#) and [Table 2](#)).

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments compute resources

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

Answer: [Yes]

Justification: We list the computational resources we used for the experiments in [Appendix B.6](#) and the runtime in [Appendix C.3](#).

Guidelines:

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

## 9. Code of ethics

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

Answer: [Yes]

Justification: We believe that our research adheres to the NeurIPS Code of Ethics.

Guidelines:

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

## 10. Broader impacts

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

Answer: [Yes]

Justification: We have discussed societal impacts in [Appendix D](#).

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

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

Answer: [NA]

Justification: We do not believe that any safe guards needs to be in place, as our approach does not pose any direct risks. As for the data, only widely used data and commonly used resources have been used and, hence, does not pose a risk for misuse.

Guidelines:

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

## 12. Licenses for existing assets

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

Answer: [Yes]

Justification: We have properly credited all works, for the datasets compare [Appendices B.3](#) and [B.5](#) and software licenses are listed in [Appendix B.7](#).

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. **New assets**

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

Answer: [Yes]

Justification: We make it clear in [Section 5](#) that our code is publicly available at <https://github.com/noegroup/ScoreMD> and provide information in the README on how to use the code.

Guidelines:

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

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

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

Answer: [NA]

Justification: No crowdsourcing or human subjects are involved in this research.

Guidelines:

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

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

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

Answer: [NA]

Justification: No human subjects are involved in this research.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

**16. Declaration of LLM usage**

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

Answer: [NA]

Justification: LLMs have been used to implement standard methods, and for writing, editing, and formatting purposes of this document and hence no further declaration is required.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.