BOLTZMANN PRIORS FOR IMPLICIT TRANSFER OPERATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurate prediction of thermodynamic properties is essential in drug discovery and materials science. Molecular dynamics (MD) simulations provide a principled approach to this task, yet they typically rely on prohibitively long sequential simulations. Implicit Transfer Operator (ITO) Learning offers a promising approach to address this limitation by enabling stable simulation with time steps orders of magnitude larger than MD. However, to train ITOs, we need extensive, unbiased MD data, limiting the scope of this framework. Here, we introduce Boltzmann Priors for ITO (BoPITO) to enhance ITO learning in two ways. First, BoPITO enables more efficient data generation, and second, it embeds inductive biases for long-term dynamical behavior, simultaneously improving sample efficiency by one order of magnitude and guaranteeing asymptotically unbiased equilibrium statistics. Further, we showcase the use of BoPITO in a new tunable sampling protocol interpolating ITO models trained on off-equilibrium simulation data and an unbiased equilibrium distribution to solve inverse problems in molecular science.

1 INTRODUCTION

Efficient molecular dynamics (MD) simulation on long time-scales is critical to a large number of scientific and engineering applications. Stable simulations rely on solvers taking tiny integration time-steps, making the simulation of most phenomena impractical with current methods. Since these simulations are stochastic, a simulation step corresponds to drawing a sample from a transition probability density $p(\mathbf{x}_{t+\tau} \mid \mathbf{x}_t)$, where τ is a tiny time-step. Recently, deep generative models have emerged as a promising strategy to potentially speed up these simulations by learning transition probability densities where τ is much larger (Schreiner et al., 2023; Klein et al., 2023; Hsu et al., 2024; Fu et al., 2023) and thereby allow efficient sampling at long time-scales. Implicit Transfer Operator (ITO) learning (Schreiner et al., 2023) learns such surrogate models at multiple time-resolutions. While ITO has shown promise in accelerating simulations, it relies on extensive unbiased simulation data which may not always be available. In Markov state models, this limitation can be mitigated by integrating off-equilibrium simulations and enhanced sampling simulations (Trendelkamp-Schroer et al., 2015; Rosta & Hummer, 2014; Wu et al., 2014; 2016). However, such estimators are so far unavailable for deep generative surrogates of the transition density.

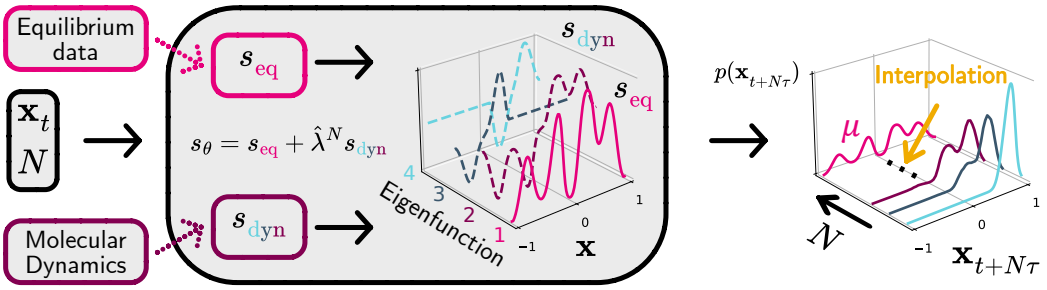


Figure 1: **Boltzmann Priors for Implicit Transfer Operators (BoPITO)** leverage pre-trained Boltzmann Generators to enable data-efficient training of surrogate models of the transition density after N simulation steps. BoPITO is implemented using Score-Based Diffusion Models. The score of the model, s_θ, separates the contributions from the first eigenfunction of the transfer operator (equilibrium density), s_{eq}, from the rest, s_{dyn}. BoPITO embeds inductive biases for long-term dynamical behavior and enables interpolation between off-equilibrium and equilibrium models.

Here, we introduce Boltzmann Priors for Implicit Transfer Operator (BoPITO) learning (Figure 1). BoPITO leverages pre-trained Boltzmann Generators (BG) (Noé et al., 2019; Viguera Diez et al., 2024; Klein & Noé, 2024; Köhler et al., 2023; Köhler et al., 2020; Midgley et al., 2024; 2023) as priors to enable data-efficient training of ITO models, leading to one order of magnitude reduction in the simulation data needed for training. As the BG encodes the *invariant measure* or Boltzmann distribution of the dynamics encoded by the transition density, BoPITO, by construction, guarantees asymptotically unbiased equilibrium statistics. In this way, BoPITO can combine off-equilibrium data and biased data encoded into a BG prior to train ITO models that predict MD across multiple time-scales. Using BoPITO we introduce a new sampling strategy to recover approximate dynamics from biased off-equilibrium data, a BG prior, and unbiased time-correlation data, providing a new method for inverse problems for molecular systems.

Our main contributions are

1. **Boltzmann Priors for Implicit Transfer Operator Learning (BoPITO)**: We provide a principled way to leverage Boltzmann Generators as priors for Implicit Transfer Operator Learning, boosting sample efficiency, ensuring the generation of uncorrelated data and allowing asymptotically unbiased equilibrium statistics for $\tau \to \infty$.
2. **BoPITO interpolators**: A tunable sampling protocol facilitating interpolation of BoPITO models trained on off-equilibrium simulation data to an unbiased equilibrium distribution.
3. We show that BoPITO interpolators can **recover approximate dynamics from models trained on biased simulations**. We frame the optimization of interpolation parameters as an inverse problem and **select the interpolated ensemble that is most consistent with unbiased observables**.

2 Background and preliminaries

2.1 Molecular dynamics and observables

Molecular dynamics (MD) is a widely used simulation method in chemistry, physics, and biology. It combines a mathematical model of the dynamics — e.g. Langevin dynamics (Langevin, 1908) — with a potential energy model, $U(\mathbf{x}) : \Omega \to \mathbb{R}$, of a system of interest, in turn providing detailed mechanistic insights of molecular systems, through the time evolution of particles in configuration space, $\mathbf{x} \in \Omega$. Practically, MD is solved by numerical integration, and time is discretized, with step τ. Consequently, we can understand MD as a Markov process with a Normal transition density $p(\mathbf{x}_{t+\tau} \mid \mathbf{x}_t)$, whose associated Markov operator has the *Boltzmann distribution*,

$$\mu(\mathbf{x}) = \mathcal{Z}^{-1} \exp(-\beta U(\mathbf{x})), \text{ with } \mathcal{Z} = \int \mathrm{d}\mathbf{x} \, \exp(-\beta U(\mathbf{x})), \tag{1}$$

as its *invariant measure* or stationary distribution, where β is the inverse temperature.

One important application of MD is to compute expectations or *observables* (Olsson, 2022):

1. **Stationary observables**:
$$O_a = \mathbb{E}_\mu \left[a(\mathbf{x}) \right]. \tag{2}$$
2. **Dynamic observables / Time correlation functions**:
$$O_{a(t),b(t+N\tau)} = \mathbb{E}_{\mathbf{x}_t \sim \mu} \left[\mathbb{E}_{\mathbf{x}_{t+N\tau} \sim p_\tau(\mathbf{x}_{t+N\tau} | \mathbf{x}_t)} \left[a(\mathbf{x}_t) \cdot b(\mathbf{x}_{t+N\tau}) \right] \right], \tag{3}$$

where $p_\tau(\mathbf{x}_{t+N\tau} | \mathbf{x}_t)$ is the conditional probability density after N simulation steps with time-step τ. The maps a and $b : \Omega \to \mathbb{R}^L$ serve as observable functions or *'forward models'* characterizing microscopic observation processes, e.g. indicating whether a drug is bound or not, or an interatomic distance, leading to observables such binding affinities and off-rates of a drug to a target protein.

Unfortunately, stable MD simulations rely on using time discretizations, τ, on the order of 10^{-15} s, whereas properties such as protein folding or ligand unbinding occur on much longer time scales, $\sim 10^{-3} - 10^{-1}$ s. Due to the sequential nature of these simulations, an impractical number of simulation steps are necessary to predict these properties in an unbiased fashion. As a result, most MD data will be *'off-equilibrium'*, e.g. trajectories exploring one or a few of the modes of the Boltzmann distribution $\mu(\mathbf{x})$. See Appendix A.1 for more precise definitions of configuration space, off-equilibrium, unbiased and biased MD.

2.2 BOLTZMANN GENERATORS

Boltzmann Generators (BG) (Noé et al., 2019) are a generative machine learning approach to draw samples i.i.d. from the Boltzmann distribution, $\mu(\mathbf{x})$, of physical many-body systems (eq. 1). BGs learn a diffeomorphic map, $\mathcal{F}_{\boldsymbol{\theta}} : \mathbb{R}^N \to \mathbb{R}^N$, from a latent space, equipped with a simple base density $p(\mathbf{z})$, to configurational space of a physical system, such that the push-forward density $\hat{p}(\mathbf{x}) = \mathcal{F}_{\boldsymbol{\theta}} \# p(\mathbf{z})$ closely approximates the Boltzmann distribution (eq. 1). In practice, $\mathcal{F}_{\boldsymbol{\theta}}$ is implemented using an invertible neural network architecture with tractable Jacobian determinants to enable efficient sampling and exact sample likelihood computation (Chen et al., 2018; Papamakarios et al., 2021). BGs are trained either with approximately equilibrated simulation data or with biased simulations, from i.e. enhanced sampling, by employing appropriate reweighting (Ferrenberg & Swendsen, 1989; Shirts & Chodera, 2008). Unbiased samples can then be generated by importance re-sampling (Nicoli et al., 2020). Similarly, unbiased expectations can be computed using the importance weights $w(\mathbf{x}) = e^{-\beta U(\mathbf{x})}/\hat{p}(\mathbf{x})$. As such, BGs are *surrogates* of the target Boltzmann distribution, but do not model time-correlation statistics.

2.3 TRANSFER OPERATORS

Transfer operators (Ruelle, 1978; Schütte et al., 2009) provide a framework to describe the evolution of probability densities over time. Let p denote an initial probability density function on Ω, and ρ its μ-weighted version, $p = \mu\rho$. The Markov operator, T_Ω, is defined using a transition density $p(\mathbf{x}_{t+\tau}|\mathbf{x}_t)$,

$$[T_\Omega(\tau) \circ \rho](\mathbf{x}_{t+\tau}) \equiv \frac{1}{\mu(\mathbf{x}_{t+\tau})} \int_\Omega \mu(\mathbf{x}_t)\rho(\mathbf{x}_t)p(\mathbf{x}_{t+\tau}|\mathbf{x}_t)\,\mathrm{d}\mathbf{x}_t. \tag{4}$$

This operator describes the μ-weighted evolution of absolutely convergent probability density functions on Ω by a discrete-time increment τ given by the dynamics encoded in a transition density $p(\mathbf{x}_{t+\tau}|\mathbf{x}_t)$. In the context of molecular dynamics, T_Ω is a μ-weighted equivalent to the Markov operator discussed above (sec. 2.1). The spectral form of the transfer operator is expressed as

$$[T_\Omega(\tau) \circ \rho](\mathbf{x}_{t+\tau}) = \sum_{i=1}^{\infty} \lambda_i(\tau)\langle\rho|\phi_i\rangle\ \psi_i(\mathbf{x}_{t+\tau}), \tag{5}$$

where $\lambda_i(\tau)$ are the eigenvalues, ψ_i and ϕ_i are the corresponding right and left eigenfunctions, respectively, and $\langle f|g\rangle = \int_\Omega f(\mathbf{x})g(\mathbf{x})\,\mathrm{d}\mathbf{x}$. The eigenvalues $\lambda_i(\tau)$ depend on the parameter τ and are related to the characteristic relaxation rates κ_i by $|\lambda_i(\tau)| = \exp(-\tau\kappa_i)$. The right and left eigenfunctions are related by the stationary density, such that $\phi_i(\mathbf{x}) = \mu(\mathbf{x})\psi_i(\mathbf{x})$. In reversible dynamics, all eigenvalues λ_i are real and lie within the interval $-1 < \lambda_i \leq 1$. Notably, there is one eigenvalue $\lambda_1 = 1$, with corresponding eigenfunctions $\psi_1(\mathbf{x}) = \mathbf{1}$ and $\phi_1(\mathbf{x}) = \mu(\mathbf{x})$.

2.4 DIFFUSION MODELS

Diffusion models (DMs) (Ho et al., 2020; Song et al., 2020) are a popular generative modeling framework that approximates data densities $p(\mathbf{x}^0)$ by learning to invert a noising process (*'forward diffusion'*). The forward diffusion process is pre-specified and incrementally transforms the data distribution into a simple prior distribution, $p(\mathbf{x}^T)$, through simulation of a time-inhomogenous Markov process represented by the SDE,

$$\mathrm{d}\mathbf{x}^{t_{\mathrm{diff}}} = f(\mathbf{x}^{t_{\mathrm{diff}}}, t_{\mathrm{diff}})\,\mathrm{d}t_{\mathrm{diff}} + g(t_{\mathrm{diff}})\,\mathrm{d}W, \tag{6}$$

where $(0 < t_{diff} < T)$ is the diffusion time, f and g are chosen functions, and $\mathrm{d}W$ is a Wiener process. To generate samples from the data distribution, $p(x^0)$, we can sample from $p(\mathbf{x}^T)$ and solve the backward diffusion process (*'denoising process'*) (Anderson, 1982),

$$\mathrm{d}\mathbf{x}^{t_{\mathrm{diff}}} = \left[f(\mathbf{x}^{t_{\mathrm{diff}}}, t_{\mathrm{diff}}) - g^2(t_{\mathrm{diff}})\nabla_{\mathbf{x}^{t_{\mathrm{diff}}}} \log p(\mathbf{x}^{t_{\mathrm{diff}}}|t_{\mathrm{diff}})\right]\mathrm{d}t_{\mathrm{diff}} + g(t_{\mathrm{diff}})\,\mathrm{d}W. \tag{7}$$

This backward process is approximated using a deep neural network $\nabla_{\mathbf{x}^{t_{\mathrm{diff}}}} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{t_{\mathrm{diff}}}|t_{\mathrm{diff}}) = s_{\boldsymbol{\theta}}(\mathbf{x}^{t_{\mathrm{diff}}}, t_{\mathrm{diff}})$. The learned backward SDE can be recast into a probability flow ODE (Maoutsa et al., 2020; Song et al., 2021), which in turn can be interpreted as a continuous normalizing flow (Chen et al., 2018) which facilitates efficient sampling and enables sample likelihood evaluation.

2.5 Implicit Transfer Operator Learning

Implicit Transfer Operator (ITO) Learning (Schreiner et al., 2023) is a framework for learning surrogate models of the transition density, $p(\mathbf{x}_{N\tau}|\mathbf{x}_0)$, from MD data, where N is an arbitrarily large integer. ITO leverages that the transfer operator framework allows us to express the transition density as

$$p(\mathbf{x}_{t+N\tau}|\mathbf{x}_t) = \sum_{i=1}^{\infty} \lambda_i^N(\tau)\psi_i(\mathbf{x}_t)\phi_i(\mathbf{x}_{t+N\tau}), \tag{8}$$

where ψ_i and ϕ_i are independent of τ and N. See Appendix A.2 for a complete derivation.

This decomposition inspired a strategy to learn a conditional generative model $\mathbf{x}_{t+N\tau} \sim p_{\boldsymbol{\theta}}(\mathbf{x}_{t+N\tau}|\mathbf{x}_t, N)$ by sampling tuples $(\mathbf{x}_{t_i}, \mathbf{x}_{t_i+N_i\tau}, N_i)$ and training a generative model to mimic the empirical transition density at multiple time-horizons, $N\tau$. Here, we approximate ITO models with a conditional denoising diffusion probabilistic model (cDDPM) of the form

$$p(\mathbf{x}^0_{t+N\tau}|\mathbf{x}_t, N) \equiv \int p(\mathbf{x}^{0:T}_{t+N\tau}|\mathbf{x}_t, N)\, \mathrm{d}\mathbf{x}^{1:T}, \tag{9}$$

where $\mathbf{x}_{1:T}$ are latent variables of the same dimension as our output, and follow a joint density describing the backward diffusion process, eq. 7, and $\mathbf{x}^T \sim \mathcal{N}(0, \mathbb{I})$. We denote diffusion time in Diffusion Models using superscripts, while physical time is represented using subscripts. The conditional sample likelihood is given by

$$\ell(\mathbf{I}; \boldsymbol{\theta}) \equiv \prod_{i \in \mathbf{I}} p_{\boldsymbol{\theta}}(\mathbf{x}^0_{t_i+N_i\tau} \mid \mathbf{x}_{t_i}, N_i) \tag{10}$$

where $\mathbf{I}$ is a list of generated indices i specifying a time t_i and a time-lag (τ) integer multiple N_i, associating two time-points in a MD trajectory of length $M\tau$, $\mathbf{x} = \{\mathbf{x}_0, \mathbf{x}_\tau, \ldots, \mathbf{x}_{(M-1)\tau}\}$.

Training is performed by optimizing an approximation of the variational bound of the log-likelihood (Ho et al., 2020),

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{i\sim\mathbf{I},\epsilon\sim\mathcal{N}(0,\mathbb{I}),t_{\text{diff}}\sim\mathcal{U}(0,T)} \left[\|\epsilon - \hat{\epsilon}_{\boldsymbol{\theta}}(\widetilde{\mathbf{x}}^{t_{\text{diff}}}_{t_i+N_i\tau}, \mathbf{x}_{t_i}, N_i, t_{\text{diff}})\|_2\right], \tag{11}$$

where $\widetilde{\mathbf{x}}^{t_{\text{diff}}}_t = \sqrt{\bar{\alpha}^{t_{\text{diff}}}}\mathbf{x}_t + \sqrt{1-\bar{\alpha}^{t_{\text{diff}}}}\epsilon$, with $\bar{\alpha}^{t_{\text{diff}}} = \prod_j^{t_{\text{diff}}}(1-\beta_j)$ and β_j is the variance of the forward diffusion process at diffusion time, j. $\hat{\epsilon}_{\boldsymbol{\theta}}(\cdot)$ is one of the two architectures presented by Schreiner et al. (2023). Following Arts et al. (2023) we express the score as

$$s_{\boldsymbol{\theta}}(\mathbf{x}^{t_{\text{diff}}}_{t+N\tau}, \mathbf{x}_t, N_i, t_{\text{diff}}) = -\frac{\hat{\epsilon}_{\boldsymbol{\theta}}(\widetilde{\mathbf{x}}^{t_{\text{diff}}}_{t_i+N_i\tau}, \mathbf{x}_{t_i}, N_i, t_{\text{diff}})}{\sqrt{1-\bar{\alpha}^{t_{\text{diff}}}}}. \tag{12}$$

See Appendix A.3 for more details and pseudo-code on training and sampling algorithms.

3 Boltzmann Priors for Implicit Transfer Operator Learning

ITO training requires extensive, unbiased MD simulations to capture the statistical distribution of rare events. This data-intensive requirement can hinder ITO's practical implementation. Furthermore, models trained on off-equilibrium simulations, which may exhibit non-representative statistics, can lead to inaccurate predictions, compromising their utility in downstream applications. In practice, however, both unbiased off-equilibrium simulations and information about the equilibrium distribution from potentially biased simulations (Hénin et al., 2022) can be cheaply generated, and both encode information about the dynamical behavior of molecular systems. We introduce Boltzmann Priors for Implicit Transfer Operator (BoPITO), as a learning paradigm for ITO models that leverage available information about the equilibrium distribution.

BoPITO uses pre-trained models of the equilibrium distribution to improve ITO learning in four ways. First, it helps ensure broad sampling of Ω proportional to the equilibrium distribution for subsequent MD simulations yielding information about the transition density $p(\mathbf{x}_\tau \mid \mathbf{x}_0)$ from across Ω. Second, we use it to fix the stationary part of the learned transition density, boosting the sample efficiency when learning models of molecular dynamics. Third, it imposes an inductive bias of long-time dynamical behavior allowing for recovery of unbiased Boltzmann distribution for long-time horizons. Fourth, using BoPITO, we introduce a novel tunable sampling protocol interpolating ITO models trained on off-equilibrium simulation data and an unbiased equilibrium distribution.

3.1 EFFICIENT DATA GENERATION

Training data for ITO models consists of several independent unbiased MD simulations. Following the adaptive sampling strategy (Bowman et al., 2010; Doerr & De Fabritiis, 2014; Viguera Diez et al., 2024; Betz & Dror, 2019), used extensively in the molecular dynamics simulation community, we use a pre-trained BG to generate initial conditions to simulations ensuring broad sampling across Ω proportional to the Boltzmann distribution. As long as these trajectories reach a 'local equilibrium' we can in principle recover an unbiased model of the molecular dynamics (Nüske et al., 2017), albeit without relying on running one or a few very long simulations to reach the global equilibrium.

3.2 LONG-TERM DYNAMICS INDUCTIVE BIAS FOR ITO

One important application of ITO is to allow for one-step sampling of long-time-scale dynamics. However, real datasets often contain a limited number of effective samples for long time-scales, leading to potential biases in models. To mitigate this issue, we propose separating the equilibrium contribution from the time-dependent components:

$$p(\mathbf{x}_{t+N\tau}|\mathbf{x}_t) = \mu(\mathbf{x}_{t+N\tau}) + \sum_{i=2}^{\infty} \lambda_i^N(\tau)\phi_i(\mathbf{x}_{t+N\tau})\psi_i(\mathbf{x}_t), \tag{13}$$

where we have used that $\lambda_1 = 1$ and its corresponding eigenfunctions are $\phi_1 = \mu$ and $\psi_1 = \mathbf{1}$. Additionally, we introduce a decay in the time-dependent component, creating an inductive bias that asymptotically samples from an available equilibrium model for long-term dynamics. Using the spectral decomposition of the transition density (eq. 13), we choose the score model as

$$s(\mathbf{x}_{t+N\tau}^{t_{\text{diff}}}, \mathbf{x}_t, N, t_{\text{diff}}, \boldsymbol{\theta}) = s_{\text{eq}}(\mathbf{x}_{t+N\tau}^{t_{\text{diff}}}, t_{\text{diff}}) + \hat{\lambda}^N s_{\text{dyn}}(\mathbf{x}_{t+N\tau}^{t_{\text{diff}}}, \mathbf{x}_t, N, t_{\text{diff}}, \boldsymbol{\theta}), \tag{14}$$

where $s_{\text{eq}}(\mathbf{x}_{\text{diff}}^t, t_{\text{diff}})$ is the score of a pre-trained surrogate of the equilibrium distribution model, $0 < \hat{\lambda} < 1$ is a hyper-parameter and $s_{\text{dyn}}(\mathbf{x}_{t+N\tau}^{t_{\text{diff}}}, \mathbf{x}_t, N, t_{\text{diff}}, \boldsymbol{\theta})$ accounts for the time-dependent components. As $N \to \infty$, s_{eq} dominates and the model samples from the equilibrium model, see Appendix A.4 for an example. By interpreting the score field as a velocity field, we can reformulate the DM as a continuous normalizing flow which we can use for Metropolized sampling of the unbiased equilibrium distribution (Klein et al., 2023). The proposed factorization is principled, corresponding to a separation of the score of the transition density into a stationary and dynamic part, where the first part is considered known, see Appendix A.5 for details. In practice, we first train s_{eq} (if not provided) using equilibrium data. Then we train s_{dyn} with unbiased, possibly off-equilibrium, MD data while keeping s_{eq} fixed. We discuss alternative formulations of the scooore in Appendix A.6.

3.3 BOPITO INTERPOLATORS

A significant issue in MD simulations is the unbiased sampling of transitions over high free-energy barriers, e.g. channels in $\mu(\mathbf{x})$ with very little probability mass, typically coinciding with mixing on Ω. This difficulty leads to an inability to predict time-correlation statistics for large $N\tau$. With BoPITO we can define an interpolation between models trained on off-equilibrium simulations and the equilibrium distribution. In this manner, we can approximate dynamics not seen explicitly in the data. For a time-dependent model, s_{dyn}, trained on off-equilibrium simulations with maximum training lag, N_{max}, we define a BoPITO interpolator as a model with the following score function:

$$s(\mathbf{x}^{t_{\text{diff}}}, \mathbf{x}_t, N_{\text{int}}, t_{\text{diff}}) = s_{\text{eq}}(\mathbf{x}^{t_{\text{diff}}}, t_{\text{diff}}) + \hat{\lambda}^{N_{\text{int}}} s_{\text{dyn}}(\mathbf{x}^{t_{\text{diff}}}, \mathbf{x}_t, N_{\text{max}}, t_{\text{diff}}), \tag{15}$$

with $N_{\text{int}} \geq N_{\text{max}}$. This defines an interpolator because $N_{\text{int}} = N_{\text{max}}$ generates samples from the model distribution with maximum lag and $N_{\text{int}} \to \infty$ generates samples from the equilibrium model.

Inspired by methods in molecular biophysics (Kolloff & Olsson, 2023; Olsson et al., 2017; Bottaro & Lindorff-Larsen, 2018; Salvi et al., 2016), we propose to choose the interpolation parameter, N_{int}, as the most consistent with an unbiased dynamical observable, such as experimental data. That is, for a lag $N > N_{\text{max}}$ and an unbiased dynamic observable O_N^*, we choose

$$N_{\text{int},N} = \arg\min_{N_{\text{int}}} |O_N^* - O_{N_{\text{int}}}|, \tag{16}$$

where $O_{N_{\text{int}}}$ is the dynamic observable estimated with samples from the BoPITO interpolator. This way BoPITO can integrate off-equilibrium and equilibrium MD with experimental data. In practice,

we find the interpolator to generate some high-energy states. However, we can alleviate the high-energy structures by alternating long-lag interpolation steps with short-lag non-interpolation steps for local relaxation. We discuss how BoPITO interpolators alleviate out-of-distribution issues in Appendix A.7.

4 Results

For detailed parameters of the experiments below, we refer to Appendix A.8.

4.1 Systems

Prinz potential is a 1D potential commonly used for benchmarking MD sampling methods (Prinz et al., 2011). We set the observable functions, a and b, eq. 3, to be the identity function for computing dynamic observables. For details, see Appendix A.9.

Alanine Dipeptide is a small peptide with 22 atoms. We use publicly available data from Dibak et al. (2022), containing 1 μs simulation time split in 20 trajectories. Simulation is performed in an implicit solvent with 2 fs integration time-step, and data is saved every 1 ps. We choose,

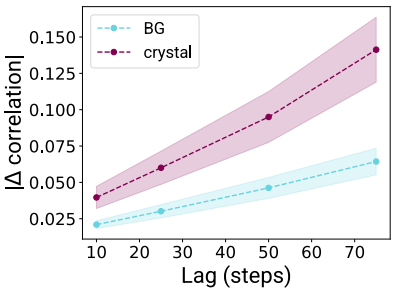
$$a(\mathbf{x}) = b(\mathbf{x}) = \begin{bmatrix} \sin\phi(\mathbf{x}) \\ \cos\phi(\mathbf{x}) \end{bmatrix}, \tag{17}$$

where $\phi(\mathbf{x})$ is a torsion angle involved in the slowest transition in the system. For details, see Appendix A.10.

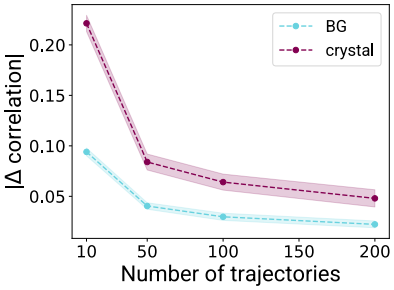
Chignolin (cln025) is a fast folding protein with 10 residues, 166 atoms and 93 heavy atoms. We use molecular dynamics data previously reported by Lindorff-Larsen et al. (2011). The data is proprietary but available upon request for research purposes. The simulations were performed in explicit solvent with a 2.5 fs time-step and the positions was saved at 200 ps intervals. We extract all heavy atoms positions from the simulations and train models on this data. We use the fraction of native contacts Lindorff-Larsen et al. (2011) to define a dynamic observable with,

$$a(\mathbf{x}) = b(\mathbf{x}) = \frac{\sum_{i=1}^{N_{\text{res}}} \sum_{j>i}^{N_i} \frac{1}{1+e^{10(d_{ij}(\mathbf{x}) - d^*_{ij} - 1)}}}{\sum_{i=1}^{N_{\text{res}}} N_i}, \tag{18}$$

where the first sum in the numerator iterates over all N_{res} residues in the protein, while the second sum considers the N_i native contacts of residue i separated by at least seven residues in the primary sequence. Here, $d_{ij}(\mathbf{x})$ and d^*_{ij} represent the $C_\alpha - C_\alpha$ distances of residues i and j in the structure $\mathbf{x}$ and the native structure, respectively. This observable quantifies the protein's foldedness, with values ranging from 0 (unfolded) to 1 (folded). For a detailed explanation, refer to Appendix A.11.



(a)



(b)

Figure 2: Absolute difference in correlation with respect to long unbiased MD simulations (lower is better) of models trained on trajectories initialized on samples from a Boltzmann Generator (BG) and a single structure (crystal, $\mathbf{x} = 0.75$) for the Prinz Potential under direct sampling. The former presents superior performance for different lag times (number of trajectories = 50) (a) and number of trajectories (b). The shaded areas correspond to 95 % confidence interval.

4.2 Boltzmann priors for training data generation

Modeling the transition density over Ω requires observing transitions across the state space. In the usual setting, only one or a few initial conditions are available when data collection starts. Here, we explore the case where a BG is available before data collection and compare it against the baseline, where only one initial condition is known. We use a BG trained on equilibrium data of the

Prinz potential to sample the initial conditions of our training trajectories. We also generate trajectories using a single starting point (crystal, $\mathbf{x} = 0.75$) and compare their performance with long simulations (Figure 2). We compare the two different data generation strategies using the metric, $|\Delta\text{correlation}|$, which measures the absolute difference of the time-correlation function (dynamic observable) compared to the MD ground truth (Appendix A.12). We find that the performance of models using Boltzmann priors for data generation is superior for different lag times (Figure 2a) and number of trajectories (Figure 2b). The gap between BG and the crystal baseline increases with lag-time, and as expected decreases with the number of generated trajectories.

4.3 BoPITO efficiently samples long-term dynamics in a low-data context

Fixing the equilibrium contribution to the score field effectively reduces the number of parameters that need to be estimated. To test whether this prior information manifests as an improved sample efficiency we compared ITO and BoPITO models against each other with varying sizes of training data.

We find that the BoPITO models achieve a higher accuracy for long-term dynamics compared to ITO models when data is scarce, as the equilibrium distribution is known *a priori* and does not need to be learned from simulation data (Figure 3). The inductive bias in BoPITO models enables them to learn long-term dynamics, even in scenarios where ITO models fail. For the Prinz Potential, we find that while ITO suffers from poor performance modeling long-term dynamics when data is scarce, BoPITO models accurately capture long-term dynamics without worsening the performance on short and medium timescales. The results for Alanine Dipeptide and Chignolin show favorable scaling with increasing system size: we find that an order of magnitude more data is needed to train an ITO to the same accuracy as a BoPITO model trained on the same data. Moreover, we show in Appendix A.13 that the energies of configurations sampled from our prior Boltzmann Generator used in our experiments match closely the energies of MD samples.

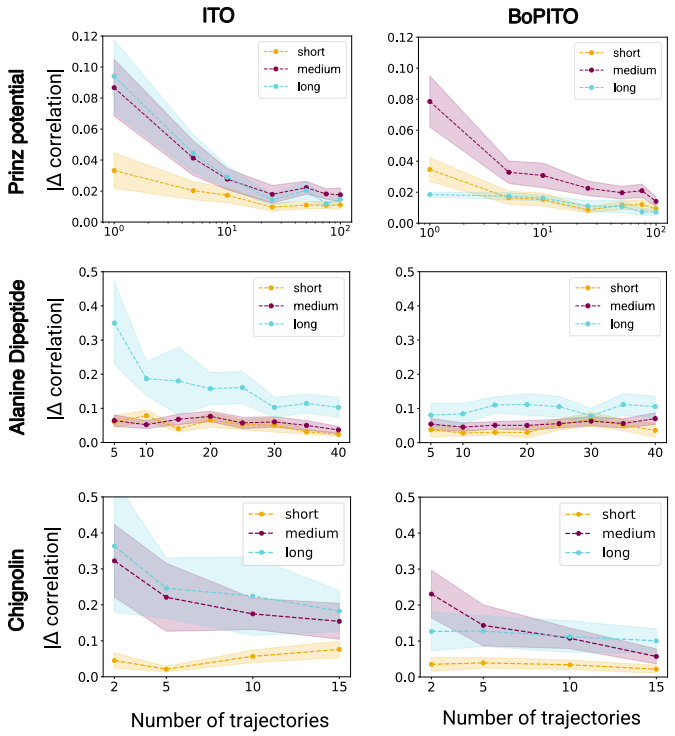


Figure 3: Absolute difference in correlation with respect to long unbiased MD simulations (lower is better, one-step sampling) of ITO (left) and BoPITO (right) split into short, medium, and long time-scales against the number of training trajectories for the Prinz potential (top) and Alanine Dipeptide (middle) and Chignolin (bottom). The shaded areas correspond to a 95 % confidence interval.

4.4 Interpolating between models trained on off-equilibrium data and the Boltzmann distribution with experimental data

In practical settings, our MD data will be off-equilibrium, e.g. having sampled only one or a few of the relevant modes in the Boltzmann distribution $\mu(\mathbf{x})$ in a given trajectory. Consequently, unless extensive data across the domain Ω can be collected, models based on such data will be biased. An alternative to collecting more simulation data is

to use a multi-modal strategy where experimental data is used to fill the gaps left by simulation data and bridge to long time-scale dynamics (Salvi et al., 2016; Kolloff & Olsson, 2023). In this section we explore the potential of BoPITO interpolators to integrate dynamic observables with off-equilibrium simulation data to recover unbiased long-term dynamics.

We showcase BoPITO interpolators on Alanine Dipeptide. By removing the transitions between the modes of the Boltzmann distribution corresponding to the slowest process in the system, e.g. ϕ crossing 0 or 2, we generate a biased simulation data, resembling a realistic scenario. We then train ITO and BoPITO models using these biased trajectories. For lags > 100, we sample the BoPITO interpolator estimated by matching to an unbiased dynamic observable defined by eq. 25, allowing us to overcome a systematic error in the correlation function observed in the biased MD data and for an ITO model trained on these data (Figure 4).

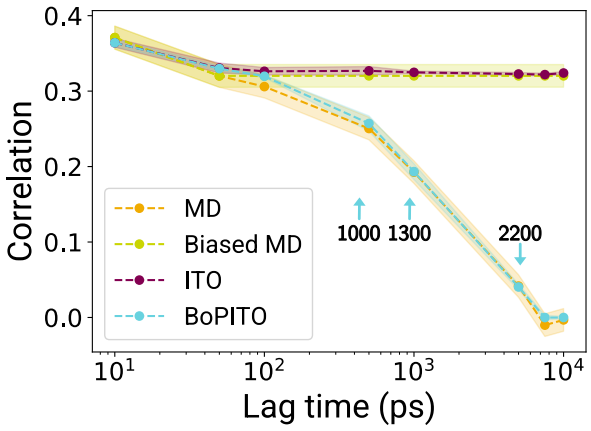


Figure 4: Fitting an interpolator of a biased BoPITO model. Both the ITO and BoPITO models are trained on biased MD simulations of Alanine Dipeptide. We fit a BoPITO interpolator selecting the most consistent ensemble with the unbiased correlation function specified in eq. 16. Numbers with blue arrows specify the interpolation parameter, N_{int}.

Beyond reproducing the provided dynamic observable, the interpolator also demonstrates a remarkable ability to capture the underlying microscopic dynamics (Figure 5). Even with excellent agreement, the generated ensembles only slightly overestimates density in the transition state region. However, this effect could easily be alleviated by annealing a short MD simulation to the interpolation as was recently shown (Viguera Diez et al., 2024).

5 Related work

Sampling the Boltzmann distribution Apart from Boltzmann Generator-based approaches, there are a number of traditional ways to draw statistical samples from the Boltzmann distribution of molecular systems. Prominent examples include, molecular dynamics or Markov Chain Monte Carlo simulations coupled with enhanced sampling strategies (Hénin et al., 2022; Kamenik et al., 2022), including replica-based approaches (Earl & Deem, 2005; Sidler et al., 2016; Pasarkar et al., 2023), conformational flooding (Grubmüller, 1995), meta-dynamics (Laio & Parrinello, 2002), and umbrella sampling (Torrie & Valleau, 1977), in particular when paired up with machine-learned collective variables (Chen & Ferguson, 2018; Wang et al., 2019b; Herringer et al., 2023; M. Sultan & Pande, 2017). The success of these approaches relies on choosing the right mechanism to enhance sampling across high free energy barriers, or low probability regions, for any given case (Carter et al., 1989). Finding such mechanisms typically involves substantial manual engineering of collective variables and other hyper-parameters. Finally, there are transition path sampling approaches (Dellago et al., 1998; Bolhuis et al., 2002), which are particularly powerful when combined with reinforcement learning (Jung et al., 2023) or deep generative priors (Plainer et al., 2023).

Latent space simulators and Coarse graining Classical (fine-grained) molecular dynamics simulation can be coarse-grained beyond the Born-Oppenheimer approximation, such that atomic nuclei are merged together into 'beads.'(Noid, 2023) This strategy, in principle enables faster simulations due to the smaller number of particles and the acceleration of kinetics caused by the coarse-graining (CG) operation (Nüske et al., 2019). CG models can be estimated to closely approximate the thermodynamics of the corresponding fine-grained system through the optimization of two equivalent variational bounds (Noid et al., 2008; Lyubartsev & Laaksonen, 1995; Ercolessi & Adams, 1994). A relaxation of this bound was recently proposed to also allow for probabilistic reconstruction of fine-

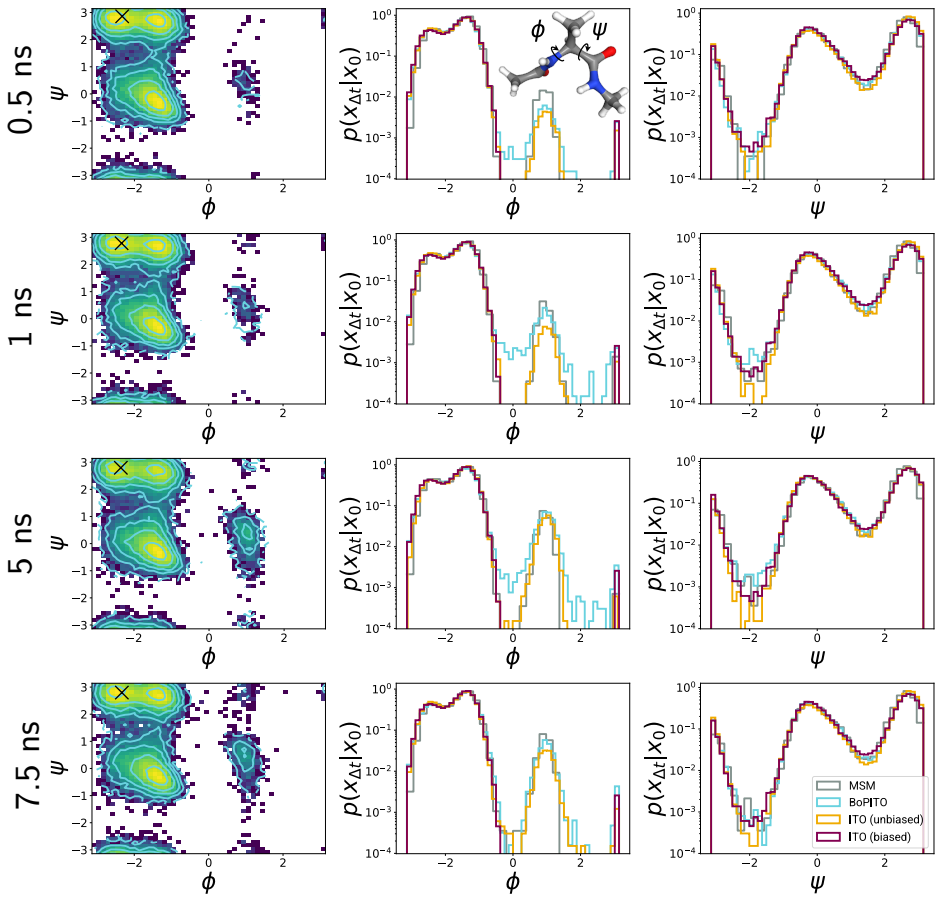


Figure 5: BoPITO can incorporate unbiased dynamic observables to correct a model trained on biased data. Rows of increasing time-lag (from top to bottom). Contour plots correspond to a BoPITO interpolator. The first column shows conditional transition densities projected onto the torsion angles ϕ and ψ (inset). The black cross indicates the initial condition. The second and third columns show marginal distributions of ϕ and ψ, respectively. MSM stands for a Markov State Model of the unbiased MD data.

grained configurations (Chennakesavalu et al., 2023). These bounds have been used extensively to build CG force-field models (Husic et al., 2020; Wang et al., 2019a; Majewski et al., 2023; Charron et al., 2023) including implicit solvation models (Chen et al., 2021; Katzberger & Riniker, 2024), using deep neural networks due to their ability to capture multibody terms (Wang et al., 2021) to ultimately accelerate the prediction of equilibrium properties of molecular systems. Similarly, the development of *'latent space simulators'* where a learned, typically low-dimensional, latent space equipped either with a propagator (Sidky et al., 2020; Chennakesavalu et al., 2023; Wang et al., 2024), or not (Wang & Gómez-Bombarelli, 2019), is learned to enable efficient simulation. These approaches, in general aim to accelerate molecular simulations akin to BoPITO, yet due to the CG operation, the molecular dynamics (kinetics) will be accelerated, and detailed knowledge of the unbiased dynamics are needed to correct this (Nüske et al., 2019; Crommelin & Vanden-Eijnden, 2011). Concurrent work, presents MDGen where conformational states are tokenized and in turn used to generate multiple frames of a MD trajectory jointly (Jing et al., 2024).

Transfer Operator surrogates Analysis of MD data often involves building transfer operator surrogates such as Markov state models (MSM) (Schütte et al., 1998; Prinz et al., 2011; Swope et al., 2004; Husic & Pande, 2018), time-lagged independent components analysis (Molgedey & Schuster, 1994; Ziehe & Müller, 1998; Pérez-Hernández et al., 2013), Markov field models or dynamic graphical models, (Olsson & Noé, 2019; Mardt et al., 2022; Hempel et al., 2022), VAMPnets

(Mardt et al., 2018; Wu & Noé, 2019), or observable operator models (Wu et al., 2015). Markov state models are time-space discrete approximations of the transfer operator and Deep Generative MSM (Wu et al., 2019) and VAMPnets (Mardt et al., 2018) learn the space discretization through deep neural networks. Dynamic graphical models or Markov field models (Hempel et al., 2022) represent a time-space discrete approximation of the transfer operator injecting a (conditional) independence assumption of molecular subsystems, when modeling the transition probability, leading to better scaling for systems with poor time-scale separation (Olsson & Noé, 2019). Apart from ITO (Schreiner et al., 2023) several other deep generative approaches for modeling the transition density of molecular dynamics have recently proposed. *Timewarp* where a normalizing flow is used to encode the transition density (Klein et al., 2023) with limited transferability to enable metropolized sampling of unbiased equilibrium distributions (Hastings, 1970). *Score dynamics* use a DDPM to model the displacements of an initial configuration towards a time-lagged one, achieving picosecond time-steps simulation and limited transferability (Hsu et al., 2024). However, unlike in the context of MSMs (Trendelkamp-Schroer et al., 2015; Rosta & Hummer, 2014; Wu et al., 2014; 2016), there are no deep generative transition density surrogates available leveraging available information about the equilibrium distribution — BoPITO is one such method.

6 Limitations and future work

Choice of hyper-parameter $\hat{\lambda}$ The hyper-parameter $\hat{\lambda}$, defines a global relaxation or mixing time-scale of the dynamics after which the model is guaranteed to sample equilibrium. In Appendix A.14 we describe a protocol to determine a bound for this parameter. However, developing a similar protocol for a transferable BoPITO model would likely require modifications to accommodate the diversity of global relaxation time-scales across different systems.

No Chapman-Kolmogorov Guarantee When training on biased or off-equilibrium data where we rely on establishing ergodicity through interpolation we cannot guarantee self-consistency of the dynamics in the Chapman-Kolmogorov sense.

Surrogate model BoPITO inherits the current limitations of ITO, such as generalization over chemical space and thermodynamic variables, and scaling. Furthermore, current models cannot guarantee unbiased sampling dynamics for non-equilibrium ensembles, which would require closed-form expressions for the target path probabilities.

7 Conclusion

We introduce Boltzmann Priors for Implicit transfer Operator Learning (BoPITO), a framework to enhance ITO learning in three ways. First, a broad sampling of configuration space is used to initialize short off-equilibrium MD simulations. Second, we parameterize the transition density as an interpolation towards a pre-trained Boltzmann Generator, improving sample efficiency by an order of magnitude. BoPITO is a principled approach to embedding prior knowledge of the stationary distribution of Markovian dynamics as an inductive bias for long-term dynamical behavior. Third, our approach enables interpolation between models trained on off-equilibrium data and the equilibrium distribution, and we can recover accurate models of unseen dynamics when informed by unbiased observables. Consequently, BoPITO is the first method to allow for the integration of multiple sources of information into the generation of deep generative surrogates of molecular dynamics.

REFERENCES

Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, May 1982. ISSN 0304-4149. doi: 10.1016/0304-4149(82)90051-5. URL http://dx.doi.org/10.1016/0304-4149(82)90051-5.

Marloes Arts, Victor Garcia Satorras, Chin-Wei Huang, Daniel Zügner, Marco Federici, Cecilia Clementi, Frank Noé, Robert Pinsler, and Rianne van den Berg. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *Journal of Chemical Theory and Computation*, 19(18):6151–6159, September 2023. ISSN 1549-9626. doi: 10.1021/acs.jctc.3c00702. URL http://dx.doi.org/10.1021/acs.jctc.3c00702.

Robin M. Betz and Ron O. Dror. How effectively can adaptive sampling methods capture spontaneous ligand binding? *Journal of Chemical Theory and Computation*, 15(3):2053–2063, January 2019. ISSN 1549-9626. doi: 10.1021/acs.jctc.8b00913. URL http://dx.doi.org/10.1021/acs.jctc.8b00913.

Peter G. Bolhuis, David Chandler, Christoph Dellago, and Phillip L. Geissler. Transition path sampling: Throwing ropes over rough mountain passes, in the dark. *Annual Review of Physical Chemistry*, 53(1):291–318, October 2002. ISSN 1545-1593. doi: 10.1146/annurev.physchem.53.082301.113146. URL http://dx.doi.org/10.1146/annurev.physchem.53.082301.113146.

Sandro Bottaro and Kresten Lindorff-Larsen. Biophysical experiments and biomolecular simulations: A perfect match? *Science*, 361(6400):355–360, July 2018. ISSN 1095-9203. doi: 10.1126/science.aat4010. URL http://dx.doi.org/10.1126/science.aat4010.

Gregory R. Bowman, Daniel L. Ensign, and Vijay S. Pande. Enhanced modeling via network theory: Adaptive sampling of markov state models. *Journal of Chemical Theory and Computation*, 6(3): 787–794, February 2010. ISSN 1549-9626. doi: 10.1021/ct900620b. URL http://dx.doi.org/10.1021/ct900620b.

E.A. Carter, Giovanni Ciccotti, James T. Hynes, and Raymond Kapral. Constrained reaction coordinate dynamics for the simulation of rare events. *Chemical Physics Letters*, 156(5):472–477, April 1989. ISSN 0009-2614. doi: 10.1016/s0009-2614(89)87314-2. URL http://dx.doi.org/10.1016/S0009-2614(89)87314-2.

Nicholas E. Charron, Felix Musil, Andrea Guljas, Yaoyi Chen, Klara Bonneau, Aldo S. Pasos-Trejo, Jacopo Venturin, Daria Gusew, Iryna Zaporozhets, Andreas Krämer, Clark Templeton, Atharva Kelkar, Aleksander E. P. Durumeric, Simon Olsson, Adrià Pérez, Maciej Majewski, Brooke E. Husic, Ankit Patel, Gianni De Fabritiis, Frank Noé, and Cecilia Clementi. Navigating protein landscapes with a machine-learned transferable coarse-grained model, 2023.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2018.

Wei Chen and Andrew L. Ferguson. Molecular enhanced sampling with autoencoders: On-the-fly collective variable discovery and accelerated free energy landscape exploration. *Journal of Computational Chemistry*, 39(25):2079–2102, September 2018. ISSN 1096-987X. doi: 10.1002/jcc.25520. URL http://dx.doi.org/10.1002/jcc.25520.

Yaoyi Chen, Andreas Krämer, Nicholas E. Charron, Brooke E. Husic, Cecilia Clementi, and Frank Noé. Machine learning implicit solvation for molecular dynamics. *The Journal of Chemical Physics*, 155(8):084101, August 2021. ISSN 1089-7690. doi: 10.1063/5.0059915. URL http://dx.doi.org/10.1063/5.0059915.

Shriram Chennakesavalu, David J. Toomer, and Grant M. Rotskoff. Ensuring thermodynamic consistency with invertible coarse-graining. *The Journal of Chemical Physics*, 158(12), March 2023. ISSN 1089-7690. doi: 10.1063/5.0141888. URL http://dx.doi.org/10.1063/5.0141888.

Daan Crommelin and Eric Vanden-Eijnden. Diffusion estimation from multiscale data by operator eigenpairs. *Multiscale Modeling & Simulation*, 9(4):1588–1623, October 2011. ISSN 1540-3467. doi: 10.1137/100795917. URL http://dx.doi.org/10.1137/100795917.

Christoph Dellago, Peter G. Bolhuis, and David Chandler. Efficient transition path sampling: Application to lennard-jones cluster rearrangements. *The Journal of Chemical Physics*, 108(22): 9236–9245, June 1998. ISSN 1089-7690. doi: 10.1063/1.476378. URL http://dx.doi.org/10.1063/1.476378.

Manuel Dibak, Leon Klein, Andreas Krämer, and Frank Noé. Temperature steerable flows and boltzmann generators, 2022. URL https://arxiv.org/abs/2108.01590.

S. Doerr and G. De Fabritiis. On-the-fly learning and sampling of ligand binding by high-throughput molecular simulations. *Journal of Chemical Theory and Computation*, 10(5):2064–2069, April 2014. ISSN 1549-9626. doi: 10.1021/ct400919u. URL http://dx.doi.org/10.1021/ct400919u.

David J. Earl and Michael W. Deem. Parallel tempering: Theory, applications, and new perspectives. *Phys. Chem. Chem. Phys.*, 7:3910–3916, 2005. doi: 10.1039/B509983H. URL http://dx.doi.org/10.1039/B509983H.

F Ercolessi and J. B Adams. Interatomic potentials from first-principles calculations: The force-matching method. *Europhysics Letters (EPL)*, 26(8):583–588, June 1994. ISSN 1286-4854. doi: 10.1209/0295-5075/26/8/005. URL http://dx.doi.org/10.1209/0295-5075/26/8/005.

Alan M. Ferrenberg and Robert H. Swendsen. Optimized monte carlo data analysis. *Physical Review Letters*, 63(12):1195–1198, September 1989.

Xiang Fu, Tian Xie, Nathan J. Rebello, Bradley D. Olsen, and Tommi Jaakkola. Simulate time-integrated coarse-grained molecular dynamics with multi-scale graph networks, 2023. URL https://arxiv.org/abs/2204.10348.

Helmut Grubmüller. Predicting slow structural transitions in macromolecular systems: Conformational flooding. *Phys. Rev. E*, 52:2893–2906, 9 1995. doi: 10.1103/PhysRevE.52.2893. URL https://link.aps.org/doi/10.1103/PhysRevE.52.2893.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970. doi: 10.1093/biomet/57.1.97. URL https://doi.org/10.1093/biomet/57.1.97.

Tim Hempel, Simon Olsson, and Frank Noé. Markov field models: Scaling molecular kinetics approaches to large molecular machines. *Current Opinion in Structural Biology*, 77:102458, December 2022. ISSN 0959-440X. doi: 10.1016/j.sbi.2022.102458. URL http://dx.doi.org/10.1016/j.sbi.2022.102458.

Jérôme Hénin, Tony Lelièvre, Michael R Shirts, Omar Valsson, and Lucie Delemotte. Enhanced sampling methods for molecular dynamics simulations [article v1.0]. *Living Journal of Computational Molecular Science*, 4(1), 2022. doi: 10.33011/livecoms.4.1.1583. URL https://doi.org/10.33011/livecoms.4.1.1583.

Nicholas S. M. Herringer, Siva Dasetty, Diya Gandhi, Junhee Lee, and Andrew L. Ferguson. Permutationally invariant networks for enhanced sampling (pines): Discovery of multimolecular and solvent-inclusive collective variables. *Journal of Chemical Theory and Computation*, 20(1):178–198, December 2023. ISSN 1549-9626. doi: 10.1021/acs.jctc.3c00923. URL http://dx.doi.org/10.1021/acs.jctc.3c00923.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Moritz Hoffmann, Martin Scherer, Tim Hempel, Andreas Mardt, Brian de Silva, Brooke E Husic, Stefan Klus, Hao Wu, Nathan Kutz, Steven L Brunton, and Frank Noé. Deeptime: a python library for machine learning dynamical models from time series data. *Machine Learning: Science and Technology*, 3(1):015009, December 2021. ISSN 2632-2153. doi: 10.1088/2632-2153/ac3de0. URL http://dx.doi.org/10.1088/2632-2153/ac3de0.

Tim Hsu, Babak Sadigh, Vasily Bulatov, and Fei Zhou. Score dynamics: Scaling molecular dynamics with picoseconds time steps via conditional diffusion model. *Journal of Chemical Theory and Computation*, 20(6):2335–2348, March 2024. ISSN 1549-9626. doi: 10.1021/acs.jctc.3c01361. URL http://dx.doi.org/10.1021/acs.jctc.3c01361.

Brooke E. Husic and Vijay S. Pande. Markov state models: From an art to a science. *Journal of the American Chemical Society*, 140(7):2386–2396, February 2018. doi: 10.1021/jacs.7b12191. URL https://doi.org/10.1021/jacs.7b12191.

Brooke E. Husic, Nicholas E. Charron, Dominik Lemm, Jiang Wang, Adrià Pérez, Maciej Majewski, Andreas Krämer, Yaoyi Chen, Simon Olsson, Gianni de Fabritiis, Frank Noé, and Cecilia Clementi. Coarse graining molecular dynamics with graph neural networks. *The Journal of Chemical Physics*, 153(19), November 2020. ISSN 1089-7690. doi: 10.1063/5.0026133. URL http://dx.doi.org/10.1063/5.0026133.

Bowen Jing, Hannes Stärk, Tommi Jaakkola, and Bonnie Berger. Generative modeling of molecular dynamics trajectories, 2024.

Hendrik Jung, Roberto Covino, A. Arjun, Christian Leitold, Christoph Dellago, Peter G. Bolhuis, and Gerhard Hummer. Machine-guided path sampling to discover mechanisms of molecular self-organization. *Nature Computational Science*, 3(4):334–345, April 2023. ISSN 2662-8457. doi: 10.1038/s43588-023-00428-z. URL http://dx.doi.org/10.1038/s43588-023-00428-z.

Anna S. Kamenik, Stephanie M. Linker, and Sereina Riniker. Enhanced sampling without borders: on global biasing functions and how to reweight them. *Physical Chemistry Chemical Physics*, 24 (3):1225–1236, 2022. ISSN 1463-9084. doi: 10.1039/d1cp04809k. URL http://dx.doi.org/10.1039/D1CP04809K.

Paul Katzberger and Sereina Riniker. A general graph neural network based implicit solvation model for organic molecules in water. *Chemical Science*, 15(28):10794–10802, 2024. ISSN 2041-6539. doi: 10.1039/d4sc02432j. URL http://dx.doi.org/10.1039/D4SC02432J.

Leon Klein and Frank Noé. Transferable boltzmann generators, 2024. URL https://arxiv.org/abs/2406.14426.

Leon Klein, Andrew Y. K. Foong, Tor Erlend Fjelde, Bruno Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noé, and Ryota Tomioka. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics, 2023. URL https://arxiv.org/abs/2302.01170.

Jonas Köhler, Leon Klein, and Frank Noe. Equivariant flows: Exact likelihood generative learning for symmetric densities. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5361–5370. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/kohler20a.html.

Jonas Köhler, Michele Invernizzi, Pim De Haan, and Frank Noe. Rigid body flows for sampling molecular crystal structures. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17301–17326. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kohler23a.html.

Christopher Kolloff and Simon Olsson. Rescuing off-equilibrium simulation data through dynamic experimental data with dynammo. *Machine Learning: Science and Technology*, 4(4):045050, December 2023. ISSN 2632-2153. doi: 10.1088/2632-2153/ad10ce. URL http://dx.doi.org/10.1088/2632-2153/ad10ce.

Alessandro Laio and Michele Parrinello. Escaping free-energy minima. *Proceedings of the National Academy of Sciences*, 99(20):12562–12566, 2002.

Paul Langevin. Sur la théorie du mouvement brownien. *C. R. Acad. Sci. (Paris)*, 146:530—-533, 1908.

Kresten Lindorff-Larsen, Stefano Piana, Ron O. Dror, and David E. Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, October 2011. ISSN 1095-9203. doi: 10.1126/science.1208351. URL http://dx.doi.org/10.1126/science.1208351.

Alexander P. Lyubartsev and Aatto Laaksonen. Calculation of effective interaction potentials from radial distribution functions: A reverse monte carlo approach. *Physical Review E*, 52(4): 3730–3737, October 1995. ISSN 1095-3787. doi: 10.1103/physreve.52.3730. URL http://dx.doi.org/10.1103/PhysRevE.52.3730.

Mohammad M. Sultan and Vijay S. Pande. tica-metadynamics: Accelerating metadynamics by using kinetically selected collective variables. *Journal of Chemical Theory and Computation*, 13(6):2440–2447, May 2017. ISSN 1549-9626. doi: 10.1021/acs.jctc.7b00182. URL http://dx.doi.org/10.1021/acs.jctc.7b00182.

Maciej Majewski, Adrià Pérez, Philipp Thölke, Stefan Doerr, Nicholas E. Charron, Toni Giorgino, Brooke E. Husic, Cecilia Clementi, Frank Noé, and Gianni De Fabritiis. Machine learning coarse-grained potentials of protein thermodynamics. *Nature Communications*, 14(1), September 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-41343-1. URL http://dx.doi.org/10.1038/s41467-023-41343-1.

Dimitra Maoutsa, Sebastian Reich, and Manfred Opper. Interacting particle solutions of fokker–planck equations through gradient–log–density estimation. *Entropy*, 22(8):802, July 2020. ISSN 1099-4300. doi: 10.3390/e22080802. URL http://dx.doi.org/10.3390/e22080802.

Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. VAMPnets for deep learning of molecular kinetics. *Nature Communications*, 9(1), jan 2018. doi: 10.1038/s41467-017-02388-1. URL https://doi.org/10.1038%2Fs41467-017-02388-1.

Andreas Mardt, Tim Hempel, Cecilia Clementi, and Frank Noé. Deep learning to decompose macromolecules into independent markovian domains. *Nature Communications*, 13(1), November 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-34603-z. URL http://dx.doi.org/10.1038/s41467-022-34603-z.

Laurence I. Midgley, Vincent Stimper, Javier Antorán, Emile Mathieu, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Se(3) equivariant augmented coupling flows, 2024. URL https://arxiv.org/abs/2308.10364.

Laurence Illing Midgley, Vincent Stimper, Gregor N. C. Simm, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Flow annealed importance sampling bootstrap, 2023. URL https://arxiv.org/abs/2208.01893.

L. Molgedey and H. G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Phys. Rev. Lett.*, 72:3634–3637, Jun 1994. doi: 10.1103/PhysRevLett.72.3634. URL https://link.aps.org/doi/10.1103/PhysRevLett.72.3634.

Kim A. Nicoli, Shinichi Nakajima, Nils Strodthoff, Wojciech Samek, Klaus-Robert Müller, and Pan Kessel. Asymptotically unbiased estimation of physical observables with neural samplers. *Physical Review E*, 101(2), February 2020. ISSN 2470-0053. doi: 10.1103/physreve.101.023304. URL http://dx.doi.org/10.1103/PhysRevE.101.023304.

W. G. Noid. Perspective: Advances, challenges, and insight for predictive coarse-grained models. *The Journal of Physical Chemistry B*, 127(19):4174–4207, May 2023. ISSN 1520-5207. doi: 10.1021/acs.jpcb.2c08731. URL http://dx.doi.org/10.1021/acs.jpcb.2c08731.

W. G. Noid, Jhih-Wei Chu, Gary S. Ayton, Vinod Krishna, Sergei Izvekov, Gregory A. Voth, Avisek Das, and Hans C. Andersen. The multiscale coarse-graining method. i. a rigorous bridge between atomistic and coarse-grained models. *The Journal of Chemical Physics*, 128(24), June 2008. ISSN 1089-7690. doi: 10.1063/1.2938860. URL http://dx.doi.org/10.1063/1.2938860.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457), September 2019. ISSN 1095-9203. doi: 10.1126/science.aaw1147. URL http://dx.doi.org/10.1126/science.aaw1147.

Feliks Nüske, Hao Wu, Jan-Hendrik Prinz, Christoph Wehmeyer, Cecilia Clementi, and Frank Noé. Markov state models from short non-equilibrium simulations—analysis and correction of estimation bias. *The Journal of Chemical Physics*, 146(9), March 2017. ISSN 1089-7690. doi: 10.1063/1.4976518. URL http://dx.doi.org/10.1063/1.4976518.

Feliks Nüske, Lorenzo Boninsegna, and Cecilia Clementi. Coarse-graining molecular systems by spectral matching. *The Journal of Chemical Physics*, 151(4), July 2019. ISSN 1089-7690. doi: 10.1063/1.5100131. URL http://dx.doi.org/10.1063/1.5100131.

Simon Olsson. Markov state models of protein–protein encounters, November 2022. URL http://dx.doi.org/10.1002/9783527830503.ch9.

Simon Olsson and Frank Noé. Dynamic graphical models of molecular kinetics. *Proceedings of the National Academy of Sciences*, 116(30):15001–15006, July 2019. doi: 10.1073/pnas.1901692116. URL https://doi.org/10.1073/pnas.1901692116.

Simon Olsson, Hao Wu, Fabian Paul, Cecilia Clementi, and Frank Noé. Combining experimental and simulation data of molecular processes via augmented markov models. *Proceedings of the National Academy of Sciences*, 114(31):8265–8270, July 2017. ISSN 1091-6490. doi: 10.1073/pnas.1704803114. URL http://dx.doi.org/10.1073/pnas.1704803114.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference, 2021. URL https://arxiv.org/abs/1912.02762.

Amey P Pasarkar, Gianluca M Bencomo, Simon Olsson, and Adji Bousso Dieng. Vendi sampling for molecular simulations: Diversity as a force for faster convergence and better exploration. *The Journal of chemical physics*, 159(14), 10 2023. ISSN 1089-7690 (Electronic). doi: 10.1063/5.0166172.

Michael Plainer, Hannes Stärk, Charlotte Bunne, and Stephan Günnemann. Transition path sampling with boltzmann generator-based mcmc moves, 2023.

Jan-Hendrik Prinz, Hao Wu, Marco Sarich, Bettina Keller, Martin Senne, Martin Held, John D. Chodera, Christof Schütte, and Frank Noé. Markov models of molecular kinetics: Generation and validation. *The Journal of Chemical Physics*, 134(17):174105, May 2011. doi: 10.1063/1.3565032. URL https://doi.org/10.1063/1.3565032.

Guillermo Pérez-Hernández, Fabian Paul, Toni Giorgino, Gianni De Fabritiis, and Frank Noé. Identification of slow molecular order parameters for markov model construction. *The Journal of Chemical Physics*, 139(1), July 2013. ISSN 1089-7690. doi: 10.1063/1.4811489. URL http://dx.doi.org/10.1063/1.4811489.

Edina Rosta and Gerhard Hummer. Free energies from dynamic weighted histogram analysis using unbiased markov state model. *Journal of Chemical Theory and Computation*, 11(1):276–285, December 2014. ISSN 1549-9626. doi: 10.1021/ct500719p. URL http://dx.doi.org/10.1021/ct500719p.

David Ruelle. *Thermodynamic Formalism*. Encyclopedia of mathematics and its applications. Longman Higher Education, Harlow, England, November 1978.

Nicola Salvi, Anton Abyzov, and Martin Blackledge. Multi-timescale dynamics in intrinsically disordered proteins from nmr relaxation and molecular simulation. *The Journal of Physical Chemistry Letters*, 7(13):2483–2489, June 2016. ISSN 1948-7185. doi: 10.1021/acs.jpclett.6b00885. URL http://dx.doi.org/10.1021/acs.jpclett.6b00885.

Mathias Schreiner, Ole Winther, and Simon Olsson. Implicit transfer operator learning: Multiple time-resolution surrogates for molecular dynamics, 2023. URL https://arxiv.org/abs/2305.18046.

Christof Schütte, Alexander Fischer, Wilhelm Huisinga, and Peter Deuflhard. A hybrid monte carlo method for essential molecular dynamics. Technical Report SC-98-04, ZIB, Takustr. 7, 14195 Berlin, 1998.

Christof Schütte, Frank Noe, Eike Meerbach, Philipp Metzner, and Carsten Hartmann. Conformation dynamics. *Proc. Int. Congr. ICIAM*, pp. 297–336, 2009.

Michael R. Shirts and John D. Chodera. Statistically optimal analysis of samples from multiple equilibrium states. *The Journal of Chemical Physics*, 129(12), September 2008. ISSN 1089-7690. doi: 10.1063/1.2978177. URL http://dx.doi.org/10.1063/1.2978177.

Hythem Sidky, Wei Chen, and Andrew L. Ferguson. Molecular latent space simulators. *Chemical Science*, 11(35):9459–9467, 2020. ISSN 2041-6539. doi: 10.1039/d0sc03635h. URL http://dx.doi.org/10.1039/D0SC03635H.

Dominik Sidler, Arthur Schwaninger, and Sereina Riniker. Replica exchange enveloping distribution sampling (re-eds): A robust method to estimate multiple free-energy differences from a single simulation. *The Journal of Chemical Physics*, 145(15), October 2016. ISSN 1089-7690. doi: 10.1063/1.4964781. URL http://dx.doi.org/10.1063/1.4964781.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2020. URL https://arxiv.org/abs/2011.13456.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models, 2021.

William C. Swope, Jed W. Pitera, and Frank Suits. Describing protein folding kinetics by molecular dynamics simulations. 1. theory. *The Journal of Physical Chemistry B*, 108(21):6571–6581, April 2004. doi: 10.1021/jp037421y. URL https://doi.org/10.1021/jp037421y.

G.M. Torrie and J.P. Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187–199, February 1977. doi: 10.1016/0021-9991(77)90121-8. URL https://doi.org/10.1016/0021-9991(77)90121-8.

Benjamin Trendelkamp-Schroer, Hao Wu, Fabian Paul, and Frank Noé. Estimation and uncertainty of reversible markov models. *The Journal of Chemical Physics*, 143(17), November 2015. ISSN 1089-7690. doi: 10.1063/1.4934536. URL http://dx.doi.org/10.1063/1.4934536.

Juan Viguera Diez, Sara Romeo Atance, Ola Engkvist, and Simon Olsson. Generation of conformational ensembles of small molecules via surrogate model-assisted molecular dynamics. *Machine Learning: Science and Technology*, 5(2):025010, April 2024. ISSN 2632-2153. doi: 10.1088/2632-2153/ad3b64. URL http://dx.doi.org/10.1088/2632-2153/ad3b64.

Dedi Wang, Yihang Wang, Luke Evans, and Pratyush Tiwary. From latent dynamics to meaningful representations. *Journal of Chemical Theory and Computation*, 20(9):3503–3513, April 2024. ISSN 1549-9626. doi: 10.1021/acs.jctc.4c00249. URL http://dx.doi.org/10.1021/acs.jctc.4c00249.

Jiang Wang, Simon Olsson, Christoph Wehmeyer, Adrià Pérez, Nicholas E. Charron, Gianni de Fabritiis, Frank Noé, and Cecilia Clementi. Machine learning of coarse-grained molecular dynamics force fields. *ACS Central Science*, 5(5):755–767, April 2019a. ISSN 2374-7951. doi: 10.1021/acscentsci.8b00913. URL http://dx.doi.org/10.1021/acscentsci.8b00913.

Jiang Wang, Nicholas Charron, Brooke Husic, Simon Olsson, Frank Noé, and Cecilia Clementi. Multi-body effects in a coarse-grained protein force field. *The Journal of Chemical Physics*, 154 (16), April 2021. ISSN 1089-7690. doi: 10.1063/5.0041022. URL http://dx.doi.org/10.1063/5.0041022.

Wujie Wang and Rafael Gómez-Bombarelli. Coarse-graining auto-encoders for molecular dynamics. *npj Computational Materials*, 5(1), December 2019. ISSN 2057-3960. doi: 10.1038/s41524-019-0261-5. URL http://dx.doi.org/10.1038/s41524-019-0261-5.

Yihang Wang, João Marcelo Lamim Ribeiro, and Pratyush Tiwary. Past–future information bottleneck for sampling molecular reaction coordinate simultaneously with thermodynamics and kinetics. *Nature Communications*, 10(1), August 2019b. ISSN 2041-1723. doi: 10.1038/s41467-019-11405-4. URL http://dx.doi.org/10.1038/s41467-019-11405-4.

Hao Wu and Frank Noé. Variational approach for learning markov processes from time series data. *Journal of Nonlinear Science*, 30(1):23–66, aug 2019. doi: 10.1007/s00332-019-09567-y. URL https://doi.org/10.1007%2Fs00332-019-09567-y.

Hao Wu, Antonia S. J. S. Mey, Edina Rosta, and Frank Noé. Statistically optimal analysis of state-discretized trajectory data from multiple thermodynamic states. *The Journal of Chemical Physics*, 141(21), December 2014. ISSN 1089-7690. doi: 10.1063/1.4902240. URL http://dx.doi.org/10.1063/1.4902240.

Hao Wu, Jan-Hendrik Prinz, and Frank Noé. Projected metastable markov processes and their estimation with observable operator models. *The Journal of Chemical Physics*, 143(14):144101, October 2015. doi: 10.1063/1.4932406. URL https://doi.org/10.1063/1.4932406.

Hao Wu, Fabian Paul, Christoph Wehmeyer, and Frank Noé. Multiensemble markov models of molecular thermodynamics and kinetics. *Proceedings of the National Academy of Sciences*, 113 (23), May 2016. ISSN 1091-6490. doi: 10.1073/pnas.1525092113. URL http://dx.doi.org/10.1073/pnas.1525092113.

Hao Wu, Andreas Mardt, Luca Pasquali, and Frank Noe. Deep generative markov state models, 2019. URL https://arxiv.org/abs/1805.07601.

Andreas Ziehe and Klaus-Robert Müller. *TDSEP — an efficient algorithm for blind separation using time structure*, pp. 675–680. Springer London, 1998. ISBN 9781447115991. doi: 10.1007/978-1-4471-1599-1_103. URL http://dx.doi.org/10.1007/978-1-4471-1599-1_103.

A APPENDIX

A.1 DEFINITIONS

- **Configuration space**: Mathematical space in which all possible states or positions of a physical system are represented. For example, in a classical MD simulation, the positions of all the atoms in the simulation.

- **Unbiased simulations**: Simulations performed with standard MD, i.e. Langevin Dynamics. They are unbiased because they generate samples from the underlying transition density. Particularly, for asymptotically large simulation times, unbiased simulations sample the Boltzmann distribution $\mu(\mathbf{x})$. However, because of energy barriers in the potential energy landscape, unbiased simulations may not explore some modes of the Boltzmann distribution if the simulation time is not long enough. Therefore, generating samples with unbiased MD often leads to off-equilibrium data.

- **Off-equilibrium data**: Simulation trajectories whose underlying statistics do not represent well those of the equilibrium distribution. One example is lack of ergodicity: when the simulation fails to explore some modes in the Boltzmann distribution. Even if having explored the full state space, it is possible to have off-equilibrium trajectories because of rare events, which require to be sampled "many" times to accurately represent equilibrium statistics.

- **Biased simulation**: Simulations performed with modified versions of naive MD that speed up the exploration of state space. These methods are a subset of enhanced sampling and include meta-dynamics (Laio & Parrinello, 2002), replica exchange (Earl & Deem, 2005) and others (Sidler et al., 2016; Pasarkar et al., 2023). Data generated with enhanced sampling do not resemble the underlying transition density, but, often, can be re-weighted to represent the equilibrium distribution.

A.2 EIGEN DECOMPOSITION OF THE TRANSFER OPERATOR AND TRANSITION DENSITY

The two-fold composition of the transfer operator, eq. 5, acting on an initial μ-weighted density, ρ, is

$$
\begin{aligned}
[T_\Omega^2(\tau) \circ \rho](\mathbf{x}_{t+2\tau}) &= \sum_{i=1}^{\infty} \lambda_i(\tau) \left\langle \sum_{j=1}^{\infty} \lambda_j(\tau) \langle \rho | \phi_j \rangle \; \psi_j | \phi_i \right\rangle \; \psi_i(\mathbf{x}_{t+2\tau}) \\
&= \sum_{i=1}^{\infty} \lambda_i(\tau) \left(\sum_{j=1}^{\infty} \lambda_j(\tau) \langle \rho | \phi_j \rangle \; \langle \psi_j | \phi_i \rangle \right) \; \psi_i(\mathbf{x}_{t+2\tau}) \\
&= \sum_{i=1}^{\infty} \lambda_i^2(\tau) \langle \rho | \phi_i \rangle \; \psi_i(\mathbf{x}_{t+2\tau}),
\end{aligned}
$$

where we have used the orthonormality of the eigenfunctions, that is $\langle \psi_j | \phi_i \rangle = \delta_{ij}$. Similarly,

$$
[T_\Omega^N(\tau) \circ \rho](\mathbf{x}_{t+N\tau}) = \sum_{i=1}^{\infty} \lambda_i^N(\tau) \langle \rho | \phi_i \rangle \; \psi_i(\mathbf{x}_{t+N\tau}). \tag{19}
$$

The transition density can be re-written in terms of the spectral decomposition of the transfer operator by choosing the initial density, p, as a Dirac delta function $\delta_{\mathbf{x}_t}(\mathbf{x})$, that is $\rho(\mathbf{x}) = \frac{\delta_{\mathbf{x}_t}(\mathbf{x})}{\mu(\mathbf{x})}$.

Then,

$$
\begin{aligned}
p(\mathbf{x}_{t+N\tau}|\mathbf{x}_t) &= \mu(\mathbf{x}_{t+N\tau}) \left[T_\Omega^N(\tau) \circ \frac{\delta_{\mathbf{x}_t}}{\mu} \right] (\mathbf{x}_{t+N\tau}) \\
&= \mu(\mathbf{x}_{t+N\tau}) \sum_{i=1}^{\infty} \lambda_i^N(\tau) \left\langle \frac{\delta_{\mathbf{x}_t}}{\mu} \middle| \phi_i \right\rangle \psi_i(\mathbf{x}_{t+N\tau}) \\
&= \sum_{i=1}^{\infty} \lambda_i^N(\tau) \langle \delta_{\mathbf{x}_t}(\mathbf{x}) | \psi_i \rangle \ \psi_i(\mathbf{x}_{t+N\tau}) \ \mu(\mathbf{x}_{t+N\tau}) \\
&= \sum_{i=1}^{\infty} \lambda_i^N(\tau) \psi_i(\mathbf{x}_t) \phi_i(\mathbf{x}_{t+N\tau}).
\end{aligned}
$$

A.3 Implicit Transfer Operator details

Implicit Transfer Operator (ITO) Learning (Schreiner et al., 2023) is a framework for learning surrogate models of the transition density, $p(\mathbf{x}_{N\tau}|\mathbf{x}_0)$. ITO models are trained sampling tuples $(\mathbf{x}_{t_i}, \mathbf{x}_{t_i+N_i\tau}, N_i)$ and training a generative model to mimic the empirical transition density at multiple time-horizons, $N\tau$. Training is done following Algorithm 1.

Algorithm 1 Training. DisExp is defined in Algorithm 4

Input: n MD-trajectories; $\mathcal{X} = \{\mathbf{x}_0^j, \ldots, \mathbf{x}_{t_j}^j\}_{j=0}^n$, ITO score-model; $\hat{\epsilon}_\theta$, max lag; N_{max}
$\mathcal{X}' = \text{Concatenate}(\{\mathbf{x}_0^j, \ldots, \mathbf{x}_{t_j - N_{\text{max}}}^j\}_{j=0}^n)$
while not converged **do**
  $\mathbf{x}_t \sim \text{Choice}(\mathcal{X}')$
  $N \sim \text{DisExp}(N_{\text{max}})$
  $t_{\text{diff}} \sim \text{Uniform}(0, T)$
  Take gradient step on:
  $\nabla_{\boldsymbol{\theta}} \left[\| \epsilon - \hat{\epsilon}_{\boldsymbol{\theta}}(\widetilde{\mathbf{x}}_{t+N\tau}^{t_{\text{diff}}}, \mathbf{x}_t, N, t_{\text{diff}}) \|_2 \right]$
end while
return $\hat{\epsilon}_{\boldsymbol{\theta}}$

Once a model is trained, it can be sampled by following Algorithm 2.

Algorithm 2 Sampling from $\hat{p}_{\boldsymbol{\theta}}(\mathbf{x}_0, N)$

Input: initial condition $\mathbf{x}_0$, lag; N, diffusion steps; T_{diff}, ITO score-model; $\hat{\epsilon}_\theta$
$\mathbf{x}_N^{T_{\text{diff}}} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$
for $t_{\text{diff}} = T_{\text{diff}} \ldots 1$ **do**
  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$
  $\mathbf{x}_N^{t_{\text{diff}}-1} = \frac{1}{\sqrt{\alpha^{t_{\text{diff}}}}} \left(\mathbf{x}_N^{t_{\text{diff}}} - \frac{1-\alpha^{t_{\text{diff}}}}{\sqrt{1-\bar{\alpha}^{t_{\text{diff}}}}} \hat{\epsilon}_\theta(\mathbf{x}_N^{t_{\text{diff}}}, \mathbf{x}_0, N, t_{\text{diff}}) \right) + \sqrt{\beta_t} \epsilon$
end for
return $\mathbf{x}_N^0$

Several sampling steps can be annealed to sample longer lag-times as depicted in Algorithm 3.

Algorithm 3 Ancestral sampling. Sampling from $p_{\boldsymbol{\theta}}$ is defined in Algorithm 2

Input: initial condition $\mathbf{x}_0$, lag N, ancestral steps n.
Allocate $\mathcal{T} \in \mathbb{R}^{(n+1)\times \dim(\mathbf{x}_0)}$
$\mathcal{T}[0] = \mathbf{x}_0$
for $i = 1 \dots n$ **do**
  $\mathbf{x}_i \sim \hat{p}_{\boldsymbol{\theta}}(\mathcal{T}[i-1], N)$
  $\mathcal{T}[i] = \mathbf{x}_i$
end for
return $\mathcal{T}$

Algorithm 4 Sampling from DisExp

$N_{\text{log}} \sim \text{Uniform}(0, \log(N_{\text{max}}))$
Return: $\text{floor}(\exp(N_{\text{log}}))$

BoPITO uses Algorithm 1 for training and Algorithm 2 for sampling as well, but uses the score mode in eq. 14.

A.4 Decay of time-dependent score term

In Figure 6, we show the average time-dependent component of the score,

$$s_N = \mathbb{E}_{i\sim\mathbf{I},t_{\text{diff}}\sim\mathcal{U}(0,T)}\left[\hat{\lambda}^N s_{\text{dyn}}(\mathbf{x}_{t_i+N\tau}^{t_{\text{diff}}}, \mathbf{x}_{t_i}, N, t_{\text{diff}}, \boldsymbol{\theta})\right], \tag{20}$$

of a trained BoPITO model of the Prinz Potential with $\hat{\lambda} = 0.994$. The expectation for $\mathbf{I}$ corresponding to the sampling of tuples as during training (Alg. 1). The model remains flexible for small lags but eventually decreases to 0, sampling from the equilibrium model.

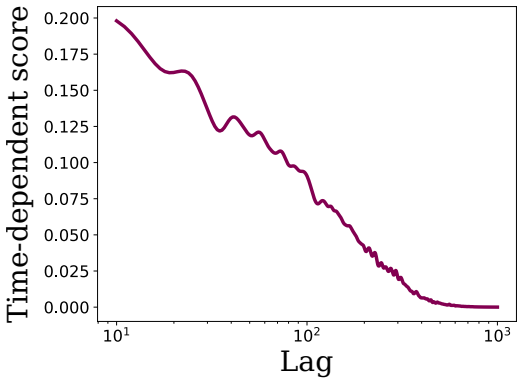


Figure 6: Average time-dependent component of the score, eq. 20, of a trained model BoPITO model of the Prinz Potential.

A.5 THE SCORE OF THE TRANSITION PROBABILITY AND BOPITO

The score of the transition probability can be written as

$$\begin{aligned}\nabla_{\mathbf{x}_{N\tau}} \log p(\mathbf{x}_{N\tau}|\mathbf{x}_0, N) &= \sum_i^{\infty} \lambda_i(\tau)^N \frac{\psi_i(\mathbf{x}_0)}{p(\mathbf{x}_{N\tau}|\mathbf{x}_0, N)} \nabla_{\mathbf{x}_{N\tau}} \phi_i(\mathbf{x}_{N\tau}) \\ &= \frac{\nabla_{\mathbf{x}_{N\tau}} \mu(\mathbf{x}_{N\tau})}{p(\mathbf{x}_{N\tau}|\mathbf{x}_0, N)} + \sum_{i=2}^{\infty} \lambda_i(\tau)^N \frac{\psi_i(\mathbf{x}_0)}{p(\mathbf{x}_{N\tau}|\mathbf{x}_0, N)} \nabla_{\mathbf{x}_{N\tau}} \phi_i(\mathbf{x}_{N\tau}) \\ &= \frac{\nabla_{\mathbf{x}_{N\tau}} \mu(\mathbf{x}_{N\tau})}{\mu(\mathbf{x}_{N\tau})} \frac{\mu(\mathbf{x}_{N\tau})}{p(\mathbf{x}_{N\tau}|\mathbf{x}_0, N)} \\ &+ \hat{\lambda}^N \sum_{i=2}^{\infty} \left(\frac{\lambda_i(\tau)}{\hat{\lambda}}\right)^N \frac{\psi_i(\mathbf{x}_0)}{p(\mathbf{x}_{N\tau}|\mathbf{x}_0, N)} \nabla_{\mathbf{x}_{N\tau}} \phi_i(\mathbf{x}_{N\tau}).\end{aligned}$$

$\frac{\nabla_{\mathbf{x}_{N\tau}} \mu(\mathbf{x}_{N\tau})}{\mu(\mathbf{x}_{N\tau})} = \nabla_{\mathbf{x}_{N\tau}} \log \mu(\mathbf{x}_{N\tau})$ is the score of the Boltzmann distribution and can be modeled with the score of a Boltzmann Generator, $s_{\text{eq}}(\mathbf{x}^{t_{\text{diff}}}, t_{\text{diff}})$. $\frac{\mu(\mathbf{x}_{N\tau})}{p(\mathbf{x}_{N\tau}|\mathbf{x}_0, N)} \to 1$ as $N \to \infty$ and can be modeled with a scalar neural network $1 + \hat{\lambda}^N f_{\boldsymbol{\theta}}(\mathbf{x}^{t_{\text{diff}}}_{N\tau}, \mathbf{x}_0, N, t_{\text{diff}})$. The last term can be modeled with $g_{\boldsymbol{\theta}}(\mathbf{x}^{t_{\text{diff}}}_{N\tau}, \mathbf{x}_0, N, t_{\text{diff}})$. The corresponding score model is

$$\begin{aligned}s_{\boldsymbol{\theta}}(\mathbf{x}^{t_{\text{diff}}}_{N\tau}, \mathbf{x}_0, N, t_{\text{diff}}) &= s_{\text{eq}}(\mathbf{x}^{t_{\text{diff}}}, t_{\text{diff}})(1 + \hat{\lambda}^N f_{\boldsymbol{\theta}}(\mathbf{x}^{t_{\text{diff}}}_{N\tau}, \mathbf{x}_0, N, t_{\text{diff}}) + \hat{\lambda}^N g_{\boldsymbol{\theta}}(\mathbf{x}^{t_{\text{diff}}}_{N\tau}, \mathbf{x}_0, N, t_{\text{diff}}) \\ &= s_{\text{eq}}(\mathbf{x}^{t_{\text{diff}}}, t_{\text{diff}}) + \hat{\lambda}^N \underbrace{\left(s_{\text{eq}}(\mathbf{x}^{t_{\text{diff}}}, t_{\text{diff}}) f_{\boldsymbol{\theta}}(\mathbf{x}^{t_{\text{diff}}}_{N\tau}, \mathbf{x}_0, N, t_{\text{diff}}) + g_{\boldsymbol{\theta}}(\mathbf{x}^{t_{\text{diff}}}_{N\tau}, \mathbf{x}_0, N, t_{\text{diff}})\right)}_{s_{\text{dyn}}(\mathbf{x}^{t_{\text{diff}}}_{N\tau}, \mathbf{x}_0, N, t_{\text{diff}})}.\end{aligned}$$

We can aggregate f and g to a single neural network component, $s_{\text{dyn}}(\mathbf{x}^{t_{\text{diff}}}_{N\tau}, \mathbf{x}_0, N, t_{\text{diff}})$, to get

$$s_{\boldsymbol{\theta}}(\mathbf{x}^{t_{\text{diff}}}_{N\tau}, \mathbf{x}_0, N, t_{\text{diff}}) = s_{\text{eq}}(\mathbf{x}^{t_{\text{diff}}}, t_{\text{diff}}) + \hat{\lambda}^N s_{\text{dyn}}(\mathbf{x}^{t_{\text{diff}}}_{N\tau}, \mathbf{x}_0, N, t_{\text{diff}}). \tag{21}$$

For simplicity, we incorporate the structure of the transition density into the diffusion model, not only for $t_{\text{diff}} = 0$ (the learned data distribution), but for all $t_{\text{diff}} > 0$. We do not observe this choice to limit model expressivity in our experiments.

A.6 ALTERNATIVE SCORE FORMULATIONS

The score in eq. 14 is principled and resembles the score of the transition density. However, in practice, we found that this score can lead to suboptimal performance in learning short time-scales if the dynamic component fails to dominate the equilibrium component for small N. We only observed this for Chignolin. We relate this issue to numerical limitations due to very different scale in neural network weights for different values of N. We remark that short timescales are the least relevant since they can be easily sampled with MD. Still, to mitigate this effect we propose the following alternative factorization,

$$s_{\boldsymbol{\theta}}(\mathbf{x}^{t_{\text{diff}}}_{N\tau}, \mathbf{x}_0, N, t_{\text{diff}}) = f(N) s_{\text{eq}}(\mathbf{x}^{t_{\text{diff}}}, t_{\text{diff}}) + \hat{\lambda}^N s_{\text{dyn}}(\mathbf{x}^{t_{\text{diff}}}_{N\tau}, \mathbf{x}_0, N, t_{\text{diff}}), \tag{22}$$

where $f(N)$ is an increasing function and tends to 1 for large N. One potential option that does not introduce extra hyper-parameters is

$$s_{\boldsymbol{\theta}}(\mathbf{x}^{t_{\text{diff}}}_{N\tau}, \mathbf{x}_0, N, t_{\text{diff}}) = \frac{\hat{\lambda}^{-N} - \hat{\lambda}^N}{\hat{\lambda}^{-N} + \hat{\lambda}^N} s_{\text{eq}}(\mathbf{x}^{t_{\text{diff}}}, t_{\text{diff}}) + \hat{\lambda}^N s_{\text{dyn}}(\mathbf{x}^{t_{\text{diff}}}_{N\tau}, \mathbf{x}_0, N, t_{\text{diff}}). \tag{23}$$

We use this version in the Chignolin experiments.

A.7 HOW BOPITO INTERPOLATORS MITIGATE OUT-OF-DISTRIBUTION EFFECTS

BoPITO interpolators mitigate out-of-distribution effects by, first, fixing $N = N_{max}$ as the argument of s_{dyn}. This choice ensures that none of our neural networks are evaluated outside of the training domain. Second, we alter long-lag interpolation steps with short-lag non-interpolation steps for local relaxation. Since we in practice see that that the long-lag steps occasionally generates ‘off data

manifold' states, e.g. structures with potential high energies, the local 'relaxation' steps projects us back onto the data manifold. In practice, just one short-lag non-interpolation step is sufficient to overcome this. Third, we use dynamic observables to fit the interpolation parameter. Without any additional source of information we don't know how to choose N_{int}. However, when we use dynamic observables, we can select the interpolated ensemble which is the most consistent with experimental observables. So this helps calibrate the time-scales to stay "in distribution".

A.8 Experimental parameters

A.8.1 Boltzmann Priors for dataset generation

For different numbers of trajectories, n, we train $5000/n$ ITO models on n Prinz potential trajectories of length 150. Trajectories do not overlap among different trainings. The maximum model lag is 100 steps.

A.8.2 BoPITO samples efficiently long-term dynamics in data-sparse scenarios

Prinz potential 10 models are trained on non-overlapping trajectory sets for different numbers of trajectories. The trajectories length is $10,000$ and the maximum model lag is $1,000$. We consider lags of $10, 25$, and 50 as short, $100, 200$, and 300 as medium, and $500, 750$, and $10,00$ as long time-scales in our experiments, see Figure 8 (b) for a visual reference. $\hat{\lambda}$ is set 0.982 for BoPITO experiments with 1, 5 and 10 trajectories and to 0.994 for the rest.

Alanine Dipeptide 10 models are trained on potentially overlapping random trajectory sets for different numbers of trajectories. The trajectories length is $12,500$ and the maximum model lag is $10,000$. We consider lags of $5, 10$, and 50 as short, $100, 500$, and 1000, as medium, and $2500, 7500$, and 10000 as long time-scales in our experiments, see Figure 9 (b). $\hat{\lambda}$ is set 0.9996 for all BoPITO models.

Chignolin 4 models are trained on potentially overlapping random trajectory sets for different numbers of trajectories. The trajectories length is $35,000$ and the maximum model lag is $30,000$. We consider lags of $10, 50$, and 100 as short, $500, 1000$, and 5000, as medium, and $10000, 20000$, and 30000 as long time-scales in our experiments, see Figure 11 (b). $\hat{\lambda}$ is set 0.99975 for all BoPITO models. BoPITO models are trained using the score 23.

A.8.3 Interpolating between models trained on off equilibrium data and the Boltzmann distribution

We remove the transitions resembling the slowest process in the system, ϕ crossing 0 or 2, to generate biased simulation data. When a transition occurs, we remove one frame before and after the transition and split the trajectory. We train both biased ITO and BoPITO models with a maximum model lag of 100 on the resulting biased dataset. For lags > 100, we sample the BoPITO interpolator fitted on the unbiased dynamic observable defined by eq. 25, see Figure 4. We perform interpolation by sampling with the score model in eq. 15 followed by one round of non-interpolation sampling with lag 100 steps for local relaxation.

A.8.4 Architectural and training and inference details

We use MB-ITO as the architecture of models for the Prinz potential, and SE3-ITO, an $SE(3)-$equivariant neural network, for Alanine Dipeptide, both introduced in Schreiner et al. (2023). We report architectural and training hyper-parameters in Table 1. Models are trained until convergence in the log-log loss plot.

A.9 Prinz potential

The Prinz potential is a 1D potential commonly used for benchmarking MD sampling methods. The potential is defined as

$$U(x) = 4\left(x^8 + 0.8e^{-80x^2} + 0.2e^{-80(x-0.5)^2} + 0.5e^{-40(x+0.5)^2}\right). \tag{24}$$

Diffusion steps	500
Noise schedule	Sigmoidal
Batch size	$2,097,152$
Learning rate	0.001
Layers	3
Embedding dimension	256
Net dimension	256
Optimizer	Adam
Inference ODE steps	50

Diffusion steps	$1,000$
Noise schedule	Polynomial
Batch size	$1,024/32$
Learning rate	0.001
Score layers	5
Embedding layers	2
n_features	64
Optimizer	Adam
Inference ODE steps	100 /50

Table 1: Architectural and training parameters of MB-ITO models (a) and SE3-ITO (b). Batch sizes and inference ODE steps refer to Alanine Dipeptide and Chignolin experiments respectively.

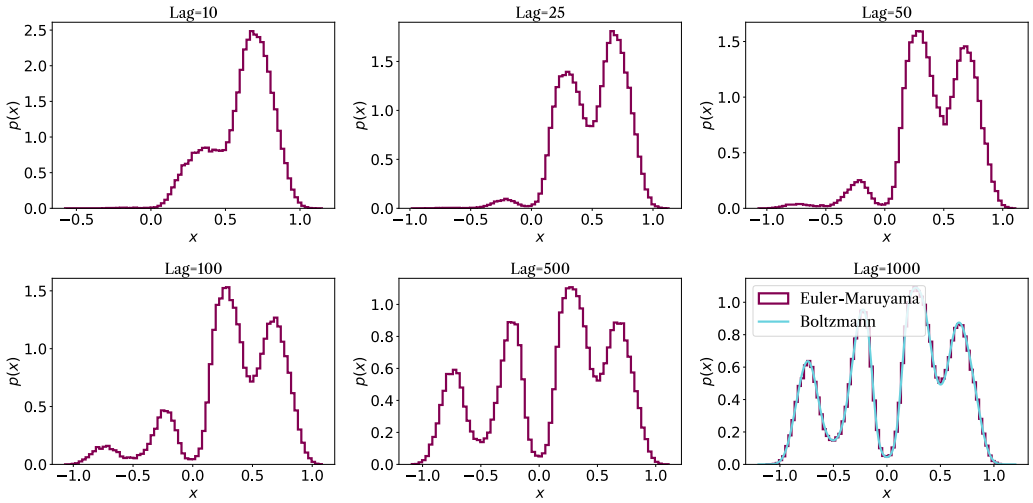


Figure 7: Conditional density of a simulation of the Prinz potential starting at $x = 0.75$ for different lags (steps). Long-term dynamics approaches the Boltzmann distribution.

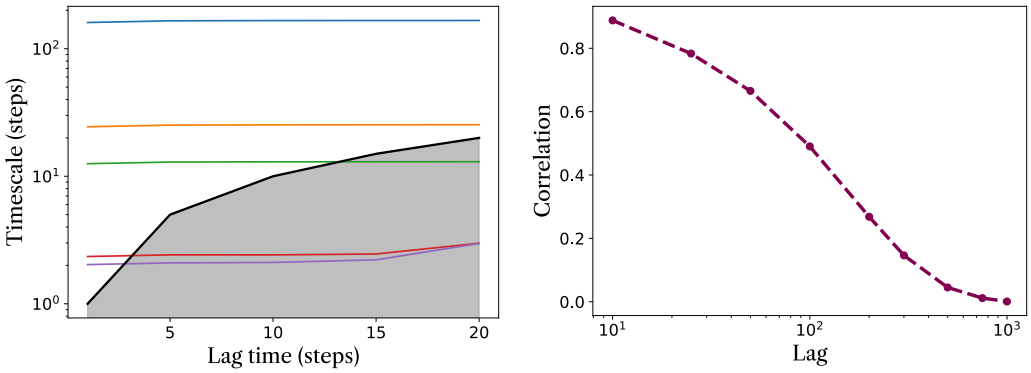
We generate trajectories using an Euler-Maruyama integrator using the library Deeptime (Hoffmann et al., 2021). We set the integrator time-step to $1 \cdot 10^{-5}$, and the temperature, mass, and damping factor to 1. In Figure 7 we show histograms of the position of a particle after N steps, starting from 0.75, and in Figure 8 (a) we report an implied time-scales plot. We use the identity function to define a dynamic observable/correlation function of this system and show it in Figure 8 (b).

A.10 Alanine Dipeptide

Alanine Dipeptide is a small peptide with 22 atoms. We use publicly available data from Dibak et al. (2022), containing a total of 1 μs simulation time split in 20 trajectories. Simulation is performed in an implicit solvent with 2 fs integration time-step and data is saved every 1 ps. We choose,

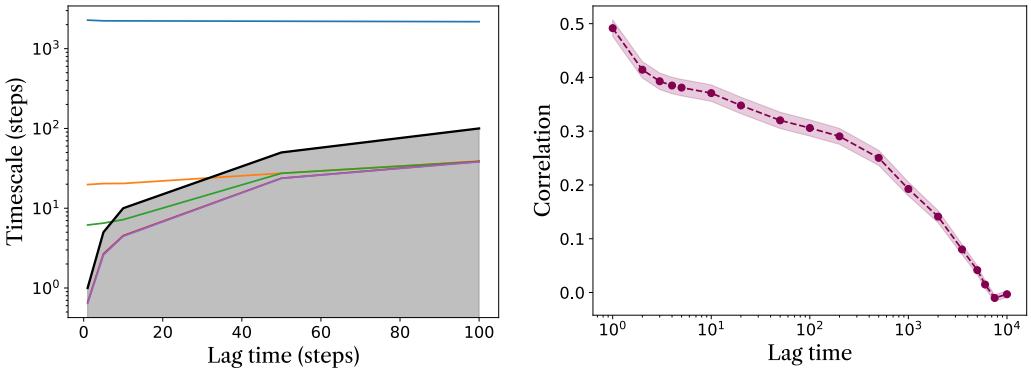
$$a(x) = b(x) = \begin{bmatrix} \sin\phi \\ \cos\phi \end{bmatrix}, \tag{25}$$

to define a dynamic observable of this system. The torsion angle ϕ is involved in the slowest transition observed in the simulation, see Figure 10. We combine these vectors taking the inner product for computing dynamic observables. We show this correlation function in Figure 9 (a), and implied time-scales plot in Figure 9 (b) and a histogram of the torsions ϕ and ψ aggregating all simulation data in Figure 10.



(a) (b)

Figure 8: Implied time-scales (a) and dynamic observable under the identity function (b) for the Prinz Potential. Errors in (b) are smaller than the dots. In (a), the implied time-scales of the 5 slowest processes in the system are computed for different lags. The color order, from longest to shortest implied time-scale, is blue, orange, green, red, and purple.



(a) (b)

Figure 9: Implied time-scales for Alanine Dipeptide computed with a Markov State Model on torsional angles ϕ and ψ (a) and dynamic observable in eq. 25 (b). In (a), the implied time-scales of the 5 slowest processes in the system are computed for different lags. The color order, from longest to shortest implied time-scale, is blue, orange, green, red, and purple.

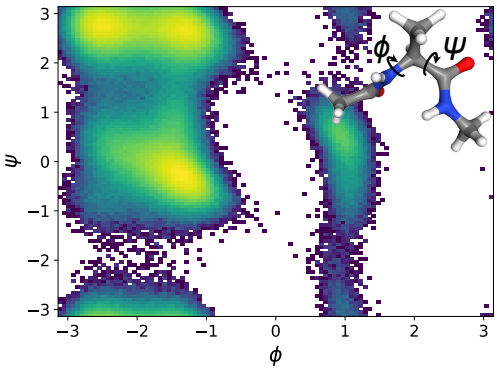
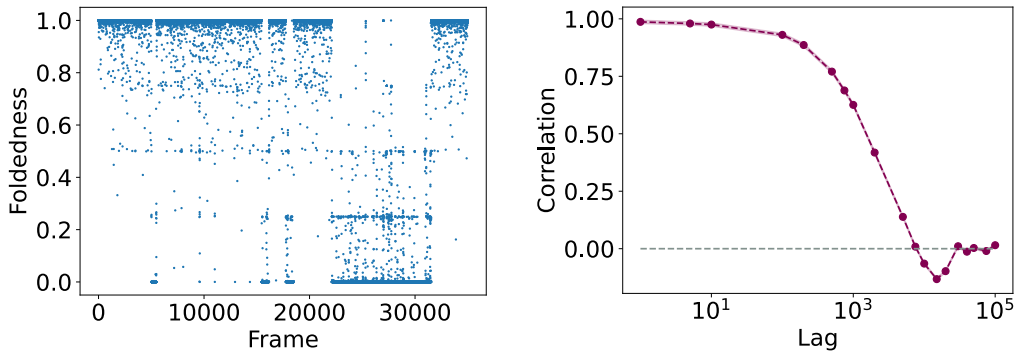


Figure 10: Histogram of the torsional angles ϕ and ψ of Alanine Dipeptide (insert). Data is aggregated among all trajectories in the dataset introduced in Dibak et al. (2022).



(a) (b)

Figure 11: Time evoluation of 'foldedness' in a subset of the reference MD simulation (a) and dynamic observable in eq. 26 (b).

A.11 Chignolin

Chignolin (cln025) is a fast folding protein with 10 residues, 166 atoms and 93 heavy atoms. We use molecular dynamics data previously reported by Lindorff-Larsen et al. (2011). The data is proprietary but available upon request for research purposes. The simulations were performed in explicit solvent with a 2.5 fs time-step and the positions was saved at 200 ps intervals. We extract all heavy atoms positions from the simulations and train models on this data. We use the reaction coordinate also proposed in Lindorff-Larsen et al. (2011) as to define a dynamic observable,

$$a(\mathbf{x}) = b(\mathbf{x}) = \frac{\sum_{i=1}^{N_{\text{res}}} \sum_{j>i}^{N_i} \frac{1}{1+e^{10(d_{ij}(\mathbf{x})-d^*_{ij}-1)}}}{\sum_{i=1}^{N_{\text{res}}} N_i}, \tag{26}$$

where the first sum in the numerator iterates over all N_{res} residues in the protein, while the second sum considers the N_i native contacts of residue i separated by at least seven residues in the primary sequence. We define native contacts as residue pairs separated by at least seven residues in the primary sequence and with C_α atoms closer than 10 Å in the native structure. Moreover, $d_{ij}(\mathbf{x})$ and d^*_{ij} represent the distances between the C_α atoms of residues i and j in the structure $\mathbf{x}$ and the native structure, respectively. This observable quantifies the protein's foldedness, with values ranging from 0 (unfolded) to 1 (folded). In Figure 11 (a) we show the evolution of eq. 26 on a subset of the reference simulation data and we observe how the protein undergoes transformations between the folded and unfolded states. In Figure 11 (b) we visualize the corresponding dynamic observable.

A.12 Metrics

We evaluate models computing differences in the dynamic observables introduced in section 4.1 w.r.t. long MD simulations,

$$|\Delta\text{correlation}|_N = |O^*_N - O^{\text{model}}_N|, \tag{27}$$

where O^*_N is the normalized observable predicted by MD for lag N and O^{model}_N is the model's prediction. See Appendix A.15 for details on normalization. We report the average difference over different training runs.

A.13 Energies for Alanine Dipeptide

In Figure 12 we observe a remarkable agreement in energy densities of samples generated with our Boltzmann Generator and MD. GBSAOBForce is the implicit solvent model energy.

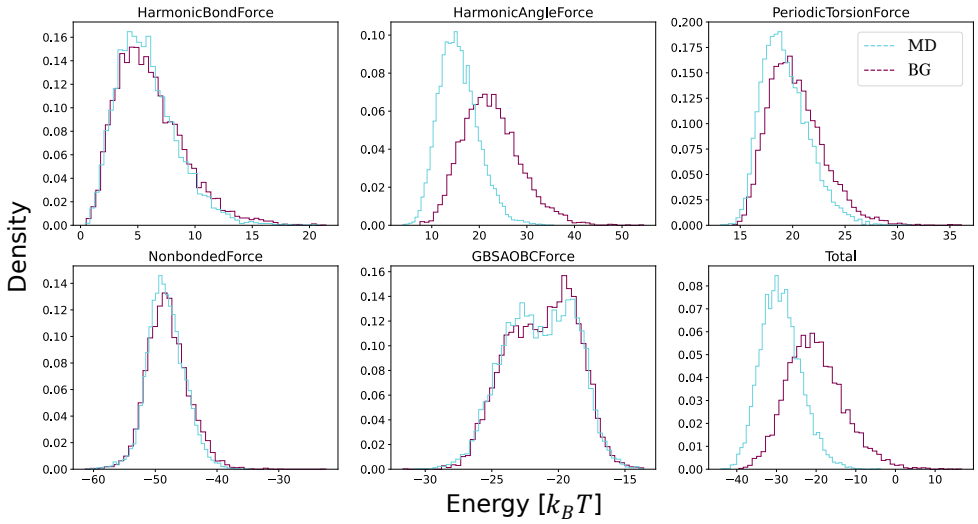


Figure 12: Energy and energy components of samples generated with a Boltzmann Generator (BG) and MD.

A.14 Fitting $\hat{\lambda}$

The hyperparameter $\hat{\lambda}$ controls the time-scale at which the BoPITO model transitions to sampling the equilibrium model. Ideally, $\hat{\lambda}$ should be similar to the eigenvalue corresponding with the slowest process in the system, λ_2. However, its value is not generally available and requires extensive unbiased simulation data to be accurately estimated. If $\hat{\lambda}$ is too small, the model may prematurely relax to equilibrium, limiting its ability to capture non-equilibrium dynamics. Conversely, if $\hat{\lambda}$ is too large, the benefits of the BoPITO framework could diminish. Therefore, careful tuning of $\hat{\lambda}$ can be crucial for optimal performance.

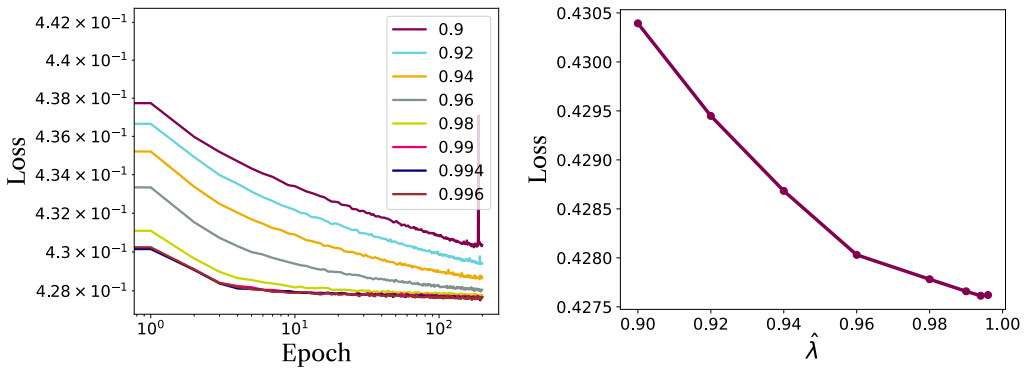
As illustrated in Figure A.14, grid-search hyperparameter tuning can select an appropriate $\hat{\lambda}$ for a single-system model. Too small values of $\hat{\lambda}$ can hinder the model's ability to capture long-time-scale dynamics, leading to increased loss. We recommend choosing the smallest $\hat{\lambda}$ that yields a plateau in the $\hat{\lambda}$-loss curve (elbow rule), as demonstrated in Figure A.14 (b). However, practitioners should be cautious about sampling longer lags than the implied time-scale defined by $\hat{\lambda}$ if they cannot guarantee the system relaxes to equilibrium for those time-scales.

A.15 Correlation function normalization

We subtract the mean and divide by correlation at lag 0 to normalize our dynamic observables. Subtracting the mean guarantees that the correlation function asymptotically decays to 0, and we divide by the correlation at lag 0 so that correlation can be read as a fraction of correlation at time 0. The resulting normalized correlation function is

$$\frac{E[(f(\mathbf{x}_t) - E[f(\mathbf{x}_t)])\,(g(\mathbf{x}_{t+\Delta t}) - E[g(\mathbf{x}_{t+\Delta t})])]}{E[(f(\mathbf{x}_t) - E[f(\mathbf{x}_t)])\,(g(\mathbf{x}_t) - E[g(\mathbf{x}_t)])]}. \tag{28}$$

When comparing different methods, we compute $E[f(\mathbf{x}_t)]$, $E[g(\mathbf{x}_{t+\Delta t})]$ and $E[(f(\mathbf{x}_t) - E[f(\mathbf{x}_t)])\,(g(\mathbf{x}_t) - E[g(\mathbf{x}_t)])]$ using long MD simulations and use these same normalizing factors for all methods.



(a) (b)

Figure 13: Average validation loss against training epochs for the Prinz potential for different values of $\hat{\lambda}$ (a) and average final loss versus $\hat{\lambda}$ (b). Averages are taken w.r.t. 10 runs. Final losses are computed as the average between epochs 190 and 200. The greatest eigenvalue of the system under our simulation parameters is 0.994.