The Journal of Chemical Physics

Ensuring thermodynamic consistency with invertible coarse-graining

Cite as: J. Chem. Phys. **158**, 124126 (2023); https://doi.org/10.1063/5.0141888
Submitted: 09 January 2023 • Accepted: 16 March 2023 • Published Online: 31 March 2023

Shriram Chennakesavalu, David J. Toomer and Grant M. Rotskoff

COLLECTIONS

Note: This paper is part of the JCP Special Topic on Machine Learning Hits Molecular Simulations.

This paper was selected as an Editor's Pick

AIP Publishing

The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

Ensuring thermodynamic consistency with invertible coarse-graining

Cite as: J. Chem. Phys. 158, 124126 (2023); doi: 10.1063/5.0141888
Submitted: 9 January 2023 • Accepted: 16 March 2023 •
Published Online: 31 March 2023

Shriram Chennakesavalu, David J. Toomer, and Grant M. Rotskoff[a)]

AFFILIATIONS

Department of Chemistry, Stanford University, Stanford, California 94305, USA

Note: This paper is part of the JCP Special Topic on Machine Learning Hits Molecular Simulations.
[a)]**Author to whom correspondence should be addressed:** rotskoff@stanford.edu

ABSTRACT

Coarse-grained models are a core computational tool in theoretical chemistry and biophysics. A judicious choice of a coarse-grained model can yield physical insights by isolating the essential degrees of freedom that dictate the thermodynamic properties of a complex, condensed-phase system. The reduced complexity of the model typically leads to lower computational costs and more efficient sampling compared with atomistic models. Designing "good" coarse-grained models is an art. Generally, the mapping from fine-grained configurations to coarse-grained configurations itself is not optimized in any way; instead, the energy function associated with the mapped configurations is. In this work, we explore the consequences of optimizing the coarse-grained representation alongside its potential energy function. We use a graph machine learning framework to embed atomic configurations into a low-dimensional space to produce efficient representations of the original molecular system. Because the representation we obtain is no longer directly interpretable as a real-space representation of the atomic coordinates, we also introduce an inversion process and an associated thermodynamic consistency relation that allows us to rigorously sample fine-grained configurations conditioned on the coarse-grained sampling. We show that this technique is robust, recovering the first two moments of the distribution of several observables in proteins such as chignolin and alanine dipeptide.

I. INTRODUCTION

Biophysical systems evolve with intricately orchestrated dynamics, and even the most subtle molecular motions can inform both their large-scale static and dynamic properties. In most biomolecular systems of interest, there is no reliable way to determine which degrees of freedom can be neglected to obtain an effective model that makes predictions in quantitative agreement with atomistic models. The coupling of both time and spatial scales creates inherent challenges for molecular simulation: many phenomena we would like to simulate, such as protein conformational change,[1,2] protein folding,[3] and multicomponent self-assembly,[4] occur rarely, requiring simulations far too costly for even the most powerful computers. The need to access conformational dynamics on very long timescales has spurred the development of many accelerated sampling methods, which can ameliorate this issue. However, these methods typically require defining a low-dimensional coordinate of interest and sampling that low-dimensional space exhaustively.

Coarse-graining, also known as dimensionality reduction, is intended to provide a model of reduced complexity that can be used, in principle, to accelerate sampling.[2,5–9] For biophysical systems, coarse-grained models are typically developed intuitively by assigning groups of atoms within a molecule to a fixed "bead" that represents a salient substructure.[5] With this representation, the coarse-grained model can be parameterized with a potential energy function of essentially the same functional form as that of the original molecular model. This strategy has been enormously successful in various contexts.[1,3,10] However, due to the dimensionality reduction, there are questions that simply cannot be answered using a coarse-grained model, no matter how accurately it has been parameterized. For example, it is not possible to compute any observables that depend on atomistic detail that has been projected out, nor is it possible to accurately compute energetic information without an auxiliary model.[11]

In this work, we ask if it is possible to accelerate dynamics through dimensionality reduction while maintaining the ability to evaluate equilibrium averages of observables defined on the

fine-grained system with quantitative accuracy. Although other works have sought to invert the coarse-grained representation, generally, they do so in a data-driven way that does not rigorously yield a physical distribution of states.[7,12] To carry out this procedure, we combine an embedding strategy based on the hierarchical dimensionality reduction for graph data[13] with a backmapping procedure that allows us to rigorously sample Boltzmann-weighted configurations of the fine-grained system conditioned on the coarse-grained configurations. This strategy ensures that we can evaluate averages of arbitrary atomistic observables, some of which cannot even be defined for the corresponding coarse-grained model.

To construct the coarse-graining map, we do not just specify *a priori* how to embed the atomic coordinates in a low-dimensional space, we instead optimize this mapping. This change requires a distinct paradigm for coarse-graining, in which we use a state-dependent embedding map that allows for a more flexible, but a nonlinear representation of the coarse-grained space. We simultaneously train the coarse-graining map, its associated potential energy function in the coarse-grained space, and a map that inverts the low-dimensional configurations and conditionally generates new fine-grained structures.

Using generative machine learning models to find low-dimensional representations has shown success in biomolecular systems,[14–19] but inverting a coarse-graining map requires a specific structure to ensure that sampling will asymptotically converge to the fine-grained Boltzmann distribution. We represent the inversion map using normalizing flows,[20–22] which have shown promise for augmenting Markov chain Monte Carlo (MCMC) sampling.[16,23–25] Although normalizing flows are challenging to optimize for sampling high-dimensional distributions with multiple metastable states,[15,24] the conditioned sampling procedure that we employ is considerably easier to train.

This work weaves together many threads being pursued independently in the machine learning literature and molecular simulation, including the use of autoencoders for dimensionality reduction, generative models for sampling, and neural networks for free energy surfaces, as mentioned above. However, we see the present work as not the introduction of an algorithm or computational procedure but rather a conceptual development in coarse-grained modeling. We believe that allowing for a more complicated, less interpretable coarse-grained space provides new opportunities to accelerate sampling in the fine-grained space. The extended notion of weak thermodynamic consistency that we introduce provides a framework on which to build new coarse-graining strategies that are targeted to particular classes of observables, which may allow for more efficient models for precise scientific questions.

II. RELATED WORK

The approach that we take here incorporates the optimization of a coarse-graining map, backmapping of the coarse-grained configurations, and new criteria for evaluating the quality of sampling. Each of these ideas individually has attracted substantial attention. For example, automated construction of coarse-graining maps has a long history,[26–29] and approaches based on determining the "essential dynamics" by identifying collective modes via the principal component analysis[30] or ranking maps based on information-theoretic criteria.[31,32] The process of inverting a coarse-graining map, or backmapping, has been studied in several contexts;[12,14,33,34] these works use either deterministic mappings or non-invertible generative models. As a result, it is impossible to exactly reweight the backmapped structures relative to the target Boltzmann distribution, which is the central desideratum of our algorithm.

Because we compute observables on the backmapped structures, which have the full molecular detail, we introduce a criterion for evaluating the quality of our sampling that is based on the notion of weak convergence from probability theory. Like our framework, previous studies have sought to compute fine-grained observables directly from coarse-grained structures.[11,35–39] Many of these works focus on an approach to mitigate the *representability problem*, which arises when the coarse-grained analog of a fine-grained observable does not naturally coincide with a conditional average of the fine-grained observable, for details cf. Ref. 39. This does not arise in our work due to the inversion and reweighting—we only evaluate the fine-grained observables.

Taken together, these works highlight the pressing need for systematic methods to incorporate generative models into sampling and coarse-graining with high data efficiency.[40] This need becomes especially acute when the coarse-graining map is a learned embedding and may yield a less interpretable coarse-grained space.

III. WEAK FORMULATION OF THERMODYNAMIC CONSISTENCY

Atomic resolution molecular models of biophysical systems can provide detailed and accurate insight into the static and dynamic properties of biomolecules, provided that there are sufficiently powerful computational resources to collect a statistically representative sample of configurations of an n-particle system, $\{\boldsymbol{x}_i\}_{i=1}^n$ with each configuration $\boldsymbol{x}_i \in \mathbb{R}^{3n}$. Due to ergodicity, an MD simulation in the canonical ensemble samples a Boltzmann distribution, and the probability of a given configuration is given by the familiar expression,

$$\rho(\boldsymbol{x})d\boldsymbol{x} = Z^{-1}e^{-\beta U(\boldsymbol{x})}d\boldsymbol{x}, \tag{1}$$

where Z is the partition function, $\beta = \frac{1}{k_B T}$ is the reduced inverse temperature, and $U : \mathbb{R}^{3n} \to \mathbb{R}$ is the fine-grained potential energy function.

When we coarse-grain a molecular system, we reduce the dimensionality and necessarily destroy the information. Hence, when carrying out this destructive process, we should ideally preserve the most important degrees of freedom required to describe the fluctuations of the system. In landmark work, Noid *et al.*[2] established the notion of *thermodynamic consistency* to provide a formal description of the requirements of a "good" coarse-graining map. In their formulation, we require equivalence between the potential of mean force $\hat{F}$ and the parametric coarse-grained potential $\hat{U}(\cdot, \boldsymbol{\theta})$,

$$\hat{F}(\boldsymbol{z}) \equiv -\beta^{-1} \log \int_\Omega e^{-\beta U(\boldsymbol{x})}\delta(\mathcal{M}(\boldsymbol{x}) - \boldsymbol{z})d\boldsymbol{x} \leftrightarrow \hat{U}(\boldsymbol{z}, \boldsymbol{\theta}), \tag{2}$$

and we emphasize that this equivalence is up to an arbitrary additive constant. Here, $\mathcal{M}$ is the coarse-graining map. The function $\hat{U}(\cdot, \boldsymbol{\theta})$ depends on the parameter $\boldsymbol{\theta}$ that is optimized so that the gradients of $\hat{U}$ match the gradients of $\hat{F}$. Throughout, we use $\hat{\cdot}$ to designate a function defined on the coarse-grained domain. This ensures that

canonically distributed samples of coarse-grained configurations sampled in proportion to

$$\hat{\rho}(\boldsymbol{z},\boldsymbol{\theta})d\boldsymbol{z} := \hat{Z}^{-1}e^{-\beta\hat{U}(\boldsymbol{z},\boldsymbol{\theta})}d\boldsymbol{z}, \tag{3}$$

will recover the projected distribution. The key observation is that equivalence is defined in the coarse-grained space.

The requirement of thermodynamic consistency is a stringent one in the sense that it enforces the equivalence at the level of the distribution rather than, for example, the equivalence of some collection of observables. Of course, there are other potentially useful notions of equivalence for probability distributions that could lead to additional flexibility in the procedure. Here, we take inspiration from the measured theoretic notion of weak convergence, which quantifies the difference between distributions through expectations (or average values) of bounded, continuous functions. Importantly, this notion could be applied in either the coarse-grained space or the fine-grained space to optimize and test a coarse-graining map. Throughout, we measure "weak thermodynamic consistency" in the fine-grained space. We refer to a coarse-graining map $\mathcal{M}$ and the associated potential $\hat{U}$ as "$\mathcal{F}$ thermodynamically consistent" if for every observable $f \in \mathcal{F}$,

$$\int f(\boldsymbol{x})p_{\text{inv}}(\boldsymbol{x}|\boldsymbol{z})\hat{\rho}(\boldsymbol{z},\boldsymbol{\theta})\,d\boldsymbol{x}d\boldsymbol{z} \longrightarrow \int f(\boldsymbol{x})\rho(\boldsymbol{x})\,d\boldsymbol{x}. \tag{4}$$

In this expression, $p_{\text{inv}}(\boldsymbol{x}|\boldsymbol{z})$ is the conditional probability of generating $\boldsymbol{x}$ from a coarse-grained configuration $\boldsymbol{z}$—finding a map that performs this inversion in a way that is suitable to reweighting or Monte Carlo is a central goal of the present work and is discussed at length in Sec. V. This definition of thermodynamic consistency differs from Ref. 2 because we only require equivalence on some set of observables $\mathcal{F}$, which could be adapted to a particular problem.

In Appendixes A–H, we prove the following straightforward proposition, which relates weak thermodynamic consistency to the definition introduced by Noid *et al.*[2]

Proposition III.1. An invertible coarse-graining is a tuple $(\mathcal{M},\hat{U},T)$ *consisting of a coarse-graining map* $\mathcal{M}:\mathbb{R}^{3n}\to\mathbb{R}^{3k}$, *a coarse-grained potential* $\hat{U}:\mathbb{R}^{3k}\to\mathbb{R}$, *and a normalizing flow* $T:\mathbb{R}^{3n}\to\mathbb{R}^{3n}$. *Let* $\mathcal{F}_*$ *denote the set of functions of continuous, bounded functions,*

$$\mathcal{F}_* := \left\{ f \in \mathcal{C}(\mathbb{R}^{3n},\mathbb{R}) \middle| f = \hat{f}\circ\mathcal{M}, f \neq 0 \ \forall \hat{f} \neq 0, \hat{f}:\mathbb{R}^{3k}\to\mathbb{R} \right\},$$

which denotes the collection of observables that are nontrivial on the coarse-grained space. If $(\mathcal{M},\hat{U},T)$ *is* $\mathcal{F}_*$ *thermodynamically consistent, then the projective coarse-graining* $(\mathcal{M},\hat{U})$ *is thermodynamically consistent in the sense of Ref. 2.*

At a high level, this statement says that if weak thermodynamic consistency holds for all observables not orthogonal to the coarse-graining map, then the projective coarse-graining is thermodynamically consistent.

IV. STATE-DEPENDENT PROJECTION MAPPING

Neural networks construct a nonlinear embedding of the input data, which is often viewed as constructing a basis in which regression or classification can be performed easily.[43,44] The fact that learned embeddings provide a highly efficient dimensionality reduction has been exploited in molecular contexts, including for reaction coordinates[45,46] and searching through chemical space.[47] Nevertheless, this strength has not been thoroughly explored in the context of coarse-graining, despite the fact using that autoencoders in the context of the reaction coordinate discovery emphasizes that machine learning is well-suited to finding low-dimensional representations capable of capturing slow degrees of freedom.[46,48–50] Instead, typically the coarse-graining map is specified at the outset based on physical intuition.

In this work, we learn the coarse-graining map, choosing only the dimensionality of the coarse-grained space (see Fig. 1). Although we have developed a framework for training a coarse-graining map that is an arbitrary nonlinear function (Appendix B), all results presented here use a state-dependent linear projection map that is itself a nonlinear function of the atomic positions. We train the model to find optimal projections using a reconstruction loss scheme described in detail in Appendix B. Our training procedure resembles the typical paradigm for autoencoders,[51] adding auxiliary loss functions to regularize the learned representation and favor locality in the coarse-grained mapping.

The embedding we use is depicted schematically in Fig. 2. Our approach is based on differentiable pooling (`DiffPool`), an algorithm developed for parameterizable graph coarsening.[13] We use a pooling layer that consists of an equivariant graph neural network that outputs a projection matrix $P_{\boldsymbol{x}}$, given a fine-grained configuration $\boldsymbol{x}$. Thus, although the projection matrix is a nonlinear function of the input coordinates, the representation in the coarse-grained

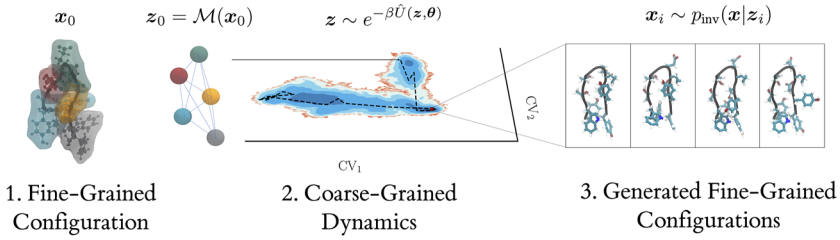

FIG. 1. A schematic overview of the coarse-graining procedure. First, a fine-grained molecular structure is embedded with a state-dependent learned projection $\mathcal{M}$. A coarse-grained potential $\hat{U}$ is used to sample coarse-grained configurations $\boldsymbol{z}$ so that they are distributed according to a Boltzmann distribution with respect to $\hat{U}$, as described in Sec. IV. These samples are subsequently used to conditionally sample fine-grained configurations as described in Sec. V.

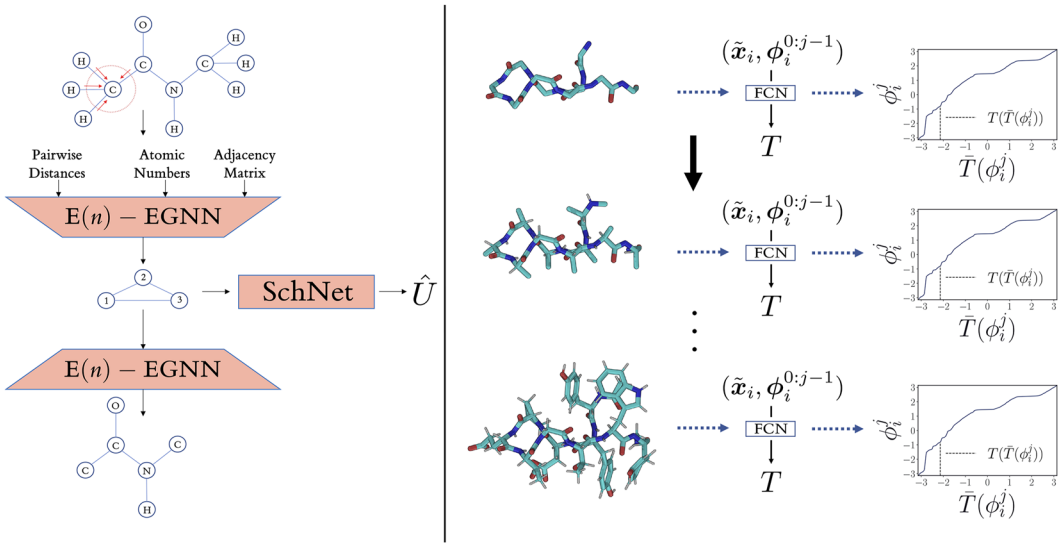

FIG. 2. Schematic of the computational pipeline for coarse-graining (left) and backmapping (right). Left. Configurations are shown as 2-D graphs for ease of visualization. Initial configuration $\boldsymbol{x}_i$ is passed through a coarse-graining neural network based on the $\mathrm{E}(n)$ − EGNN graph neural network architecture[41] to determine the corresponding coarse-grained configuration $\mathbf{z}_i$. Coarse-Grained configuration is passed through a decoding network to reconstruct a set of target atoms $\tilde{\boldsymbol{x}}_i$. Coarse-grained potential energy $\hat{U}$ is computed as a function of $\mathbf{z}_i$ and is represented using a neural network based on the SchNet[42] architecture. Right. With an autoregressive rational-quadratic neural spline flow (RQ-NSF), internal coordinates ϕ_i^j are iteratively generated conditioned on $\tilde{\boldsymbol{x}}_i$ and $\phi^{0:j-1}$. These internal coordinates are then used to sequentially reconstruct the entire fine-grained structure.

space is a linear transformation of the input coordinates. This means that we can train the potential energy function for the coarse-grained model with a typical force-matching objective.[9] In our examples, with the auxiliary loss functions we employ, we find that the coarse-graining map is typically only weakly state-dependent, as depicted in the first panel of Fig. 1. Admittedly, the representation of the coarse-grained configuration becomes more difficult to interpret due to the complicated relationship between the fine-grained configuration and the resulting projection matrix. The loss of interpretability instead motivates us to invert the coarse-graining map directly.

V. INVERTING THE COARSE-GRAINED SAMPLES

We often use molecular simulations to investigate properties or observables $f : \mathbb{R}^{3n} \to \mathbb{R}$ that *require* atomic resolution. Such observables cannot be mapped onto a coarse-grained configuration, and finding appropriate proxies in the coarse-grained space for a detailed molecular property is challenging in general—there is no unique strategy. Moreover, the strategy we use to embed molecular configurations is not amenable to a physical interpretation. In this work, we carry out a two-stage process that allows us to reconstruct fine-grained configurations while also leveraging the reduced dimensionality of the coarse-grained system to accelerate the exploration of the free energy landscape. We first sample the collections of coarse-grained configurations in proportion to the coarse-grained Boltzmann probability $\hat{p}(\boldsymbol{z})d\boldsymbol{z}$. We subsequently employ an invertible neural network, known as a normalizing flow, to harvest a statistical sample of fine-grained configurations in such a way that we can reweight fine-grained configurations in proportion to their true Boltzmann weight.

It requires care to ensure that the sampling and subsequent reconstruction can be combined to obtain the correct statistics for the target Boltzmann distribution of the fine-grained system. In general, there are two options: Metropolis Monte Carlo or reweighting. To do so, we must be able to compute the exact likelihood of generating a fine-grained configuration—this depends on both the probability of a coarse-grained configuration in our sampling scheme and the conditional probability of generating the fine-grained configuration from a coarse-grained configuration. We elaborate on how to compute both contributions to the likelihood below.

Sampling the coarse-grained system with the Metropolis-adjusted Langevin Algorithm (MALA) leads to a collection of coarse-grained configurations $\{\boldsymbol{z}_i\}_{i=1}^k$ that are distributed in proportion to the Boltzmann distribution associated with the coarse-grained energy function, that is $\hat{p}(z) \propto \exp\left(-\beta\hat{U}(z, \boldsymbol{\theta})\right)$. We use the conventional force-matching paradigm to train the parameters $\boldsymbol{\theta}$ of $\hat{U}$, for which we use a standard implementation of the SchNet architecture;[42] the framework is obviously extensible to other neural networks. We train the energy function so that it reconstructs the potential of mean-force $\hat{F}$ associated with the fine-grained distribution. Although the coarse-grained energy function $\hat{U}$ is often represented with an empirical potential functional form, recently more general functional forms have been employed, not only using traditional empirical potentials but also adding a general neural network.[8,9] Because the optimization of the force-matching objective relies on data collected from fine-grained molecular simulations, the coarse-grained potential will not exactly match the potential of mean force $\hat{F}$. We note that because we are only focused on generating samples, not accurately recapitulating dynamical information, we do not need to incorporate a spatially dependent diffusion tensor.[52]

We invert the coarse-grained representation in two steps; first, we map the coarse-grained configuration via a linear back-projection decoding map $\mathcal{M}^{\text{dec}} : \mathbb{R}^{3k} \to \mathbb{R}^{3\tilde{n}}$. In our examples, the backbone is reconstructed from this map. We then conditionally sample fine-grained configurations $\boldsymbol{x}$ using a normalizing flow T, which parameterizes a conditional distribution $p_{\text{inv}}(\boldsymbol{x}|\boldsymbol{z})$. The normalized value of this conditional probability is required to exactly reweight or metropolize, which is only tractable with a normalizing flow neural network architecture.[21] Reweighting or metropolization yield Boltzmann-distributed configurations, even when the normalizing flow does not exactly generate a Boltzmann distribution.

Viewed as an MCMC algorithm, the generation probabilities are explicit functions of the generated configuration and do not depend on the previous configuration within the Markov chain. Given the current state of the Markov chain, $\boldsymbol{x}_i$, we can write the probability of generating a new state $\boldsymbol{x}_{i+1}$ as

$$p_{\text{gen}}(\boldsymbol{x}_{i+1}|\boldsymbol{x}_i) = \hat{p}(\boldsymbol{z}_{i+1}) p_{inv}(\boldsymbol{x}_{i+1}|\boldsymbol{z}_{i+1}) d\boldsymbol{z}_{i+1}, \tag{5}$$

where the conditional distribution can be computed using the push-forward probability density. The pushforward density is given by inverting the map and evaluating its probability in the Gaussian density,

$$T\sharp\rho(\boldsymbol{x}) = \rho(\tilde{T}(\boldsymbol{x}))|\nabla\tilde{T}(\boldsymbol{x})|, \tag{6}$$

where $|\nabla\tilde{T}(\boldsymbol{x})|$ denotes the determinant of the Jacobian of the inverse of the normalizing flow T, and ρ is the density of a Gaussian with mean zero and identity covariance. The architecture of the neural network T is constructed so that all the terms above are easily computable.[53]

VI. RESULTS

A. Alanine dipeptide

Alanine dipeptide is a standard benchmark for molecular simulation; its dynamics is well-described by two dihedral angles ϕ and ψ.[54,55] We carried out an MD simulation of alanine dipeptide in explicit solvent for a total duration of over 0.5 μs. From this trajectory, we generated a dataset by sub-sampling 50 000 data points, consisting of positions and forces on all 22 atoms of alanine dipeptide. With this sub-sampling approach, we automatically coarse-grain all water molecules out before training. Using this dataset, we trained a coarse-graining map $\mathcal{M}$, a coarse-grained potential energy $\hat{U}$, and a normalizing flow to carry out backmapping using the procedure described in Secs. V and VII and Appendixes B and C.

Our coarse-graining map $\mathcal{M}$ projects the 22 atoms from the fine-grained configuration to a coarse-grained configuration consisting of 6 beads. Although we allow for state-dependent embeddings to be learned, we observe that the embeddings are the same across all fine-grained configurations in our dataset. Furthermore, we observe that the learned coarse-graining map projects the five backbone heavy atoms and the C_β onto the coarse-grained space. Importantly, this learned map is consistent with physical intuition and with other coarse-graining works that investigate alanine dipeptide.[8,9]

Using the learned coarse-grained potential $\hat{U}$, we run Langevin dynamics to carry out sampling in the coarse-grained space. We ran 15 trajectories with different initial points across the different metastable basins and sampled 20 000 coarse-grained configurations from these trajectories. Finally, for each of these 20 000 configurations, we carry out our backmapping procedure and generate 2000 fine-grained configurations for a total of 4×10^7 fine-grained configurations. Of course, there is no guarantee that the collection of fine-grained generated configurations is Boltzmann-distributed; to ensure this, we carry out a reweighting procedure followed by a single step of overdamped Brownian dynamics (see Appendix F).

We plot the free-energy surface as a function of the ϕ and ψ dihedral angles in Fig. 3. Interestingly, we see that in comparison to the free-energy surface obtained from MD simulation, the generated configurations before reweighting greatly oversample one of the basins; however, with reweighting, the free energy surface of generated configurations very closely approximates the free-energy basin obtained via MD simulation and is in strong qualitative agreement with free-energy surfaces documented in the

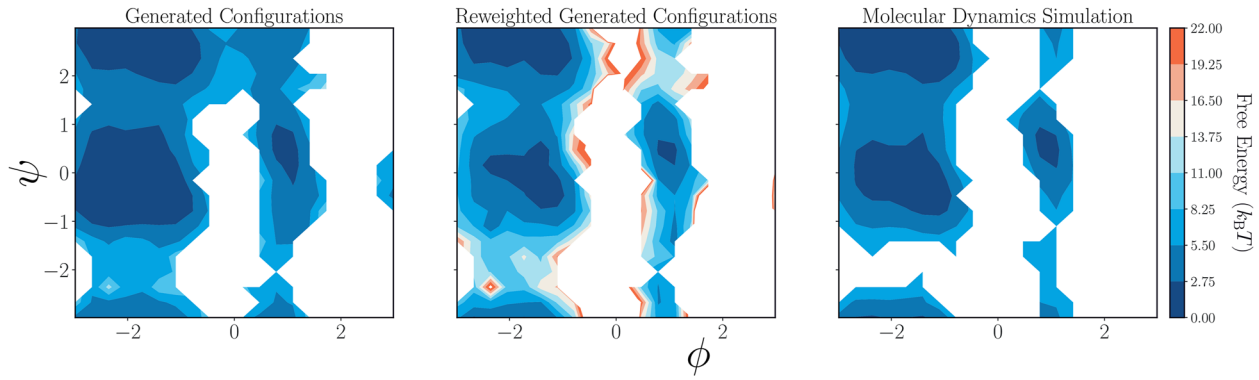

FIG. 3. Free-energy landscape of alanine dipeptide as a function of ϕ and ψ dihedral angles. Generated configurations from coarse-grained simulations backmapped into fine-grained configurations (left). Generated configurations from coarse-grained simulations backmapped into fine-grained configurations with reweighting to ensure configurations are Boltzmann-distributed (center). The training dataset consists of configurations sampled via MD simulation (right).

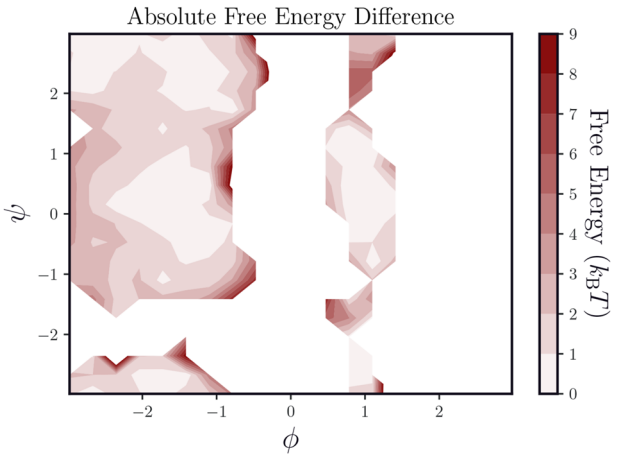

FIG. 4. Absolute error between free energy from MD and inverted CG sampling for alanine dipeptide. The error is less than 1 k_BT on all metastable states and is only appreciable in very rare regions that are undersampled by the MD.

literature.[56] Figure 4 illustrates the extent of quantitative agreement between a long MD simulation and the reweighted generated configurations.

Finally, we examine two observables that we are unable to compute for a coarse-grained configuration (Fig. 5). First, we look at the potential energy computed via an implicit solvent model for the configurations in our original dataset and our reweighted-generated configurations and observe that the mean and variance agree quantitatively. Next, we considered the dihedral angle that was computed using two adjacent methyl-hydrogens and the two nearest backbone atoms (see Fig. 5 for an illustration). We plot the combined distributions obtained from all six of these dihedral angles computed for the two methyl caps at the termini of the protein. We are unable to compute these angles directly from the coarse-grained configurations: backmapping is essential. Again, we observe quantitative agreement between the distribution of these dihedral angles with our MD dataset and the reweighted generated configurations.

B. Chignolin

Next, we carried out a similar analysis for the CLN025 variant of chignolin, a fast-folding micro-protein. Chignolin is a ten-residue protein that adopts a β-hairpin structure in its folded state.[57] Unlike alanine dipeptide, chignolin does not have a set of physical collective variables that can aptly characterize the conformational dynamics. Instead, we utilize the time-lagged independent component analysis (tICA) to determine the necessary collective variables.[58,59] Using a publicly available trajectory of chignolin simulated in an explicit solvent,[9] we collated a dataset consisting of 50 000 positions and forces on all 175 atoms of chignolin. We again coarse-grain out all water molecules before training. As with alanine dipeptide, we train a $\mathcal{M}$, $\hat{U}$, and a normalizing flow using the procedure described in Secs. V and VII and Appendixes B and C.

For chignolin, the coarse-graining map $\mathcal{M}$ projects the 175 atoms in the fine-grained configuration to a coarse-grained configuration consisting of 30 beads. As with alanine dipeptide, we observe that the state-dependent embeddings weakly depend on the fine-grained configurations in our dataset. Furthermore, we observe that the coarse-graining map learns to project backbone atoms (C, C_α, N) onto the coarse-grained space, again learning an intuitive and physically meaningful map.

With our trained $\hat{U}$, we carried out coarse-grained dynamics consisting of 27 trajectories with different initial points. From these coarse-grained trajectories, we backmapped 54 000 coarse-grained configurations, where we generated 750 fine-grained configurations per coarse-grained configuration. The generated configurations were generally high-energy configurations; however, this was generally a result of minor structural deformities as opposed to major flaws in the reconstruction procedure. To alleviate this, we carry out short overdamped Brownian dynamics to relax the structure. Finally, we carried out a reweighting step to ensure that the configurations were Boltzmann-distributed (see Appendix F).

We plot the free-energy surface as a function of the two leading tICA coordinates (see Fig. 6), where we determined the tICA coordinates according to the procedure detailed in Ref. 9. The bottom left (first) basin corresponds to the unfolded state, the bottom right (second) basin corresponds to the folded state, and the top (third) basin corresponds to the misfolded state. In comparison to the free-energy surface of the dataset obtained via MD simulation, the free-energy

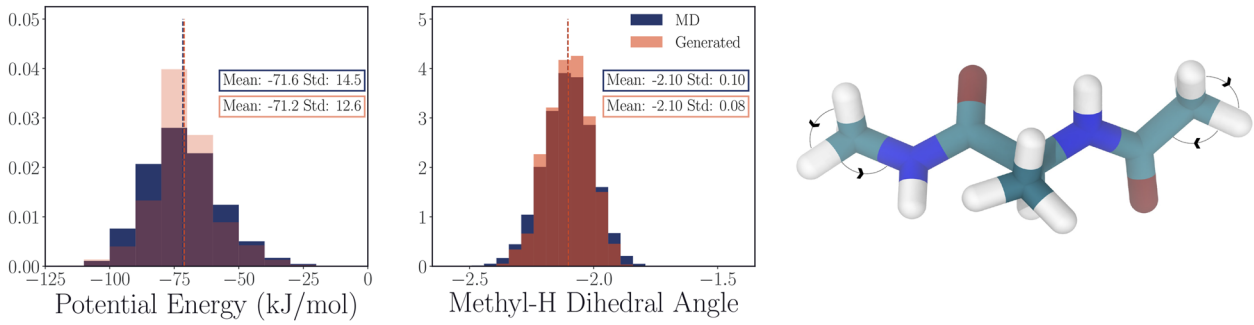

FIG. 5. Potential energy of alanine dipeptide computed via an implicit model (left) and dihedral angle involving all six pairs of adjacent methyl-hydrogens in N-terminus and C-terminus methyl groups (right) for reweighted generated configurations (orange) and training dataset obtained via MD (blue). The sample configuration on right illustrates the dihedral angle.

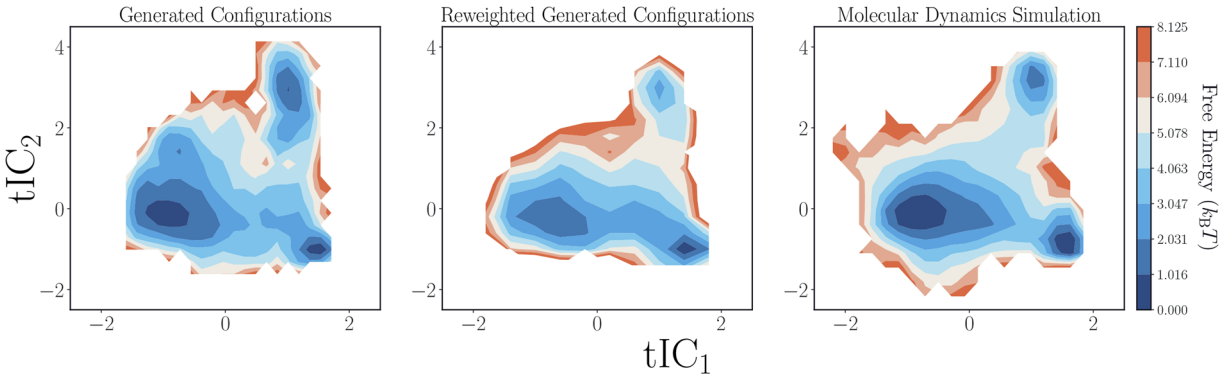

FIG. 6. The free-energy landscape of chignolin as a function of time-lagged independent coordinates (tICs). Generated configurations from coarse-grained simulations backmapped into fine-grained configurations (left). Generated configurations from coarse-grained simulations are backmapped into fine-grained configurations with reweighting to ensure that configurations are Boltzmann-distributed (center). The training dataset consisting of configurations sampled via MD simulation (right).

surface of the generated configurations (before reweighting) is heavily populated in regions adjacent to the basins. However, we observe that with reweighting, the free-energy surface of generated configurations more closely matches that of the free-energy surface obtained from MD simulation.

Finally, we compare the two observables that we cannot compute with a coarse-grained configuration. As with alanine dipeptide, we use an implicit solvent model to compute potential energies and observe that the distribution of potential energies from our MD dataset of configurations closely matches the distribution of potential energies of our generated configuration after reweighting (Fig. 7). Finally, the hydrophobic side-chains are strongly implicated in the folding of chignolin;[60] we probe the rotameric states of the bulky tryptophan (Trp) residue in chignolin to assess the ability of our backmapping procedure to faithfully generate the appropriate rotamers. The conformation of the Trp residue is highly dependent on the overall conformation of the protein (i.e., unfolded, folded, or misfolded), and with our coarse-grained representation, we are unable to investigate the nature of the rotameric states of the Trp residue. For our reweighted generated structures, we analyze the rotameric state of tryptophan using the dihedral angle between the O, C, C_α, and C_β atoms of the Trp residue. From the MD dataset, it is

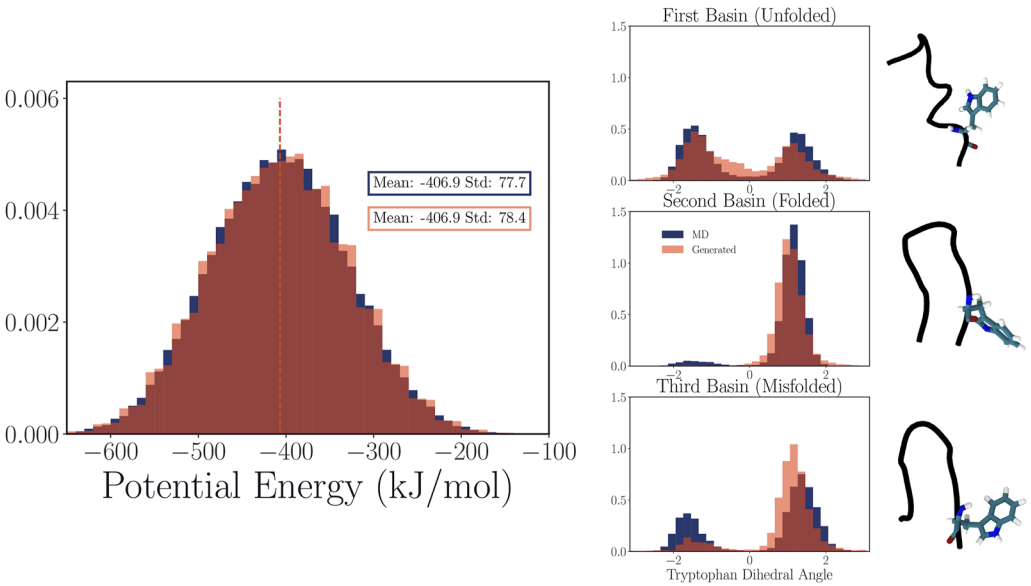

FIG. 7. Potential energy of chignolin computed via an implicit model (left) and dihedral angle (between O, C, C_α, and C_β) of the bulky tryptophan side chain for three different basins of chignolin (right) for reweighted generated configurations (orange) and training dataset obtained via MD (blue). Configuration sampled from reweighted generated configurations for each basin shown, with only backbone and tryptophan residue shown.

clear that the distribution of the dihedral angle of the Trp residue is highly dependent on the global conformational state of the protein. We observe that our generated structures are able to closely, but not perfectly, approximate the rotameric distribution of the Trp residue from the MD dataset. Ultimately, we believe that this is a limitation of either the normalizing flow architecture or the decoder; we anticipate that with improved models, it will be possible to more closely realize the true distribution of rotameric states.

VII. METHODS

We have developed a computational approach that consists of three interdependent components: a graph neural network representing the coarse-graining map $\mathcal{M}: \mathbb{R}^{3n} \to \mathbb{R}^{3k}$, a neural network representing the coarse-grained potential energy function $\hat{U}: \mathbb{R}^{3k} \to \mathbb{R}$, and a normalizing flow $T: \mathbb{R}^{3n} \to \mathbb{R}^{3n}$ that can generate configurations in the fine-grained space conditioned on a coarse-grained configuration. We work under the assumption that an effective coarse-graining $\mathcal{M}$ integrates out degrees of freedom that relax on short timescales while retaining pertinent information from slower-moving degrees of freedom that inform the free-energy landscape. Additionally, we consider an effective $\mathcal{M}$ to be one for which we can train a coarse-grained potential $\hat{U}$ that closely matches the potential of mean force $\hat{F}$ with a necessarily finite dataset. To account for this multi-task objective, we consider a cyclic training scheme that alternates between training $\mathcal{M}$ and $\hat{U}$, where $\mathcal{M}$ informs the training of $\hat{U}$ and vice versa. Finally, we train the normalizing flow independently of the training of $\hat{U}$ and $\mathcal{M}$.

We represent our configurations using a three-dimensional graph, where nodes correspond to atoms and edges between nodes correspond to bonded and nonbonded interactions. The graph structure is then coarse-grained through a clustering process, where each cluster corresponds to a coarse-grained "bead" and consists of a weighted combination of a collection of nodes. This is achieved via a hierarchical graph pooling technique based on `DiffPool`[13] using the E(n) – EGNN graph neural network architecture, which imposes necessary rotational and physical invariance constraints.[41] The resulting embedding is state-dependent; the exact clustering is dependent on the input molecular configuration. We represent our coarse-grained potential energy $\hat{U}$ using the SchNet architecture,[42] which expands inter-bead distances into Gaussian basis functions with learnable parameters; these Gaussian basis functions are then passed through multiple neural network layers. This architecture similarly imposes the necessary physical constraints, most notably rotational and translation invariance.

To train $\mathcal{M}$, we consider the encoding-decoding tasks of learning an embedding to determine a coarse-grained configuration $\boldsymbol{z} = \mathcal{M}(\boldsymbol{x}) \in \mathbb{R}^{3k}$, which can then be used to accurately reconstruct a set of target atoms $\tilde{\boldsymbol{x}} = \mathcal{M}^{\text{dec}}(\boldsymbol{z}) \in \mathbb{R}^{3\tilde{n}}$ from the original fine-grained configuration $\boldsymbol{x} \in \mathbb{R}^{3n}$. The decoding process is achieved via "inverse pooling" that is similarly predicated on `DiffPool`[13] with the E(n) – EGNN architecture.[41] The target atoms that are reconstructed using the decoder need to be specified *a priori*. In practice, this requires limited knowledge of the system. For the two systems we consider here, we designate the backbone as the set of target atoms to reconstruct.

We first train our coarse-graining network via a reconstruction loss and a suite of auxiliary losses (Appendix B). The reconstruction loss is the mean squared deviation (MSD) between $\mathcal{M}^{\text{dec}}(\mathcal{M}(\boldsymbol{x}))$ and $\tilde{\boldsymbol{x}}$. With an initial coarse-graining map, we then train $\hat{U}$ using a mean-force-matching scheme,

$$\mathcal{L}_{\text{mf}} = \left[-\nabla_z \hat{U}(\boldsymbol{z}) - \hat{F}_{\text{inst}}(\boldsymbol{z})\right]^2, \tag{7}$$

where $\hat{F}_{\text{inst}}(\boldsymbol{z})$ is the instantaneous mean force, which we use as an estimate of the true mean force (see Appendix B for details on the mean force). Importantly, $\hat{F}_{\text{inst}}(\boldsymbol{z})$ is a function of $\mathcal{M}$, ensuring that $\mathcal{M}$ informs the training of $\hat{U}$.

We can then use $\hat{U}$ to inform the training $\mathcal{M}$ using the same objective in Eq. (7). In subsequent epochs, we train $\mathcal{M}$ using $\mathcal{L}_{\text{mf}}$ in addition to the reconstruction and auxiliary losses. Including $\mathcal{L}_{\text{mf}}$ penalizes contributions from atoms that contribute to large variances in $F_{\text{inst}}(\boldsymbol{z})$, the estimate of the mean force. Physically, these atoms correspond to fast-moving degrees of freedom. We emphasize that when training $\mathcal{M}$, $\hat{U}$ is held fixed, and vice versa.

Finally, from a coarse-grained configuration $\boldsymbol{z}$, we would like to generate configurations in the fine-grained space $\boldsymbol{x} \sim p_{\text{inv}}(\boldsymbol{x}|\boldsymbol{z})$, which we achieve using our decoder $\mathcal{M}^{\text{dec}}$ and a normalizing flow. To reduce the computational burden of the generative process, we work with internal coordinates: bond angles, bond lengths, and dihedral angles. Furthermore, because the distribution of bond angles and bond lengths are well approximated by independent Gaussian distributions, with narrow variances, we train a normalizing flow to only sample dihedral angles $\phi \in [-\pi, \pi]^m$ with the bond lengths and bond angles set as the median of their corresponding distributions. We represent our normalizing flow using rational-quadratic neural spline flows (RQ-NSF) with autoregressive layers, primarily because this architecture imposes bounded domains.[53] After training our normalizing flow using a forward loss (see Appendix C), we can sample dihedral angles $\phi \sim p_{\text{NF}}(\phi|\tilde{\boldsymbol{x}})$. With ϕ and $\tilde{\boldsymbol{x}}$, we can easily reconstruct our full fine-grained configuration, a configuration in the fine-grained space (see Fig. 2). Thus, our backmapping procedure amounts to sampling $\boldsymbol{x} \sim p_{inv}(\boldsymbol{x}|\boldsymbol{z})$.

VIII. DISCUSSION

Coarse-graining has traditionally relied on physical intuition to design effective models for complicated, nonlinear dynamical systems. With the commensurate loss of intuitive interpretability, we simultaneously exploit the unique capabilities of generative models to reconstruct the atomistic coordinates of the system. To interpret convergence, we introduce a conceptual framework for thermodynamic consistency that establishes a notion of equivalence based on averages of observables in the fine-grained space. Additionally, we introduce a highly modular computational pipeline that adheres to this framework. Finally, we demonstrate the capability of our method to quantitatively compute key observables for two proteins: a simple model system, alanine dipeptide, and a fast-folding protein with multiple metastable states, chignolin. This includes structural observables, such as RMSD (Fig. 8).

The computational procedure we introduce here is a synthesis of a variety of methods originating from the machine learning and coarse-graining communities. Although practically we were only

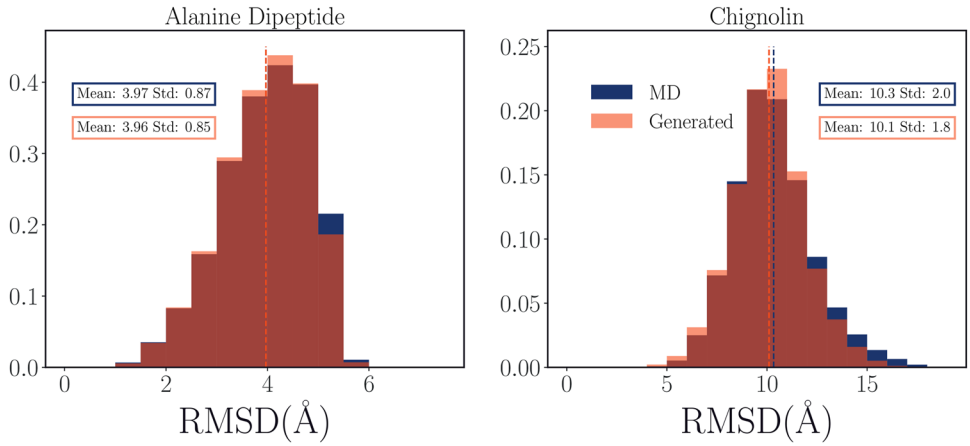

FIG. 8. Distribution of root-mean-square-deviation (RMSD) of generated configurations from configurations in MD dataset (orange) and RMSD of configurations in MD dataset from configurations in MD dataset (blue). Given n_{gen} generated configurations and n_{MD} configurations in the MD dataset, orange histogram consists of $n_{\text{MD}}n_{\text{gen}}$ RMSDs and blue histogram consists of $n_{\text{MD}}(n_{\text{MD}}-1)$ RMSDs (RMSD was not computed between same configuration).

able to test a small number of different neural network architectures, we believe there is substantial opportunity for improvement as the embedding, the representation of the coarse-grained potential, and the inversion map all become more sophisticated. Fortunately, the strategy that we have introduced is highly modular and could even be applied to classical force-matching, projective coarse-graining maps simply by augmenting them with the inversion map.

In the two systems we investigated here, we had access to a dataset for which the metastable states of interest were well sampled. More complex biomolecular systems that are less well characterized will require more sophisticated sampling strategies to generate sufficient data for a generalizable coarse-graining. We anticipate a feedback approach, where coarse-grained simulations and full molecular dynamic simulations are used in tandem to map out the free energy landscape of the system.

ACKNOWLEDGMENTS

The authors thank Sherry Li for helpful discussions on normalizing flows. This material was based upon work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Award No. DE-SC0022917.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Shriram Chennakesavalu: Conceptualization (equal); Software (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **David J. Toomer**: Conceptualization (equal); Software (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Grant M. Rotskoff**: Conceptualization (equal); Formal analysis (equal); Methodology (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request. Our code is available at https://github.com/rotskoff-group/thermodynamic-consistency.

APPENDIX A: NOTATION

Below we abbreviate fine-grained (FG) and coarse-grained (CG).

n	Number of atoms in the FG space
K	Number of "beads" in the CG space
ρ	FG probability density function
$\boldsymbol{X}$	Typical notation for a configuration in $\mathbb{R}^{3n}$
$Z(\beta)$	FG canonical partition function
$\rho(\boldsymbol{x})d\boldsymbol{x} = Z^{-1}e^{-\beta(\boldsymbol{x})}d\boldsymbol{x}$	Probability of FG configuration
$\mathcal{M}: \mathbb{R}^{3n} \to \mathbb{R}^{3k}$	$\boldsymbol{x}$ Notation for CG map
$\hat{F}(\boldsymbol{z})$	The potential of mean force associated with $\mathcal{M}$
$\hat{\cdot}$	Generic indication of a function defined on the CG domain
$\boldsymbol{\Theta}$	Parameters of the CG potential energy
$\hat{U}(\cdot, \boldsymbol{\theta})$	CG potential energy, a parametric function
$\hat{Z}(\beta, \boldsymbol{\theta})$	Partition function associated with the CG potential energy
$\hat{\rho}(\boldsymbol{z}, \boldsymbol{\theta})d\boldsymbol{z} = \hat{Z}^{-1}e^{-\beta\hat{U}(\boldsymbol{z},\boldsymbol{\theta})}d\boldsymbol{z}$	Probability of $\boldsymbol{z}$ with CG energy $\hat{U}$
T	a normalizing flow
$\rho: \mathbb{R}^{3n} \to \mathbb{R}$	The base density of the normalizing flow
$\tilde{n} < n$	Number of DOFs in the partially backmapped structure
$\tilde{\cdot}$	Generic indication of a function defined on $\mathbb{R}^{3\tilde{n}}$
$p_{\text{inv}}(\boldsymbol{x}\|\boldsymbol{z})$	Notation for the conditional distribution

APPENDIX B: COMPUTATIONAL DETAILS FOR COARSE-GRAINING

We use a training scheme that alternates between training a coarse-graining function Θ and the coarse-grained energy function $\hat{U}$. In this scheme, we learn an initial Θ_0, which is then used to learn an initial $\hat{U}_0$. This $\hat{U}_0$ is then used to partially inform the training of a new Θ_1, with this feedback approach continuing for a predefined number of epochs $\boldsymbol{n}_{\text{freeze\_cg}}$. After $\boldsymbol{n}_{\text{freeze\_cg}}$ training epochs, we fix the coarse-graining function Θ and train $\hat{U}$ until convergence.

We represent our configurations using a three-dimensional graph, where nodes correspond to atoms and edges between nodes correspond to bonded and nonbonded interactions. For a given atom, atoms within a prespecified nonbonded edge cutoff are connected via a nonbonded edge if a bonded edge does not already exist. We use atomic numbers as input node features and the type of the edge (bonded, nonbonded, self) as the input edge feature.

To learn a coarse-graining function Θ, we consider the encoding-decoding tasks of learning an embedding to determine a coarse-grained configuration $\boldsymbol{z}_i = \Theta(\boldsymbol{x}_i) \in \mathbb{R}^{3k}$, which can then be used to accurately reconstruct a set of target atoms $\tilde{\boldsymbol{x}}_i = \Theta^{\text{dec}}(\boldsymbol{z}_i) \in \mathbb{R}^{3\tilde{n}}$ from the original fine-grained configuration $x_i \in \mathbb{R}^{3n}$ (see Algorithm 1). The target atoms that are reconstructed using the decoder need to be specified *a priori*. In practice, this requires limited knowledge of the system. For the two model systems we consider here, we designate the backbone as the set of target atoms to reconstruct.

We model our encoder-decoder (Fig. 2) scheme after `DiffPool`, a hierarchical graph pooling approach.[13] Our encoder consists of a single pooling layer, where we learn a fine-grained-configuration-dependent projection matrix $P_{\boldsymbol{x}_i}$, such that $\boldsymbol{z}_i = \Theta(\boldsymbol{x}_i) = P_{\boldsymbol{x}_i}\boldsymbol{x}_i$. Similarly, the decoder consists of a single "inverse-pooling" layer, where we learn a coarse-grained-configuration-dependent projection matrix, $P_{\boldsymbol{z}_i}$ in order to reconstruct $\tilde{\boldsymbol{x}}_i = \Theta^{\text{dec}}(\boldsymbol{z}_i) = P_{\boldsymbol{z}_i}\boldsymbol{z}_i$. Importantly, the encoder-decoder is state-dependent, ensuring it is consistent with the coarse-graining framework we introduce in the main text.

Here, $P_{\boldsymbol{x}_i} = \text{softmax}(\text{GNN}(\boldsymbol{x}_i)) \in \mathbb{R}^{k\times n}$ and $P_{\boldsymbol{z}_i} = \text{GNN}(\boldsymbol{x}_i) \in \mathbb{R}^{\tilde{n}\times k}$, where the softmax is computed row-wise to ensure that the sum of all atomic contributions to each coarse-grained "bead" is 1. The `DiffPool` scheme allows for any general message-passing GNN architecture to be used; we use the E(n) – EGNN graph neural network architecture, which imposes necessary rotational and physical invariance constraints.[41]

ALGORITHM 1. Encoder-decoder training.

```
Initialize encoder Θ and decoder Θ^dec and corresponding
  optimizer and scheduler
for e = 0 ... M do
    if e < n_freeze_cg then
        for t = 0 ... n_cg do
            Compute L_Θ and carry out optimization step of Θ
        end for
        (Re)-initialize Û and corresponding optimizer and
        scheduler
    end if
    for t = 0 ... n_u do
        Compute L_Û and carry out optimization step of Û
    end for
end for
```

Our loss function to train this encoder-decoder consists of a reconstruction loss and a suite of auxiliary losses, which aid in regularization,

$$\mathcal{L}_\Theta = \mathcal{L}_{\text{r}} + \lambda[\mathcal{L}_{\text{link}} + \mathcal{L}_{\text{ent}} + \mathcal{L}_{\text{assgn}} + \lambda_{\text{mf}}\mathcal{L}_{\text{mf}}], \tag{B1}$$

where λ and λ_{mf} are hyperparameters controlling the weight of the four auxiliary losses. The reconstruction loss $\mathcal{L}_{\text{r}} = \|P_{\boldsymbol{z}}P_{\boldsymbol{x}}\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_2^2$, where $P_{\boldsymbol{x}}$ and $P_{\boldsymbol{z}}$ are dependent on the initial fine-grained configuration x and the coarse-grained configuration $\boldsymbol{z} = P_{\boldsymbol{x}}\boldsymbol{x}$ respectively. In our implementation, we remove translational shifts before computing the distance. Additionally, when computing $\mathcal{L}_{\text{r}}$, we weigh all backbone carbon and nitrogen atoms by $\alpha_{\text{bb}} > 1$ to ensure that these atoms get reconstructed with a higher fidelity.

The link loss $\mathcal{L}_{\text{link}} = \left\|D \odot P_{\boldsymbol{x}}^T P_{\boldsymbol{x}}\right\|_F$ ensures that proximal atoms are projected onto the same coarse-grained bead; here $\|\cdot\|_F$ denotes the Frobenius norm and $D \in \mathbb{R}^{n\times n}$ is the matrix of all pairwise distances between atoms of the fine-grained configuration x. $(P_x^T P_x)_{ij}$ gives a measure of the overlap of the bead assignments between atom i and atom j. We want atoms that have overlap in bead assigments to correspond to atoms that are proximal and therefore have a lower interatomic distance. The Frobenius norm of the Hadamard product between $P_x^T P_x$ and D penalizes assigning two atoms to the same bead that have a large interatomic distance.

The entropy loss $\mathcal{L}_{\text{ent}} = \frac{1}{k}\sum_{i=1}^{k} H(P_{\boldsymbol{x}}^i)$, where $P_{\boldsymbol{x}}^i$ the i-th row of the projection matrix $P_{\boldsymbol{x}}$ and H denotes the entropy function. This loss ensures that the fractional weights of the atoms assigned to each bead are concentrated around a few atoms.

The assignment loss $\mathcal{L}_{\text{assgn}} = \text{diag}\,(P_{\boldsymbol{x}}^T P_{\boldsymbol{x}})$ ensures that an atom is not assigned to multiple beads, where diag corresponds to the sum of the diagonal entries of the matrix. If an atom is assigned to multiple beads, $P_x^T P_x$ will have diagonal entries greater than 1. The $\mathcal{L}_{\text{assgn}}$ loss, therefore, penalizes the assignment of an atom to multiple beads.

Finally, we elaborate on the mean-force loss $\mathcal{L}_{\text{mf}}$ below after a brief discussion on the computation of the mean-force.

For the systems we explore here, we cannot easily compute a coarse-grained potential energy function $\hat{U}$; instead, we learn $\hat{U}$ using force-matching.[5,6] We match $\nabla_{\boldsymbol{z}}\hat{U}$ to the mean force of a coarse-grained configuration. The mean force formally is

$$F(z) = \langle \Theta_F(\nabla_{\boldsymbol{x}}U(\boldsymbol{x}))\rangle_{\Theta(\boldsymbol{x})=\boldsymbol{z}}, \tag{B2}$$

where Θ_F projects the fine-grained force $\nabla_{\boldsymbol{x}}U(\boldsymbol{x})$ into the coarse-grained space.[2] Computing this average exactly can be costly; instead, we compute an estimate of the mean force using the instantaneous coarse-grained force,[61,62]

$$F_{\text{inst}}(\boldsymbol{z}) = (P_{\boldsymbol{x}}P_{\boldsymbol{x}}^T)^{-1}P_{\boldsymbol{x}}\nabla_{\boldsymbol{x}}U(\boldsymbol{x}), \tag{B3}$$

where $\boldsymbol{z} = \Theta(\boldsymbol{x}) = P_{\boldsymbol{x}}\boldsymbol{x}$. With this estimate, we train $\hat{U}$ by minimizing the following loss:

$$\mathcal{L}_{\hat{U}} = \left[-\nabla_{\boldsymbol{z}}\hat{U}(\boldsymbol{z}) - F_{\text{inst}}(\boldsymbol{z})\right]^2. \tag{B4}$$

The light blue term is the one in which gradients with respect to the parameters are being tracked. We use the SchNet architecture to represent $\hat{U}$.[42] Briefly, SchNet transforms interbead distances in to Gaussian basis functions with learnable parameters; these Gaussian basis functions are then passed through multiple neural network layers. We assign each bead a unique input feature $(0 \ldots k)$, analogous to an atomic number.

Finally, we use our coarse-grained potential energy $\hat{U}$ to inform the coarse-grained map Θ that we learn using $\mathcal{L}_{\text{mf}}$,

$$\mathcal{L}_{\text{mf}} = \left[-\nabla_z \hat{U}(\boldsymbol{z}) - F_{\text{inst}}(\boldsymbol{z})\right]^2. \tag{B5}$$

The light blue term is the one in which gradients with respect to the parameters are being tracked. While, $\mathcal{L}_{\text{mf}}$ has the same form as $\mathcal{L}_{\hat{U}}$, in Eq. (B4) we fix our coarse-graining map and allow $\hat{U}$ to train. In Eq. (B5) on the other hand, we fix the coarse-grained potential energy function $\hat{U}$ and instead allow our coarse-graining map to vary. Using $\hat{U}$ to inform Θ via $\mathcal{L}_{\text{mf}}$ reduces contributions from atoms that increase the variance of the mean force estimate $F_{\text{inst}}(\boldsymbol{z})$, an observation also made using a similar regularization term in Ref. 7. Empirically, we observe that this results in a lower entropy projection map. For the experiments included here, we are able to learn a low-entropy map during the first epoch of training Θ, so the inclusion of $\mathcal{L}_{\text{mf}}$ has limited practical utility for the systems considered here.

APPENDIX C: COMPUTATIONAL DETAILS FOR BACKMAPPING

In the coarse-graining scheme discussed in Appendix B, we train an encoder-decoder that deterministically embeds a fine-grained configuration x_i into a coarse-grained configuration $\boldsymbol{z}_i$; the coarse-grained configuration is then used to deterministically reconstruct $\tilde{\boldsymbol{x}}_i$, which corresponds to a set of target atoms from the original configuration. Of course, we cannot compute observables dependent on the full fine-grained configuration using this approach. Here, we describe the backmapping procedure used to generate fine-grained configurations from a coarse-grained configuration $p(\boldsymbol{x}|\boldsymbol{z}_i)$.

To simplify the backmapping process, we work with internal coordinates, namely bond lengths, bond angles and dihedral angles. From a set of internal coordinates, the Cartesian coordinates of each atom in the system can be easily computed. We use a normalizing flow to generate internal coordinates conditioned on a coarse-grained configuration. Normalizing flows are a class of invertible neural networks that enable a transformation between two distributions. Importantly, normalizing flows enable efficient and exact density estimation.[24]

For the systems we consider here, the distributions of bond angles and distances are unimodal Gaussian distributions with narrow variances. On the other hand, the distributions of dihedral angles are generally multimodal with large variances. To simplify the generative process, we set all bond angles and distances to be the median of their respective distributions. Thus, we only consider the task of generating dihedral angles $\phi_i = [\phi_i^0, \phi_i^1 \ldots \phi_i^m] \in [-\pi, \pi]^m$.

To carry out the backmapping process, we first use the decoder Θ^{dec} to transform $\boldsymbol{z}_i$ into $\tilde{\boldsymbol{x}}_i$. We then compute $\phi^{\text{seed}} \in [-\pi, \pi]^{\tilde{m}}$, the dihedral angles of $\tilde{\boldsymbol{x}}_i$. Finally, we use our trained normalizing flow to sample a set of dihedral angles $\phi_i \sim p(\phi|\phi^{\text{seed}})$. Using $\tilde{\boldsymbol{x}}_i$ and ϕ_i, we can easily generate $\boldsymbol{x}_i$, a configuration in the fine-grained space. Thus, our backmapping procedure amounts to sampling $\boldsymbol{x}_i \sim p(\boldsymbol{x}|\boldsymbol{z}_i)$.

In order to train our normalizing flow, we seek to learn a map T_* that transports the base distribution ρ, a multivariate Gaussian distribution with mean zero and identity covariance, to the target distribution $\tilde{\rho}$, the distribution of internal coordinates that correspond to Boltzmann-distributed configurations. Denoting the map represented by normalizing flow T,

$$T\sharp\rho(\phi) = \rho(\tilde{T}(\phi))\left|\nabla_\phi \tilde{T}\right|, \tag{C1}$$

where $\tilde{T}$ denotes the inverse map. In order to learn a map T that approximates T_*, we seek to minimize the forward Kullback-Leibler Divergence, $D_{\text{KL}}(\tilde{\rho}|T\sharp\rho)$. The KL divergence, up to a constant, can be estimated using the following loss function,

$$\mathcal{L}_{\text{NF}} = -\frac{1}{N}\sum_{i=1}^{N} \log\, T\sharp\rho(\phi_i), \tag{C2}$$

where the exact density $T\sharp\rho$ can be computed using Eq. (C1) and N is the number of data points.

We use the rational-quadratic neural spline flow (RQ-NSF) architecture to represent our normalizing flow.[53] We make this choice because the domain of dihedral angles is bounded to $[-\pi, \pi]$ and the RQ-NSF approach considers transformations between bounded domains. Finally, within this architecture we use autoregressive layers to carry out the actual normalizing flow. Practically, this amounts to using the first j dihedral angles, $\phi^{0:j-1}$, to inform the generative process of ϕ^j.

With RQ-NSF, we define a set of m neural networks $\text{FCN}^0, \text{FCN}^1, \ldots, \text{FCN}^m$ that are used in the transformation of $\phi^0, \phi^1, \ldots, \phi^m$, respectively. Each of these neural networks are fully connected with a single hidden layer. Given a sample from the base distribution $\phi_{b_i} \sim \rho_b$, we can compute $\theta_i^j = \text{FCN}^j(\phi_i^{0:j-1}, \phi^{\text{seed}})$. Here, θ_i^j correspond to the parameters of a rational quadratic spline g. Finally, we can compute $\phi_i^j = g_{\theta_i^j}(\phi_{b_i}^j)$. See Algorithm 2 for a summary of the backmapping process. For a thorough description of RQ-NSF, see Ref. 53.

There is a natural hierarchy to the backmapping procedure here, where atoms that are a single bond away from the atoms in $\tilde{x}$ are reconstructed first with atoms further away being reconstructed later. When carrying out the autoregressive flow, we remain faithful to this hierarchy, ensuring that atoms reconstructed first influence the internal coordinates of atoms reconstructed later.

ALGORITHM 2. Sampling $\boldsymbol{x}_i$ from $p(\boldsymbol{x}|\boldsymbol{z}_i)$.

1: Train $\text{FCN}^0, \text{FCN}^1, \ldots\ \text{FCN}^m$
2: Compute ϕ^{seed} from $\tilde{\boldsymbol{x}}_i = \Theta^{\text{dec}}(\boldsymbol{z}_i)$
3: Sample $\phi_{b_i} \sim \rho$
4: **for** $j = 0 \ \ldots \ m$ **do**
5: Compute $\theta_i^j = \text{FCN}(\phi_i^{0:j-1}, \phi^{\text{seed}})$
6: Compute $\phi_i^j = g_{\theta_i^j}(\phi_{b_i}^j)$
7: **end for**
8: Reconstruct $\boldsymbol{x}_i$ from $\tilde{\boldsymbol{x}}_i$ and ϕ_i

APPENDIX D: NUMERICAL EXPERIMENTS

As described in the main text, we carry out our coarse-graining and backmapping procedures for two proteins: Alanine Dipeptide and Chignolin. Here, we briefly expand on some of the experimental details used when investigating these systems, including a list of hyperparameters used (see Tables I and II).

We selected the dimensionality of the coarse-grained space (i.e., the number of coarse-grained) beads to balance the trade-off between accelerating sampling and maintaining overlap of our generated distribution to our target distribution. We empirically determined the number of beads that could balance the trade-off for both chignolin and alanine dipeptide (see Table I). For future systems, we recommend an initial strategy that prioritizes rapid exploration. An initial coarse-graining should be one that reduces the dimensionality significantly; if the reconstructed fine-grained configurations have poor overlap with the target Boltzmann distribution, then we recommend increasing the coarse-grained dimensionality.

During training, we use the ReduceLROnPlateau scheduler to train Θ and $\hat{U}$. The scheduler is assigned a metric of interest and it anneals the learning rate as this metric converges. For Θ, we use the reconstruction loss $\mathcal{L}_r$ on the validation set as the metric for the scheduler. For $\hat{U}$, we use the mean absolute error (MAE) between $\nabla\hat{U}(z)$ and $\boldsymbol{F_{inst}(z)}$ on the validation set as the metric for the scheduler. For both Chignolin and Alanine Dipeptide, we use a 80/20 train-validation split.

TABLE II. Relevant hyperparameters for normalizing flow training.

Hyperparameters		
Hyperparameter	Chignolin	Alanine dipeptide
RQ-NSF layers	2	2
Width	256	128
Optimizer	Adam	Adam
Learning rate	3×10^{-4}	3×10^{-4}
Number of datapoints	50 000	50 000
Batch size	128	64

TABLE I. Relevant hyperparameters for encoder-decoder ($\Theta - \Theta^{dec}$) and $\hat{U}$ training.

Hyperparameters		
Hyperparameter	Chignolin	Alanine dipeptide
k (No. of beads)	30	6
$\tilde{n}$ (No. of target atoms)	40	13
α_{bb}	5.0	2.0
$\hat{U}$ optimizer	Adam	Adam
$\hat{U}$ learning rate (LR)	3×10^{-4}	3×10^{-4}
$\hat{U}$ scheduler	ReduceLROnPlateau	ReduceLROnPlateau
$\hat{U}$ patience	5	5
$\hat{U}$ factor	0.8	0.8
$\hat{U}$ minimum LR	1×10^{-6}	1×10^{-6}
$\hat{U}$ # of SchNet layers	2	2
$\hat{U}$ width	128	128
$\hat{U}$ cutoff (Å)	15	10
$\hat{U}$ $\boldsymbol{n}_{gaussians}$	25	25
Θ optimizer	Adam	Adam
Θ learning rate (LR)	1×10^{-4}	1×10^{-4}
Θ scheduler	ReduceLROnPlateau	ReduceLROnPlateau
Θ patience	5	5
Θ factor	0.8	0.8
Θ minimum LR	1×10^{-6}	1×10^{-6}
Θ # of E(n) layers	2	2
Θ width	128	2
Θ nonbonded-edge cutoff (Å)	5	2
Λ	5.0	0.2
λ_{mf}	0.001	0.001
$\boldsymbol{n}_{freeze\_cg}$	2	5
$\boldsymbol{n}_{cg}$	15	10
$\boldsymbol{n}_{u}$	10	10
Batch size	8	8
Number of datapoints	50 000	50 000

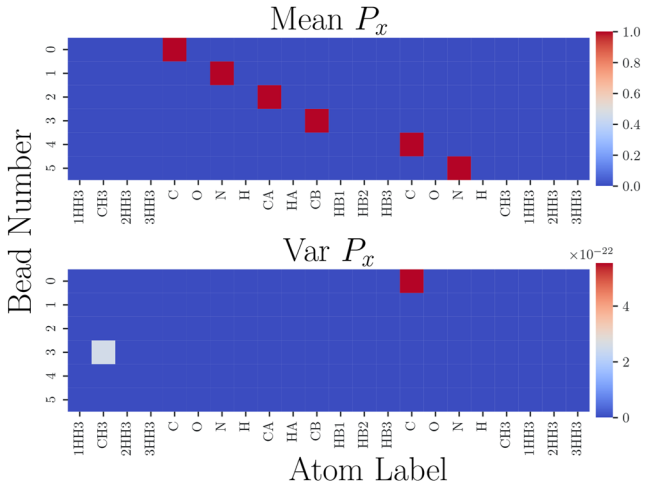

FIG. 9. Alanine dipeptide. Projection matrices $P_{\boldsymbol{x}} = \Theta(\boldsymbol{x})$ computed for all fine-grained configurations $\boldsymbol{x}$ in dataset. Mean $P_{\boldsymbol{x}}$ (top) and variance $P_{\boldsymbol{x}}$ (bottom). Coarse-graining maps are essentially the same across all fine-grained configurations in dataset and heavily weigh backbone atoms. Bead atoms are reindexed for ease of visualization such that lower bead indices correspond to C-terminus side and higher bead indices correspond to N-terminus.

We plot the mean and variance of all the projection matrices computed for each configuration in our dataset for alanine dipeptide (Fig. 9) and chignolin (Fig. 10). For each of the systems we investigate here, the projection matrices learned are essentially the same across for all the configurations in dataset. Importantly, the learned coarse-graining map corresponds to a physically meaningful map, where backbone atoms $(\mathrm{C}, \mathrm{C}_\alpha, \mathrm{N})$ are given a high weight, with side chain atoms given a zero weight. The conformational dynamics of the two proteins we investigate are dominated by backbone behavior and the coarse-graining maps we use respect this behavior. For more complex biomolecular systems and/or for a coarse-graining aimed at a greater dimensionality reduction, the coarse-graining map will be less physically obvious. We anticipate the approach we introduce here will remain robust to these potential challenges.

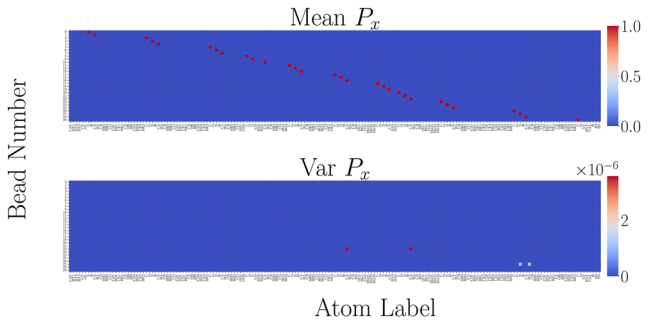

FIG. 10. Chignolin. Projection matrices $P_{\boldsymbol{x}} = \Theta(\boldsymbol{x})$ computed for all fine-grained configuration $\boldsymbol{x}$ in dataset. Mean $P_{\boldsymbol{x}}$ (top) and variance $P_{\boldsymbol{x}}$ (bottom). Coarse-graining maps are essentially the same across all fine-grained configurations in dataset and heavily weigh backbone atoms. Bead atoms are reindexed for ease of visualization such that lower bead indices correspond to C-terminus side and higher bead indices correspond to N-terminus.

APPENDIX E: COARSE-GRAINED SIMULATIONS

Given a trained coarse-grained potential function $\hat{U}$, we can sample the coarse-grained space via standard Langevin dynamics. We carry out dynamics using an "OVRVO" integration scheme.[63] We use the `OpenMM`[64] simulation platform to carry out all coarse-grained simulations. Finally, we use the `TorchForce` plugin to interface our coarse-grained potential energy function $\hat{U}$ into the simulation.

Given a vector $\boldsymbol{m} = [m^1, m^2, \ldots, m^n]$ consisting of all the masses of the atoms in the fine-grained configuration, we define the mass of a bead to be $\boldsymbol{m}\langle P_{\boldsymbol{x}}\rangle_{\mathcal{D}}^T$, where $\langle P_{\boldsymbol{x}}\rangle_{\mathcal{D}}$ is the average projection matrix across all configurations in our dataset $\mathcal{D}$. For the coarse-graining maps we learn, our projection matrices are essentially the same across all fine-grained configuration in $\mathcal{D}$, so this approach is a sensible one. For systems with more complex coarse-graining maps, a different strategy to account for masses will likely be necessary.

For alanine dipeptide, we ran 15 coarse-grained trajectories each for a total of 2 000 000 steps. We used a time step of dt = 0.0001 ps and friction coefficient $\gamma = 100$ ps^{-1}. These trajectories were initialized at positions across the free-energy landscape to ensure sufficient coverage. We subsample a total of 20 000 coarse-grained configurations and for each configurations, we generate 2000 fine-grained configurations for a total of 4×10^7 fine-grained configurations. Finally, we carry out a reweighting step followed by a relaxation step detailed below.

For chignolin, we initially ran 24 trajectories each for a total of 2 000 000 steps. For these trajectories, we used a time step of dt = 0.001 ps and friction coefficient $\gamma = 100$ ps^{-1}. We started 8 of these trajectories in the first basin (unfolded state), 8 of these trajectories in the second basin (folded state), and 8 of these trajectories in the third basin (misfolded state). When carrying out the backmapping process, we observed that trajectories that started in the misfolded state quickly transitioned to one of the other 2 basins, resulting in limited sampling of the misfolded state. To alleviate this, we ran three more trajectories starting in this basin with a dt = 0.000 01 ps and friction coefficient $\gamma = 100$ ps^{-1}, also of length 2 000 000 steps. From these 27 trajectories, we backmap 54 000 coarse-grained configurations and generate 750 fine-grained configuration per coarse-grained configuration. Finally, we carry out a relaxation step followed by a reweighting step detailed below.

APPENDIX F: REWEIGHTING PROCEDURE

We sample from the Boltzmann distribution via a sampling scheme that combines coarse-grained dynamics with a backmapping procedure that utilizes a normalizing flow. There is no theoretical guarantee that this sampling scheme allows us to sample from the true Boltzmann distribution; instead, we must carry out a final reweighting to ensure that samples are appropriately weighted. Below, we highlight two different reweighting procedures, one that explicitly uses $p(\boldsymbol{z})$ and $p(\boldsymbol{x}|\boldsymbol{z})$ the other that implicitly accounts for it. To limit the computational burden, all energies are computed via

an implicit model using OpenMM. In addition, we carry out a short relaxation procedure of overdamped Brownian dynamics—again using an implicit model—in order to relax any minor structural deformities and ensure configurations are reflective of ambient temperature.

1. Alanine dipeptide

Given a fine-grained configuration x_j backmapped from a coarse-grained configuration $\boldsymbol{z}_i$ [i.e., $\boldsymbol{x}_j \sim p(\boldsymbol{x}|\boldsymbol{z}_i)$], we weigh each configuration according to $w_{ij}/\sum_{ij}(w_{ij})$, where $w_{ij} = \frac{\exp(-\beta U_{\text{implicit}}(\boldsymbol{x}_j))}{\exp(-\beta \hat{U}(\boldsymbol{z}_i)) * p(\boldsymbol{x}_j|\boldsymbol{z}_i)}$. The sum is taken over all fine-grained configurations $\boldsymbol{x}_j$ backmapped from all coarse-grained configurations $\boldsymbol{z}_i$. Finally, we carry out a single relaxation step of overdamped Brownian dynamics with a time-step of 0.000 02 ps and friction coefficient $\gamma = 10$ ps^{-1}. The time-step we use for this relaxation is 200× lower than the time-step used to carry out the original MD simulation.

2. Chignolin

For chignolin, the fine-grained structures we backmap are generally high-energy configurations; however, this was generally a result of minor structural deformities as opposed to major flaws in the reconstruction procedure. Because of this high-energy, we cannot carry out the same reweighting procedure we used with alanine dipeptide. However, we do have access to the distribution of potential energies—recomputed using an implicit solvent model—of the configurations in our MD-dataset, which we reweight with respect to. First, we carry out 300 steps of overdamped Brownian dynamics with a time-step of 0.00002 ps and friction coefficient $\gamma = 10$ ps^{-1} in order to relax the structure to a reasonable energy.

Then, we reweight the distribution of potential energies from our relaxed configurations with respect to distribution of potential energies from our MD dataset. Given a histogram binning function $h : \mathbb{R} \to [0, 1]$ that computes the probability of the bin a potential energy belongs to, we assign to each fine-grained configuration $\boldsymbol{x}_j$ backmapped from a coarse-grained configuration a weight $w_{ij}/\sum_{ij}(w_{ij})$, where $w_{ij} = \frac{h^{\text{MD}}(U_{\text{implicit}}(\boldsymbol{x}_j))}{h^{\text{gen}}(U_{\text{implicit}}(\boldsymbol{x}_j))}$.

APPENDIX G: DATASETS

We carried out a simulation of alanine dipeptide in explicit solvent at a temperature of 300 K with the AMBER ff99SB force field and the TIP3P water model. We used a time-step of 0.004 ps and a friction coefficient of $\gamma = 0.1$ ps^{-1}. The total simulation time was over 0.5 μs. Finally, we saved the configurations every 1 ps for a dataset of over 500 000 points. From this dataset, we subsampled 50 000 data points consisting of positions and forces, which were then used for training.

We used the dataset of chignolin trajectories from Ref. 9. See supplement in Ref. 9 for simulation details. From this dataset, we similarly subsampled 50 000 data points consisting of positions and forces to carry out our training.

APPENDIX H: WEAK THERMODYNAMIC CONSISTENCY

In this appendix, we state and prove the proposition stated in the main text. Throughout, simply for notational clarity, we assume that U and $\hat{U}$ are shifted so that the normalization constants are unity. In this section, the parameters of $\hat{U}$ are assumed to be fixed at an optimal value, so we suppress them for notational clarity. In what follows: the map $\mathcal{M}$ can be viewed as a map from $\mathbb{R}^{3n} \to \mathbb{R}^{3\tilde{n}}$, which is the dimensionality of the space after use of the decoder.

Definition H.1. An invertible coarse-graining is a tuple $(\mathcal{M}, \hat{U}, T)$ *consisting of coarse-graining map* $\mathcal{M} : \mathbb{R}^{3n} \to \mathbb{R}^{3k}$, *a coarse-grained potential* $\hat{U} : \mathbb{R}^{3k} \to \mathbb{R}$, *and a normalizing flow* $T : \mathbb{R}^{3n} \to \mathbb{R}^{3n}$.

Definition H.2. A projective coarse-graining is a pair $(\mathcal{M}, \hat{U})$ *consisting of a coarse-graining map* $\mathcal{M} : \mathbb{R}^{3n} \to \mathbb{R}^{3k}$ *and a coarse-grained potential* $\hat{U} : \mathbb{R}^{3k} \to \mathbb{R}$.

Definition H.3. We call a projective coarse-graining $(\mathcal{M}, \hat{U})$ *thermodynamically consistent if*

$$\hat{U}(\boldsymbol{z}) = -\beta^{-1} \log \int \delta(\mathcal{M}(\boldsymbol{x}) - \boldsymbol{z}) e^{-\beta U(\boldsymbol{x})} d\boldsymbol{x}. \tag{H1}$$

Proposition H.4. Let $(\mathcal{M}, \hat{U}, T)$ *be an invertible coarse-graining. Let* $\mathcal{F}_*$ *denote the set of functions of continuous, bounded functions,*

$$\mathcal{F}_* := \left\{ f \in \mathcal{C}(\mathbb{R}^{3n}, \mathbb{R}) \middle| f = \hat{f} \circ \mathcal{M}, \hat{f} \in \mathcal{C}(\mathbb{R}^{3k}, \mathbb{R}) \right\},$$

If $(\mathcal{M}, \hat{U}, T)$ *is* $\mathcal{F}_*$ *thermodynamically consistent, then the projective coarse-graining* $(\mathcal{M}, \hat{U})$ *is thermodynamically consistent in the sense of Ref. 2.*

We define the set $\mathcal{F}_*$ so that it includes only functions that can be written as a composition of a function $\hat{f}$ on the coarse-grained configuration with the coarse-graining map. A bigger set of functions, for example all bounded continuous functions, would yield the same result, but includes many observables on the fine-grained space that project trivially. The proof of this proposition is straightforward; we simply use $\mathcal{F}_*$ weak thermodynamic consistency in the fine-grained space and project. The assumption we make is that the degrees of freedom orthogonal to $\mathcal{M}(x)$ do not affect the value of the output of functions in $\mathcal{F}_*$. First, let $f \in \mathcal{F}_*$. By assumption,

$$\langle f \rangle = \int f(\boldsymbol{x}) e^{-\beta U(\boldsymbol{x})} d\boldsymbol{x} = \int f(\boldsymbol{x}) p_{\text{inv}}(\boldsymbol{x}|\boldsymbol{z}) e^{-\beta \hat{U}(\boldsymbol{z})} d\boldsymbol{z} d\boldsymbol{x}, \tag{H2}$$

so, in particular, for any bounded, continuous $\hat{f} : \mathbb{R}^{3k} \to \mathbb{R}$,

$$\begin{aligned} \int \hat{f}(\mathcal{M}(\boldsymbol{x})) e^{-\beta U(\boldsymbol{x})} d\boldsymbol{x} &= \int \hat{f}(\mathcal{M}(\boldsymbol{x})) p_{\text{inv}}(\boldsymbol{x}|\boldsymbol{z}) e^{-\beta \hat{U}(\boldsymbol{z})} d\boldsymbol{z} d\boldsymbol{x} \\ &= \int \hat{f}(\boldsymbol{z}) e^{-\beta \hat{U}(\boldsymbol{z})} d\boldsymbol{z}, \end{aligned} \tag{H3}$$

which follows from the normalization of p_{inv}. The equality obtained here is the weak form of Definition H.3. To obtain pointwise convergence, we construct a sequence of Gaussian distributions centered at each $\boldsymbol{z}$ that approach a δ-function. Note that we do not

prove that $\mathcal{F}_*$ is the minimal set of functions yielding thermodynamic consistency for $(\mathcal{M}, \hat{U})$, though we believe that this is the case.

REFERENCES

[1] A. J. Pak and G. A. Voth, "Advances in coarse-grained modeling of macromolecular complexes," Curr. Opin. Struct. Biol. **52**, 119–126 (2018).
[2] W. G. Noid, J.-W. Chu, G. S. Ayton, V. Krishna, S. Izvekov, G. A. Voth, A. Das, and H. C. Andersen, "The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models," J. Chem. Phys. **128**, 244114 (2008).
[3] C. Clementi, "Coarse-grained models of protein folding: Toy models or predictive tools?," Curr. Opin. Struct. Biol. **18**, 10–15 (2008).
[4] M. G. Saunders and G. A. Voth, "Coarse-graining of multiprotein assemblies," Curr. Opin. Struct. Biol. **22**, 144–150 (2012).
[5] S. Izvekov and G. A. Voth, "A multiscale coarse-graining method for biomolecular systems," J. Phys. Chem. B **109**, 2469–2473 (2005).
[6] S. Izvekov and G. A. Voth, "Multiscale coarse graining of liquid-state systems," J. Chem. Phys. **123**, 134105 (2005).
[7] W. Wang and R. Gómez-Bombarelli, "Coarse-graining auto-encoders for molecular dynamics," npj Comput. Mater. **5**, 125 (2019).
[8] J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N. E. Charron, G. de Fabritiis, F. Noé, and C. Clementi, "Machine learning of coarse-grained molecular dynamics force fields," ACS Cent. Sci. **5**, 755–767 (2019).
[9] B. E. Husic, N. E. Charron, D. Lemm, J. Wang, A. Pérez, M. Majewski, A. Krämer, Y. Chen, S. Olsson, G. de Fabritiis, F. Noé, and C. Clementi, "Coarse graining molecular dynamics with graph neural networks," J. Chem. Phys. **153**, 194101 (2020).
[10] P. Das, M. Moll, H. Stamati, L. E. Kavraki, and C. Clementi, "Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction," Proc. Natl. Acad. Sci. U. S. A. **103**, 9885–9890 (2006).
[11] K. M. Lebold and W. G. Noid, "Dual-potential approach for coarse-grained implicit solvent models with accurate, internally consistent energetics and predictive transferability," J. Chem. Phys. **151**, 164113 (2019).
[12] J. Walther, P. D. Dans, A. Balaceanu, A. Hospital, G. Bayarri, and M. Orozco, "A multi-modal coarse grained model of DNA flexibility mappable to the atomistic level," Nucleic Acids Res. **48**, e29 (2020).
[13] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Advances in Neural Information Processing Systems*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates, Inc., 2018), Vol. 31.
[14] W. Wang, M. Xu, C. Cai, B. K. Miller, T. Smidt, Y. Wang, J. Tang, and R. Gomez-Bombarelli, "Generative coarse-graining of molecular conformations," in *Proceedings of the 39th International Conference on Machine Learning, Proceedings of Machine Learning Research*, edited by K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato (PMLR, 2022), Vol. 162, pp. 23213–23236.
[15] J. Köhler, A. Krämer, and F. Noe, "Smooth normalizing flows," in *Advances in Neural Information Processing Systems*, edited by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (Curran Associates, Inc., 2021), Vol. 34, pp. 2796–2809.
[16] F. Noé, S. Olsson, J. Köhler, and H. Wu, "Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning," Science **365**, eaaw1147 (2019).
[17] Y. Wang, L. Herron, and P. Tiwary, "From data to noise to data for mixing physics across temperatures with generative artificial intelligence," Proc. Natl. Acad. Sci. U. S. A. **119**, e2203656119 (2022).
[18] J. Köhler, Y. Chen, A. Krämer, C. Clementi, and F. Noé, "Flow-matching—Efficient coarse-graining molecular dynamics without forces," J. Chem. Theory Comput. **19**, 942–952 (2023).
[19] A. H. Mahmoud, M. Masters, S. J. Lee, and M. A. Lill, "Accurate sampling of macromolecular conformations using adaptive deep learning and coarse-grained representation," J. Chem. Inf. Model. **62**, 1602–1617 (2022).
[20] E. G. Tabak and E. Vanden-Eijnden, "Density estimation by dual ascent of the log-likelihood," Commun. Math. Sci. **8**, 217–233 (2010).
[21] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International Conference on Machine Learning* (PMLR, 2015), pp. 1530–1538.
[22] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," J. Mach. Learn. Res. **22**, 2617 (2021).
[23] M. S. Albergo, G. Kanwar, and P. E. Shanahan, "Flow-based generative models for Markov chain Monte Carlo in lattice field theory," Phys. Rev. D **100**, 034515 (2019).
[24] M. Gabrié, G. M. Rotskoff, and E. Vanden-Eijnden, "Adaptive Monte Carlo augmented with normalizing flows," Proc. Natl. Acad. Sci. U. S. A. **119**, e2109420119 (2022).
[25] M. Gabrié, G. M. Rotskoff, and E. Vanden-Eijnden, "Efficient bayesian sampling using normalizing flows to assist Markov chain Monte Carlo methods," in ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models, 2021.
[26] A. Arkhipov, P. L. Freddolino, K. Imada, K. Namba, and K. Schulten, "Coarse-grained molecular dynamics simulations of a rotating bacterial flagellum," Biophys. J. **91**, 4589–4597 (2006).
[27] M. Chakraborty, C. Xu, and A. D. White, "Encoding and selecting coarse-grain mapping operators with hierarchical graphs," J. Chem. Phys. **149**, 134106 (2018).
[28] M. A. Webb, J.-Y. Delannoy, and J. J. de Pablo, "Graph-based approach to systematic molecular coarse-graining," J. Chem. Theory Comput. **15**, 1199–1208 (2019).
[29] M. Giulini, M. Rigoli, G. Mattiotti, R. Menichetti, T. Tarenzi, R. Fiorentini, and R. Potestio, "From system modeling to system analysis: The impact of resolution level and resolution distribution in the computer-aided investigation of biomolecules," Front. Mol. Biosci. **8**, 676976 (2021).
[30] Z. Zhang, L. Lu, W. G. Noid, V. Krishna, J. Pfaendtner, and G. A. Voth, "A systematic methodology for defining coarse-grained sites in large biomolecules," Biophys. J. **95**, 5073–5083 (2008).
[31] M. Giulini, R. Menichetti, M. S. Shell, and R. Potestio, "An information theory-based approach for optimal model reduction of biomolecules," J. Chem. Theory Comput. **16**, 6795–6813 (2020).
[32] T. T. Foley, K. M. Kidder, M. S. Shell, and W. G. Noid, "Exploring the landscape of model representations," Proc. Natl. Acad. Sci. U. S. A. **117**, 24061–24068 (2020).
[33] M. Stieffenhofer, M. Wand, and T. Bereau, "Adversarial reverse mapping of equilibrated condensed-phase molecular structures," Mach. Learn.: Sci. Technol. **1**, 045014 (2020).
[34] M. Schöberl, N. Zabaras, and P.-S. Koutsourelakis, "Predictive collective variable discovery with deep Bayesian models," J. Chem. Phys. **150**, 024109 (2019).
[35] E. Pretti and M. S. Shell, "A microcanonical approach to temperature-transferable coarse-grained models using the relative entropy," J. Chem. Phys. **155**, 094102 (2021).
[36] G. Sivaraman and N. E. Jackson, "Coarse-grained density functional theory predictions via deep kernel learning," J. Chem. Theory Comput. **18**, 1129–1141 (2022).
[37] J. C. Maier and N. E. Jackson, "Bypassing backmapping: Coarse-grained electronic property distributions using heteroscedastic Gaussian processes," J. Chem. Phys. **157**, 174102 (2022).
[38] A. E. P. Durumeric and G. A. Voth, "Adversarial-residual-coarse-graining: Applying machine learning theory to systematic molecular coarse-graining," J. Chem. Phys. **151**, 124110 (2019).
[39] J. W. Wagner, J. F. Dama, A. E. P. Durumeric, and G. A. Voth, "On the representability problem and the physical meaning of coarse-grained models," J. Chem. Phys. **145**, 044108 (2016).
[40] J. Jin, A. J. Pak, A. E. P. Durumeric, T. D. Loose, and G. A. Voth, "Bottom-up coarse-graining: Principles and perspectives," J. Chem. Theory Comput. **18**, 5759–5791 (2022).
[41] V. G. Satorras, E. Hoogeboom, and M. Welling, "E(n) equivariant graph neural networks," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, Proceedings of Machine Learning Research*, edited by M. Meila and T. Zhang (PMLR, 2021), Vol. 139, pp. 9323–9332.

[42]K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, "SchNet—A deep learning architecture for molecules and materials," J. Chem. Phys. **148**, 241722 (2018).
[43]F. Bach, "Breaking the curse of dimensionality with convex neural networks," Journal of Machine Learning Research **18**, 1–53 (2017).
[44]G. Rotskoff and E. Vanden-Eijnden, "Trainability and accuracy of artificial neural networks: An interacting particle system approach," Commun. Pure Appl. Math. **75**, 1889–1935 (2022).
[45]S. Chen, C. W. Peterson, J. A. Parker, S. A. Rice, A. L. Ferguson, and N. F. Scherer, "Data-driven reaction coordinate discovery in overdamped and non-conservative systems: Application to optical matter structural isomerization," Nat. Commun. **12**, 2548 (2021).
[46]Z. Smith, D. Pramanik, S.-T. Tsai, and P. Tiwary, "Multi-dimensional spectral gap optimization of order parameters (SGOOP) through conditional probability factorization," J. Chem. Phys. **149**, 234105 (2018).
[47]R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, "Automatic chemical design using a data-driven continuous representation of molecules," ACS Cent. Sci. **4**, 268–276 (2018).
[48]A. Mardt, L. Pasquali, H. Wu, and F. Noé, "VAMPnets for deep learning of molecular kinetics," Nat. Commun. **9**, 5 (2018).
[49]H. Sidky, W. Chen, and A. L. Ferguson, "Molecular latent space simulators," Chem. Sci. **11**, 9459–9467 (2020).
[50]J. M. L. Ribeiro, P. Bravo, Y. Wang, and P. Tiwary, "Reweighted autoencoded variational Bayes for enhanced sampling (RAVE)," J. Chem. Phys. **149**, 072301 (2018).
[51]D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in 2nd International Conference on Learning Representations (ICLR 2014), Banff, AB, Canada, 14-16 April 2014, arXiv:1312.6114.
[52]G. Hummer, "Position-dependent diffusion coefficients and free energies from Bayesian analysis of equilibrium and replica molecular dynamics simulations," New J. Phys. **7**, 34 (2005).
[53]C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, "Neural spline flows," in *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019), Vol. 32.
[54]D. J. Tobias and C. L. Brooks, "Conformational equilibrium in the alanine dipeptide in the gas phase and aqueous solution: A comparison of theoretical results," J. Phys. Chem. **96**, 3864–3870 (1992).
[55]B. Montgomery Pettitt and M. Karplus, "The potential of mean force surface for the alanine dipeptide in aqueous solution: A theoretical approach," Chem. Phys. Lett. **121**, 194–201 (1985).
[56]N. Deng, B. W. Zhang, and R. M. Levy, "Connecting free energy surfaces in implicit and explicit solvent: An efficient method to compute conformational and solvation free energies," J. Chem. Theory Comput. **11**, 2868–2878 (2015).
[57]S. Honda, T. Akiba, Y. S. Kato, Y. Sawada, M. Sekijima, M. Ishimura, A. Ooishi, H. Watanabe, T. Odahara, and K. Harata, "Crystal structure of a ten-amino acid protein," J. Am. Chem. Soc. **130**, 15327–15331 (2008).
[58]G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. Noé, "Identification of slow molecular order parameters for Markov model construction," J. Chem. Phys. **139**, 015102 (2013).
[59]C. R. Schwantes and V. S. Pande, "Improvements in Markov state model construction reveal many non-native interactions in the folding of NTL9," J. Chem. Theory Comput. **9**, 2000–2009 (2013).
[60]K. A. McKiernan, B. E. Husic, and V. S. Pande, "Modeling the mechanism of CLN025 beta-hairpin formation," J. Chem. Phys. **147**, 104107 (2017).
[61]G. Ciccotti, T. Lelièvre, and E. Vanden-Eijnden, "Projection of diffusions on submanifolds: Application to mean force computation," Commun. Pure Appl. Math. **61**, 371–408 (2008).
[62]E. Kalligiannaki, V. Harmandaris, M. A. Katsoulakis, and P. Plecháč, "The geometry of generalized force matching and related information metrics in coarse-graining of molecular systems," J. Chem. Phys. **143**, 084105 (2015).
[63]D. A. Sivak, J. D. Chodera, and G. E. Crooks, "Time step rescaling recovers continuous-time dynamical properties for discrete-time Langevin integration of nonequilibrium systems," J. Phys. Chem. B **118**, 6466–6474 (2014).
[64]P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, R. P. Wiewiora, B. R. Brooks, and V. S. Pande, "OpenMM 7: Rapid development of high performance algorithms for molecular dynamics," PLOS Comput. Biol. **13**, e1005659 (2017).