UvA-DARE (Digital Academic Repository)

Two for One: Diffusion Models and Force Fields for Coarse-Grained Molecular Dynamics

Arts, M.; García Satorras, V.; Huang, C.-W.; Zügner, D.; Federici, M.; Clementi, C.; Noé, F.; Pinsler, R.; van den Berg, R.

Citation for published version (APA):
Arts, M., García Satorras, V., Huang, C.-W., Zügner, D., Federici, M., Clementi, C., Noé, F., Pinsler, R., & van den Berg, R. (2023). Two for One: Diffusion Models and Force Fields for Coarse-Grained Molecular Dynamics. *Journal of Chemical Theory and Computation*, *19*(18), 6151-6159. https://doi.org/10.1021/acs.jctc.3c00702

Two for One: Diffusion Models and Force Fields for Coarse-Grained Molecular Dynamics

Marloes Arts,*,¶ Victor Garcia Satorras,*,¶ Chin-Wei Huang, Daniel Zügner, Marco Federici, Cecilia Clementi, Frank Noé, Robert Pinsler, and Rianne van den Berg

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Coarse-grained (CG) molecular dynamics enables the study of biological processes at temporal and spatial scales that would be intractable at an atomistic resolution. However, accurately learning a CG force field remains a challenge. In this work, we leverage connections between score-based generative models, force fields, and molecular dynamics to learn a CG force field without requiring any force inputs during training. Specifically, we train a diffusion generative model on protein structures from molecular dynamics simulations, and we show that its score function approximates a force field that can directly be used to simulate CG molecular dynamics. While having a vastly simplified training setup compared to previous work, we demonstrate that our approach leads to improved performance across several protein simulations for systems up to 56 amino acids, reproducing the CG equilibrium distribution and preserving the dynamics of all-atom simulations such as protein folding events.
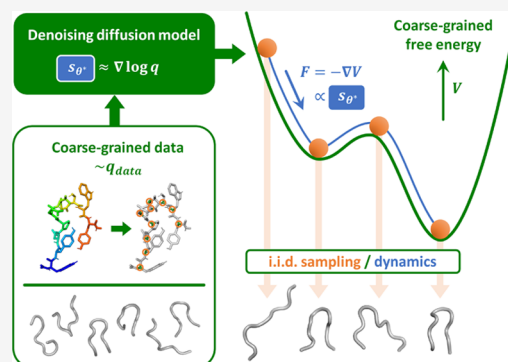
1. INTRODUCTION

Coarse-grained (CG) molecular dynamics (MD) promises to scale simulations to larger spatial and time scales than are currently accessible through atomistic MD simulations.[1−4] Scaling up MD by orders of magnitude would enable new studies on macromolecular dynamics over longer ranges of time, such as large protein folding events and slow interactions between large molecules.

To obtain a CG simulation model, one first maps the all-atom, or fine-grained, representation to a CG representation, e.g., by grouping certain atoms together to form so-called CG beads. Second, a CG force field needs to be designed such that CG MD simulations reproduce relevant features of molecular systems.

In top-down approaches, a CG model is often defined to reproduce specific macroscopic observables as experimentally measured and/or simulated on fine-grained models.[5−8] In bottom-up approaches, one seeks to obtain a CG model reproducing the microscopic behavior (e.g., thermodynamics and kinetics) of a fine-grained model.[9−11] In the latter case, a common approach is to define a CG force field for the chosen CG representation by enforcing thermodynamic consistency.[2] This requires that simulations following the CG model have the same equilibrium distribution as those obtained by projecting equilibrated all-atom simulations onto the CG resolution.

Traditional bottom-up coarse-graining techniques that rely on the thermodynamic consistency principle have produced significant results in the last decade,[12−14] in particular when used in combination with machine learning methods.[15,16] Two commonly used approaches are variational force matching and relative entropy minimization.

Variational force matching minimizes the mean squared error between the model's CG forces and the atomistic forces projected onto the CG space, which must be included in the data.[9] However, due to the stochastic nature of the projected forces, this noisy force-matching estimator has a large variance, leading to data-inefficient training. Alternatively, relative entropy minimization approaches[10] perform density estimation in the CG space without accessing atomistic forces. The majority of this class of methods are equivalent to energy-based models.[17] Since training these models requires iteratively drawing samples from the model to estimate log-likelihood gradients, such methods demand significantly higher computational costs.[18]

Flow-matching[19] is a hybrid approach that does not require atomistic forces for training (like relative entropy minimization) while also retaining good sample efficiency. The method has two training stages. First, a CG density is modeled with an augmented normalizing flow.[20−23] A second learning stage with a force-matching-like objective is then required to extract
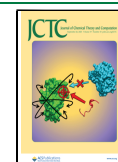
a deterministic CG force field that can be used in CG MD simulations. Köhler et al.[19] demonstrated that flow-matching improves performance on several fast-folding proteins.[24] However, the learned CG models are not yet accurate enough to reproduce the thermodynamics of the corresponding fine-grained models, and scaling to larger proteins leads to instability.

In this work, we leverage the recently popularized class of denoising diffusion models,[25,26] which have already shown promising results for protein structure generation,[27−31] conformer generation,[32] and docking.[33] In particular, we train a score-based generative model on CG structures sampled from the CG equilibrium distribution. By highlighting connections between score-based generative models,[34] force fields, and MD, we demonstrate that learning such a generative model with a standard denoising loss and a conservative score yields a single model that can be used to produce i.i.d. CG samples and which can be used directly as a CG force field for CG MD simulations. An overview is shown in Figure 1. In addition to
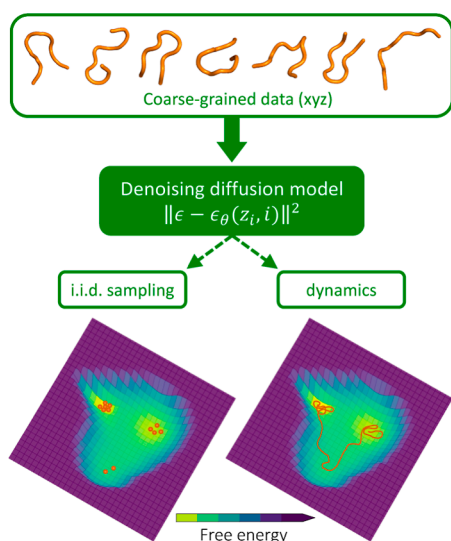

Figure 1. Denoising diffusion model is trained with a standard loss on atomistic (fine-grained) equilibrium samples projected onto the CG space. By leveraging connections between score-based generative modeling, force fields, and MD, we obtain a single model that can generate i.i.d. equilibrium CG samples and whose neural network can be used as a CG force field in CG MD simulations.

having a single-stage training setup, our method leads to improved performance across several protein simulations for systems up to 56 amino acids, reproducing the CG equilibrium distribution, and preserving the dynamical mechanisms observed in all-atom simulations such as protein folding events. We also provide evidence that our diffusion CG model allows for scaling to a larger protein than previously accessible through flow-matching.

2. BACKGROUND

Coarse graining can be described by a dimensionality reduction map $\Xi: \mathbb{R}^{3N} \rightarrow \mathbb{R}^{3n}$ that transforms a high-dimensional atomistic representation $\mathbf{x} \in \mathbb{R}^{3N}$ in 3D space to a lower-dimensional CG representation $\mathbf{z} \in \mathbb{R}^{3n}$, where $n \ll N$. For molecular systems, the CG map is usually linear, $\Xi \in \mathbb{R}^{3n \times 3N}$, and returns the Cartesian coordinates $\mathbf{z}$ of CG

"beads" as a linear combination of the Cartesian coordinates $\mathbf{x}$ of a set of representative atoms.

The probability density of the atomistic system at a particular temperature $\mathcal{T}$ is described by the Boltzmann distribution $q(\mathbf{x}) \propto \exp(-U(\mathbf{x})/k_B\mathcal{T})$, where $U(\mathbf{x})$ is the system's potential energy, and k_B is the Boltzmann constant. By identifying the ensemble of atomistic configurations $\mathbf{x}$ that map into the same CG configuration $\mathbf{z}$, we can explicitly express the probability density of the CG configurations $\mathbf{z}$ as

$$q(\mathbf{z}) = \frac{\int \exp(-U(\mathbf{x})/k_B\mathcal{T})\delta(\Xi(\mathbf{x}) - \mathbf{z})d\mathbf{x}}{\int \exp(-U(\mathbf{x}')/k_B\mathcal{T})d\mathbf{x}'} \quad (1)$$

where $\delta(\cdot)$ is the Dirac delta function. Up to an additive constant, this distribution uniquely defines the thermodynamically consistent effective CG potential of mean force $V(\mathbf{z})$[9]

$$V(\mathbf{z}) = -k_B\mathcal{T}\log q(\mathbf{z}) + \text{cst}$$

$$= -k_B\mathcal{T}\log \int e^{-U(\mathbf{x})/k_B\mathcal{T}}\delta(\Xi(\mathbf{x}) - \mathbf{z})d\mathbf{x} + \text{cst}$$

Unfortunately, computing the integral is usually intractable. Therefore, methods that approximate thermodynamically consistent effective CG potentials have been proposed. Below, we briefly summarize two commonly used approaches.

2.1. Variational Force Matching. Noid et al.[9] showed that under certain constraints of the coarse-graining mapping Ξ, a more tractable consistency equation between the CG force field $-\nabla_{\mathbf{z}}V(\mathbf{z})$, and the atomistic force field $-\nabla_{\mathbf{x}}U(\mathbf{x})$ can be obtained. More specifically, if Ξ is a linear map and if each bead has at least one atom with a nonzero coefficient only for that specific bead, then the following relation holds: $-\nabla_{\mathbf{z}}V(\mathbf{z}) = \mathbb{E}_{q(\mathbf{x}|\mathbf{z})}[\Xi_f(-\nabla_{\mathbf{x}}U(\mathbf{x}))]$. Here, Ξ_f is a linear map whose coefficients are related to the linear coefficients of the CG map Ξ.[35] Noid et al.[9] showed that the above relation can be used to approximate a thermodynamically consistent CG potential $V_\theta(\mathbf{z})$ with parameters θ by minimizing the following variational loss

$$\mathbb{E}_{q(\mathbf{x},\mathbf{z})}[\|\nabla_{\mathbf{z}}V_\theta(\mathbf{z}) - \Xi_f(\nabla_{\mathbf{x}}U(\mathbf{x}))\|_2^2] \quad (2)$$

2.2. Relative Entropy Minimization. Another approach to obtaining the CG forces is via relative entropy minimization, where optimizing the density implicitly leads to optimized mean potential functions. Concretely, we seek to estimate the CG density by minimizing the relative entropy, or Kullback−Leibler divergence, $\mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z}) - \log p_\theta(\mathbf{z})]$, which is equivalent to optimizing the maximum likelihood when a finite number of samples is drawn from $q(\mathbf{z})$. The approximate CG forces can be extracted from the optimized model density $p_\theta(\mathbf{z})$ through $-\nabla_{\mathbf{z}}V_\theta(\mathbf{z}) \propto \nabla_{\mathbf{z}}\log p_\theta(\mathbf{z})$. Unlike variational force matching, relative entropy minimization does not impose any constraints on the CG map, and no atomistic forces are required for training.

Traditionally, an unnormalized version of p_θ is modeled by directly parameterizing the CG potential V_θ, yielding $p_\theta(\mathbf{z}) \propto \exp(-V_\theta(\mathbf{z})/k_B\mathcal{T})$. To minimize the relative entropy, one would need to either estimate the free energy (i.e., the normalizing constant) of the model[10] or draw i.i.d. samples from the model for gradient estimation,[18,36] which renders this approach impractical for higher-dimensional problems.

An alternative is to use an explicit density model such as a normalizing flow,[20,37,38] which allows for straightforward maximum-likelihood density estimation and force field learning. However, learning expressive invertible functions is challenging, so instead, Köhler et al.[19] opted for augmented normalizing flows.[22,23] The introduction of auxiliary random variables increases the expressivity of the flow at the cost of an intractable marginal likelihood, yielding a minimization objective that is a variational upper bound to the relative entropy. Furthermore, one can only extract a stochastic estimate for the CG force from the augmented normalizing flow model. In order to distill a deterministic approximate CG force to simulate the CG dynamics, Köhler et al.[19] proposed a teacher-student setup akin to variational force-matching. This two-stage approach was dubbed flow-matching.

3. DIFFUSION MODELS FOR CG MD

Denoising diffusion probabilistic models (DDPMs)[25,26] sample from a probability distribution by approximating the inverse of a diffusion process, i.e., a denoising process. The diffusion (forward) process is defined as a Markov chain of L steps $q(\mathbf{z}_{1:L}|\mathbf{z}_0) = \prod_{i=1}^{L} q(\mathbf{z}_i|\mathbf{z}_{i-1})$, where $\mathbf{z}_0$ is a sample from the unknown data distribution $q(\mathbf{z}_0)$. The learned reverse process is defined as a reverse-time Markov chain of L denoising steps $p(\mathbf{z}_{0:L}):= p(\mathbf{z}_L)\prod_{i=1}^{L} p_\theta(\mathbf{z}_{i-1}|\mathbf{z}_i)$ that starts from the prior $p(\mathbf{z}_L)$. For real-valued random variables, the choice of distribution for the forward process is typically Gaussian, $q(\mathbf{z}_i|\mathbf{z}_{i-1}) = \mathcal{N}(\mathbf{z}_i; \sqrt{1-\beta_i}\mathbf{z}_{i-1}, \beta_i\mathbf{I})$, with $\{\beta_i\}$ predetermined variance parameters that increase as a function of i such that the Markov chain has a standard normal stationary distribution. The reverse process distributions are chosen to have the same functional form: $p(\mathbf{z}_L) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $p_\theta(\mathbf{z}_{i-1}|\mathbf{z}_i) = \mathcal{N}(\mathbf{z}_{i-1}; \mu_\theta(\mathbf{z}_i, i), \sigma_i^2\mathbf{I})$. Here, $\mu_\theta(\mathbf{z}_i, i)$ is a learnable function with parameters θ, and σ_i^2 is a fixed variance for noise level i that is determined by β_i. By making use of closed-form marginalization for Gaussian distributions and by parameterizing the means as $\mu_\theta(\mathbf{z}_i, i) = \frac{1}{\sqrt{\alpha_i}}\left(\mathbf{z}_i - \frac{\beta_i}{\sqrt{1-\bar{\alpha}_i}}\epsilon_\theta(\mathbf{z}_i, i)\right)$, with $\epsilon_\theta(\mathbf{z}_i, i)$ the noise prediction neural network, training proceeds by minimizing the loss[25]

$$\sum_{i=1}^{L} K_i \mathbb{E}_{q(\mathbf{z}_0)}\mathbb{E}_{\mathcal{N}(\varepsilon;\mathbf{0},\mathbf{I})}\left[\left\|\varepsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_i}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_i}\varepsilon, i)\right\|^2\right]$$

(3)

Here, $\alpha_i = 1 - \beta_i$ and $\bar{\alpha}_i = \prod_{s=1}^{i}\alpha_s$. Up to a constant, eq 3 is a negative evidence lower bound if $K_i = \frac{\beta_i^2}{2\sigma_i^2\alpha_i(1-\bar{\alpha}_i)}$. However, Ho et al.[25] found that a reweighted loss with $K_i = 1$ worked best in practice.

In this study, the data consists of samples from the CG Boltzmann distribution: $q(\mathbf{z}_0) \propto e^{-V(\mathbf{z})/k_B\mathcal{T}}$. Given a trained diffusion model parameterized through a noise prediction network $\epsilon_\theta(\mathbf{z}_i, i)$, we can produce i.i.d. samples of the approximate CG distribution through ancestral sampling from the graphical model $p(\mathbf{z}_L)\prod_{i=1}^{L} p_\theta(\mathbf{z}_{i-1}|\mathbf{z}_i)$.

3.1. Extracting Force Fields from Diffusion Models.
Song et al.[34] demonstrated that the DDPM loss in eq 3 with K_i

= 1 is equivalent to the following weighted sum of denoising score matching objectives[39]

$$\sum_{i=1}^{L} (1 - \bar{\alpha}_i)\mathbb{E}_{q(\mathbf{z}_0)}\mathbb{E}_{q(\mathbf{z}_i|\mathbf{z}_0)}\left[\left\|s_\theta(\mathbf{z}_i, i) - \nabla_{\mathbf{z}_i}\log q(\mathbf{z}_i|\mathbf{z}_0)\right\|^2\right]$$

(4)

Here, $q(\mathbf{z}_i|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_i; \sqrt{\bar{\alpha}_i}\mathbf{z}_0, (1 - \bar{\alpha}_i)\mathbf{I})$, and $s_\theta(\mathbf{z}_i, i)$ is the score model. While this was not made explicit by Song et al.,[34] the equivalence of these two losses is achieved by relating the score model $s_\theta(\mathbf{z}_i, i)$ to the noise prediction network $\epsilon_\theta(\mathbf{z}_i, i)$ through $s_\theta(\mathbf{z}_i, i) = -\frac{\epsilon_\theta(\mathbf{z}_i, i)}{\sqrt{1-\bar{\alpha}_i}}$, see Supporting Information Section SA.1. Given a sufficiently expressive model and sufficient amounts of data, the optimal score $s_{\theta*}(\mathbf{z}_i, i)$ will match the score, $\nabla_{\mathbf{z}_i}\log q(\mathbf{z}_i)$[39] where $q(\mathbf{z}_i) = \int d\mathbf{z}_0 q(\mathbf{z}_i|\mathbf{z}_0)q(\mathbf{z}_0)$ is the marginal distribution at level i of the forward diffusion process. At sufficiently low noise levels, the marginal distribution $q(\mathbf{z}_i)$ will resemble the data distribution $q(\mathbf{z}_0)$, such that $s_{\theta*}(\mathbf{z}_i, i)$ effectively approximates the score of the unknown data distribution. When the latter is equal to the CG Boltzmann distribution $q(\mathbf{z}_0) \propto e^{-V(\mathbf{z})/k_B\mathcal{T}}$, the optimal score $s_{\theta*}(\mathbf{z}_i, i)$ at level $i = 1$ will approximately match the CG forces $\nabla_{\mathbf{z}}\log q(\mathbf{z}) = \frac{-\nabla_{\mathbf{z}}V(\mathbf{z})}{k_B\mathcal{T}} = \frac{\mathbf{F}_{\mathbf{z}}}{k_B\mathcal{T}}$. Finally, by using the relation between $s_\theta(\mathbf{z}_i, i)$ and the noise prediction network $\epsilon_\theta(\mathbf{z}_i, i)$, we can extract the approximate CG forces from a denoising diffusion model trained with the loss in eq 3

$$\mathbf{F}_{\mathbf{z}}^{\text{DFF}} = -\frac{k_B\mathcal{T}}{\sqrt{1-\bar{\alpha}_i}}\epsilon_{\theta*}(\mathbf{z}, i)$$

(5)

We will refer to such an approximate CG force field as a denoising force field (DFF). While in principle the lowest level ($i = 1$) should provide the best approximation to the CG forces, in practice we treat i as a hyperparameter and pick the best i by cross-validating the simulated dynamics.

Connections between force fields and denoising diffusion models have been made in previous work. Zaidi et al.[40] pre-trained a property prediction graph neural network in a denoising diffusion setup by denoising molecular structures that locally maximize the Boltzmann distribution (or minimize the energy). By approximating the data distribution as a mixture of Gaussians centered around these local minima, they demonstrate that the score matching objective is equivalent to learning the force field of this approximate mixture of Gaussian data distribution. Similarly, Xie et al.[41] connected the learned score in a denoising network for small noise levels to a harmonic force field around energy local minima structures. A key point is that these connections only provide approximate force fields around the local minima structures, making them of limited use in downstream tasks. In this work, we show that training denoising diffusion models on samples from the equilibrium Boltzmann distribution—rather than only the locally maximizing structures—allows us to learn an approximate force field in an unsupervised manner for the entire equilibrium distribution. This is crucial for running stable and reliable CG MD simulations with the extracted CG force field.

3.2. MD with the DFF.
With the DFF from eq 5, we can perform CG MD simulations by propagating the Langevin equation

$$M\frac{d^2\mathbf{z}}{dt^2} = -\nabla_{\mathbf{z}}V(\mathbf{z}) - \gamma M\frac{d\mathbf{z}}{dt} + \sqrt{2M\gamma k_B \mathcal{T}}\,\mathbf{w}(t) \qquad (6)$$

where we substitute $-\nabla_{\mathbf{z}}V(\mathbf{z}) = \mathbf{F}_{\mathbf{z}}^{\mathrm{DFF}}$. M represents the mass of the CG beads, γ is a friction coefficient, and $\mathbf{w}(t)$ is a delta-correlated stationary Gaussian process $\mathbb{E}_{p(\mathbf{x})}[\mathbf{w}(t)\cdot\mathbf{w}(t')] = \delta(t - t')$ with mean $\mathbb{E}_{p(\mathbf{x})}[\mathbf{w}(t)] = 0$. In our experiments, we set γ and $\mathcal{T}$ to the same values as those used in the original atomistic simulations that produced the data. Therefore, given a trained network ε_θ, the only hyperparameter left to tune is the noise level i. Further information regarding the trade-off of noise level i can be found in Supporting Information Section SB.3.

A well-known limit of the Langevin equation (eq 6) is that of a negligible mass and a large friction coefficient (with a finite $\eta = \gamma M$), called Brownian dynamics or overdamped Langevin dynamics. Interestingly, in Supporting Information Section SA.2, we show that iteratively diffusing and denoising at a low noise level (e.g., $i = 1$) approximates Brownian dynamics with a simulation timestep Δt implicitly defined through $\Delta t \frac{k_B \mathcal{T}}{M\gamma} = 1 - \overline{\alpha}_1 = \beta_1$.

3.3. DFF Architecture. The choice of the neural network ε_θ is heavily influenced by the physical symmetries of the system under study. For instance, the CG force field must be conservative, i.e., it must equal the negative gradient of the CG energy potential $V_\theta(\mathbf{z})$. Therefore, we parameterize $\varepsilon_\theta(\mathbf{z}, i)$ as the gradient of an energy neural network with a scalar output, i.e., $\varepsilon_\theta(\mathbf{z}, i) = \nabla_{\mathbf{z}_i}\mathrm{nn}_\theta(\mathbf{z}, i)$, with $\mathrm{nn}_\theta: \mathbb{R}^{3n} \times \{1, \ldots, L\} \to \mathbb{R}$. Previous studies on image generation by Salimans and Ho[42] yielded no empirical difference in sample quality when using an unconstrained score network or a score that is parameterized as the gradient of an energy function. However, in Supporting Information Section SB.1, we demonstrate that using a conservative score in a diffusion model is crucial for stable CG MD simulations with the extracted DFF.

Furthermore, the force field must be translation-invariant and rotation-equivariant. We ensure the model is translation-invariant by using the coordinates of the CG beads only through pairwise difference vectors $\mathbf{z}_{(i)} - \mathbf{z}_{(j)}$ as input to the network. While the forces must be equivariant to rotations, we explicitly do not want reflection equivariance to avoid generating mirrored proteins, as reported by Trippe et al.[28] In other words, our goal is to achieve equivariance with respect to SO(3) instead of O(3), as opposed to other works that use relative distances as the input representation.[43] A simple strategy to approximate SO(3) equivariance without requiring the more expensive spherical harmonics or angular representations is the use of data augmentation. Previous work by Gruver et al.[44] showed that learned equivariance with transformers can be competitive with actual equivariant networks. In Supporting Information Section SB.4, we show that our DFF learns to be rotation-equivariant on a validation set with a relative squared error introduced by rotations of $<10^{-6}$.

In this work, we model the network nn_θ as a graph transformer adapted to the above symmetry constraints. Further architecture details are given in Supporting Information Section SC.1. Note that previous works on neural-network-based CG force fields also often add a prior energy term in the scalar energy neural network to enforce better behavior of the CG force field further away from the training

dataset.[15,16,19] In contrast, we did not find this to be necessary to obtain stable CG MD simulations with our denoising CG force field.

4. EXPERIMENTS

By training our diffusion model on samples from a CG equilibrium distribution, we simultaneously obtain an i.i.d. sample generator (denoted DFF i.i.d.) as well as a CG force field for running CG MD simulations (DFF sim.). In this section, we evaluate the performance and scalability of our model for both use cases on (i) alanine dipeptide and (ii) several fast-folding proteins.[24] In particular, we investigate how well the CG equilibrium distribution and the dynamics can be reproduced.

We compare our model to three baselines: Flow i.i.d. and Flow-CGNet sim. from Köhler et al.[19] and CGNet sim.[15] CGNet sim. is a pure force-matching neural network trained on CG forces that were projected from the fine-grained representation onto the CG representation. Flow i.i.d. is the force-agnostic augmented normalizing flow model trained as a density estimator in the first stage of the flow-matching setup.[19] This flow model can only be used to produce i.i.d. samples. Flow-CGNet sim. performs CG simulations using the deterministic CGNet force field distilled from the gradient of the augmented normalizing flow model in the second teacher–student distillation stage of flow-matching. Recall that for our method, we do not require a teacher–student setup since the same network can be used for i.i.d. sampling and for CG simulations. We also provide reference data, which is the original MD simulation projected onto the CG resolution. Lastly, note that while we often show results for both i.i.d. and simulation-based methods, the latter have the more challenging task of modeling the dynamics in order to obtain correct equilibrium distributions. We therefore expect the proposed i.i.d. methods to perform better when analyzing equilibrium distributions.

4.1. CG Simulation—Alanine Dipeptide. First, we evaluate our method on a CG representation of the well-studied alanine dipeptide system. We use the same CG representation as those of Wang et al.,[15] Husic et al.,[16] and Köhler et al.,[19] which project all atoms onto the five central backbone atoms of the molecule (see Supporting Information Figure S4). The simulated data[19] consists of four independent runs of length 500 ns, with 250,000 samples saved per simulation (2 ps intervals). We evaluate the model using four-fold cross-validation, where three of the simulations are used for training and validation and one is used for testing. We consider different training dataset sizes, ranging from 10 to 500K training samples. For the Langevin dynamics simulation, we follow the same settings as Köhler et al.,[19] i.e., we run the simulation at 300 kelvin for 1 M steps with a step size of 2 fs and store the samples every 250 time steps. However, unlike Köhler et al.,[19] we do not use parallel tempering, which is known to improve the mixing of the dynamics. Our denoising network nn_θ (Section 3.3) consists of two graph transformer layers with 96 features in the hidden layers. Further implementation details are in Supporting Information Section SC.4.1.

4.1.1. Metrics. Following Wang et al.[15] and Köhler et al.,[19] we evaluate the quality of the generated samples by analyzing statistics over the two dihedral angles (ϕ, ψ) computed along the CG backbone of alanine dipeptide. Each angle describes a four-body interaction, representing the main degrees of
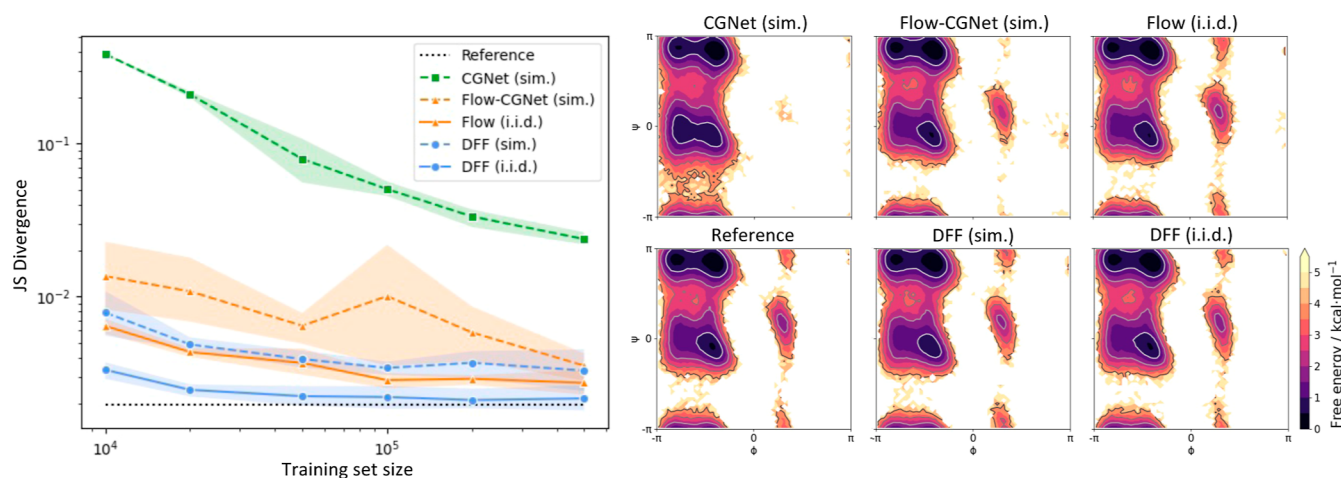
Figure 2. Experimental results for alanine dipeptide. Left: JS divergence between dihedral distributions produced by several CG methods for different training set sizes and the test partition of all-atom simulation data projected onto the CG resolution. Results are averaged over four runs, and error bars denote a 95% confidence interval. Right: Ramachandran plots showing the dihedral distributions for the different methods trained on 500K samples.

Table 1. Experimental Results for Fast Folders[a]

		Chignolin		Trp-cage		Bba		Villin		Protein G	
		TIC JS	PWD JS	TIC JS	PWD JS	TIC JS	PWD JS	TIC JS	PWD JS	TIC JS	PWD JS
reference		0.0057	0.0002	0.0026	0.0002	0.0040	0.0002	0.0032	0.0004	0.0014	0.0002
flow	i.i.d.	0.0106	0.0022	0.0078	0.0057	0.0229	0.0073	0.0109	0.0142	n/a	n/a
DFF		**0.0096**	**0.0005**	**0.0052**	**0.0007**	**0.0111**	**0.0017**	**0.0073**	**0.0009**	0.0131	0.0009
flow-CGNet	sim.	0.1875	0.1271	0.1009	0.0474	0.1469	0.0594	0.2153	0.0535	n/a	n/a
DFF		**0.0335**	**0.0067**	**0.0518**	**0.0403**	**0.1289**	**0.0408**	**0.0564**	**0.0244**	0.2260	0.0691

[a]The table displays the JS divergence for TIC distributions and pairwise distance (PWD) distributions, where in the latter case an average is taken over all entries of the upper triangle of the PWD matrix with offset three. The JS divergences compare distributions from the atomistic MD simulations that were projected on CG space with the distributions produced by the learned CG methods.

freedom of the system. We generate a Ramachandran plot by computing the free energy as a function of these two angles, binning values into a 2D histogram, and taking the negative logarithm of the probability density. To provide a quantitative analysis, we measure the empirical Jensen-Shannon (JS) divergence between the dihedral distributions of samples drawn from the model and the test set. A reference comparing the training and test sets is provided as a lower bound.

4.1.2. Results. As shown in Figure 2 (left), DFF sim. significantly outperforms previous CG simulation methods (Flow-CGNet sim. and CGNet sim.), especially in the low-data regime, and even performs comparably to the i.i.d. sampling method Flow i.i.d. Moreover, DFF i.i.d. outperforms its counterpart, Flow i.i.d., with a significant margin, almost approaching the performance of the lower bound (reference). The right side of Figure 2 shows the Ramachandran plots after training on 500 K samples, further highlighting that our model is able to generate realistic samples.

4.2. CG Simulation—Fast-Folding Proteins. Next, we evaluate our model on a more challenging set of fast-folding proteins.[24] Such proteins exhibit folding and unfolding events, which makes their simulated trajectories particularly interesting. We pick the same proteins as in Köhler et al.,[19] namely, Chignolin, Trp-cage, Bba, and Villin. These were CG by slicing out the C_α atom for every amino acid, yielding one bead per residue (10, 20, 28, and 35 beads, respectively). For these proteins, we produced the Flow i.i.d. and Flow-CGNet sim. plots through samples that were made publicly available by the

authors. Since scaling to larger proteins was found to be challenging for flow-matching,[19] we additionally included the larger "Protein G" (56 beads) to analyze the scalability of our method. All-atom simulations vary in length, but for each trajectory, the frames are shuffled and split 70−10−20% into a training, validation, and test set. More dataset details are given in Supporting Information Section SC.5.1.

4.2.1. Equilibrium Analysis. 4.2.1.1. Metrics. We use several metrics to evaluate the quality of the generated equilibrium distributions. First, we analyze the slowest changes in the protein conformation, which are usually related to (un-)folding events. For this, we calculate the time-lagged independent component analysis[45−47] using the Deeptime library[48] and pick the first two TIC coordinates, resulting in a 2D distribution over the slowest processes. Basins in these 2D distributions are associated with meta-stable states. Furthermore, we compute the JS divergence of the obtained TIC distributions between each model and the reference MD data (denoted by TIC JS). As a qualitative analysis, we plot the log of the obtained TIC distributions.

To assess the global structure of the proteins, we compare pairwise distance distributions by calculating the JS divergence relative to the test MD distribution for all distances within the upper triangle of the pairwise distance matrix with a diagonal offset larger than three (denoted by PWD JS). The offset is chosen to avoid over-representing the local structure. Moreover, we plot the free energy as a function of the root mean squared distance (RMSD) between the generated samples and
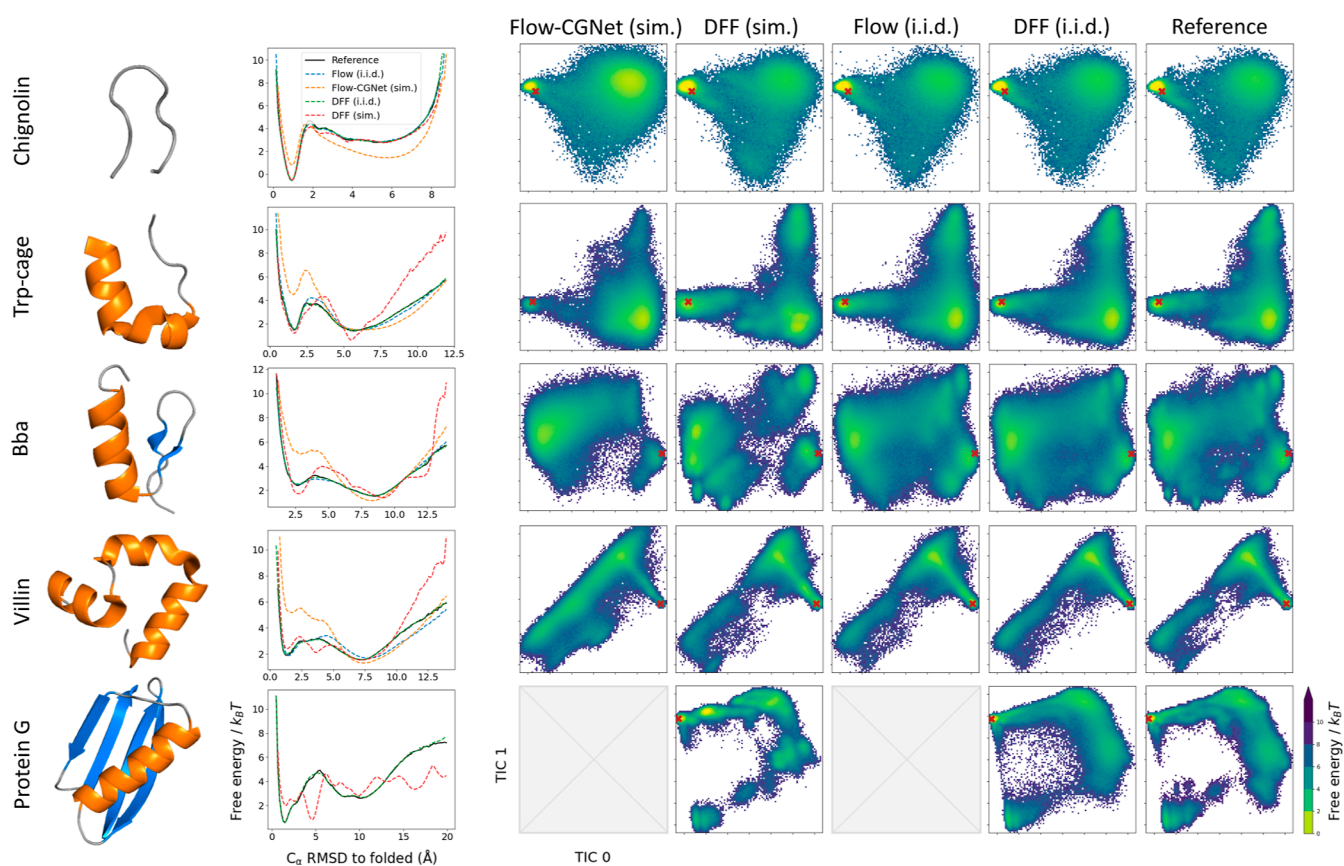
Figure 3. Left: native structure visualization with α-helices in orange and β-sheets in blue. Middle: C_α-RMSD free energy with respect to the folded native structure. Right: joint density plots for the two slowest TIC coordinates, where the color indicates the free energy value. The red cross indicates the location of the native structure.
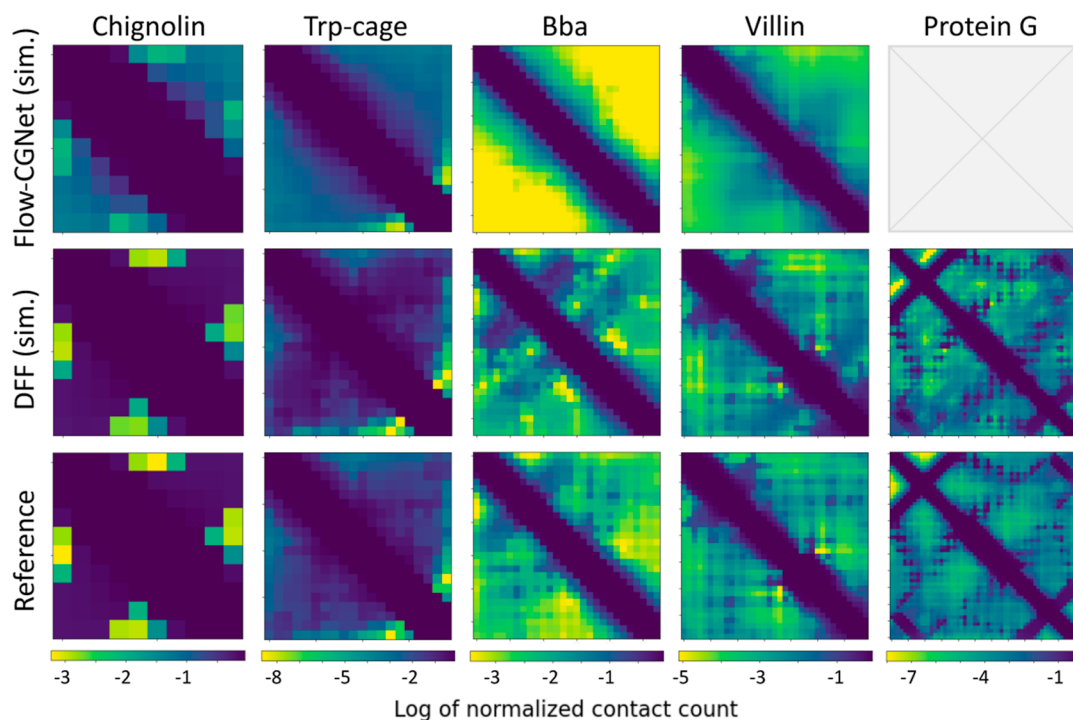

Figure 4. Contact probability maps for the Flow-CGNet sim., DFF sim., and the reference MD data for fast-folding proteins. The contact threshold is set to 10 Å. The axes in the plot represent atom indices. The color indicates the normalized contact count (i.e., "contact probability") for the corresponding pairwise distance.
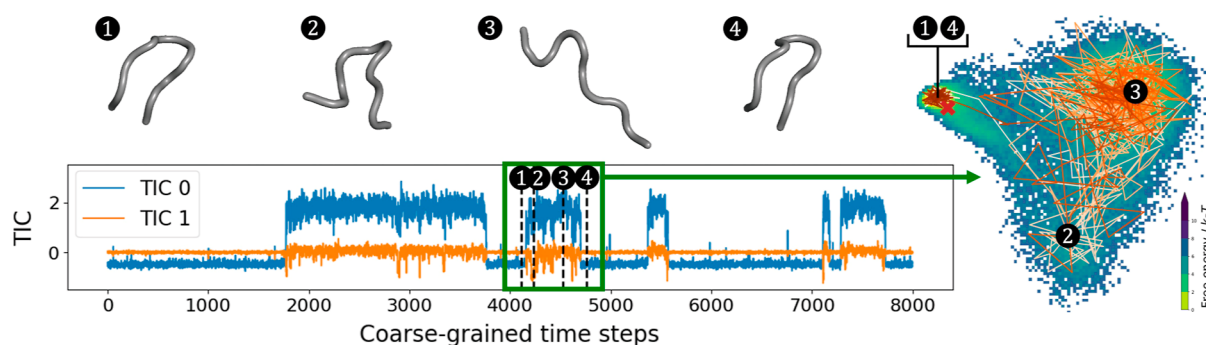
Figure 5. First two TIC coordinates tracked over "CG time" (the CG time step is 2 fs, but there is no direct mapping from coarse- to fine-grained time) for Chignolin, with zoom-in on an unfolding and folding event, showing a 2D trajectory through TIC space and four structures along the path.

Table 2. Average and State-Probability-Weighted JS Divergence between the Reference MD Data and Model Simulations for Transition Probabilities of the Estimated Markov Model

	Chignolin		Trp-cage		Bba		Villin	
	average	weighted	average	weighted	average	weighted	average	weighted
flow-CGNet	2.5×10^{-2}	5.7×10^{-3}	4.8×10^{-2}	1.8×10^{-2}	6.9×10^{-2}	$6.8 \cdot 10^{-2}$	3.1×10^{-2}	2.9×10^{-2}
DFF	9.7×10^{-4}	5.1×10^{-4}	1.3×10^{-3}	7.5×10^{-4}	4.0×10^{-3}	4.2×10^{-3}	1.2×10^{-4}	2.1×10^{-5}

the native, folded structure for all C_α atoms. Dips in the resulting curve correspond to meta-stable ensembles with lower free energy. Finally, we analyze the normalized count over contact maps, which results in a 2D histogram that shows the probability of two atoms being in contact, i.e., within a threshold of 10 Å of one another.

4.2.1.2. Results. Table 1 shows that DFF i.i.d. and DFF sim. consistently outperform their respective baselines across the equilibrium metrics (TIC JS and PWD JS). As shown in Figure 3, the TIC 2D free energy landscapes for our model look more similar to the reference MD distribution (as reflected in the TIC JS metrics), especially for Chignolin and Villin. The similarity is weaker for Bba, where local modes are more dominant. We hypothesize that this is because β-sheets rely strongly on non-local contacts compared to α-helices, making them notoriously harder to model. This is particularly challenging in the simulation setting, which is sensitive to the bias-variance trade-off (as discussed in Supporting Information Section SB.2). Furthermore, the free energy curves as a function of the RMSD are always overlapping with the reference curve for DFF i.i.d., and are close to the reference MD curve in regions with low free energy for DFF sim.

Figure 4 shows further qualitative results in the form of a normalized count over contact maps, i.e., "contact probabilities", for DFF sim. and Flow-CGNet sim.; for i.i.d. models, see Supporting Information Section SC.5.3. As can be seen, the DFF models capture contact probabilities much better than the flow-based models in all proteins, especially in the off-diagonal regions that represent global structure. These results are closely related to the JS divergences between pairwise off-diagonal distances (Table 1). Taken together, these results indicate that diffusion-based models capture global structures better than their flow-based baselines. Moreover, the analysis in C.5.3 shows that the CG fast folder samples produced by DFF sim. do not display chemical integrity violations such as bond dissociations or backbone crossings; therefore, DFF sim. does not require energy prior as used in Flow-CGNet sim. to run stable simulations. Finally, Protein G is a larger and more

complex protein compared to the other fast-folders and is out of reach for flow-matching models. Our results show that diffusion-based models are scalable to this larger protein and can capture the global structure.

As a limitation in our method, we found that while the DFF i.i.d. model generally improves as we increase the number of features/layers in our neural network, the performance of the simulations obtained by DFF sim. is sensitive to the bias/variance trade-off in the network, and it can actually decrease for more flexible networks. See Supporting Information Section SB.2 for an example in Chignolin.

4.2.2. Dynamics Analysis. 4.2.2.1. Metrics. We qualitatively assess the simulated trajectories by tracking the first two TIC coordinates over "CG time" and showing (part of) the corresponding trajectory in 2D TIC space. We visualize (un-)folding events with the corresponding structures along the path. As a quantitative measure, we extract the transition probabilities from one conformational state to the other as follows: first, we use K-means clustering to divide the 2D TIC space into K clusters for the full (unsplit) MD dataset, with K determined by the elbow method. Next, all transitions are counted and normalized to obtain a transition probability matrix corresponding to the estimated Markov model,[49] where each row can be compared to MD data using the JS divergence. Even though the relation between fine-grained and CG time is non-trivial,[11,50] leading to different time lags, we can still evaluate how well a CG model reproduces the kinetic model of the fine-grained reference distribution. We show the average JS divergence over all starting states as well as the average weighted by the overall state probability as estimated from the reference data. Note that this metric can only be calculated for simulation samples (i.e., Flow-CGNet and DFF simulations). Since we compare transition probabilities against the full, unshuffled dataset, there is no test set in this experiment, and therefore, we cannot calculate a reference value here.

4.2.2.2. Results. Figure 5 depicts the first two TIC coordinates for a DFF sim. trajectory in "CG time", clearly showing transitions from the folded to unfolded conforma-

tions. This is further highlighted by zooming in on part of the trajectory, revealing how the trajectory moves in 2D TIC space and what the conformations look like in different parts of the landscape. The results of the transition probability analysis are shown in Table 2 for all fast-folders except Protein G since no Flow-CGNet sim. samples were available for this larger protein. The DFF sim. model outperforms the Flow-CGNet sim. model across all fast-folders, showing better preservation of dynamics. More results on transition probability matrices and the clustering of 2D TIC space are in Supporting Information Section SC.5.3.

5. CONCLUSIONS

We have presented a new approach to CG MD modeling based on denoising diffusion models, motivated by connections between score-based generative models, force fields, and MD. This results in a simple training setup as well as improved performance and scalability compared to previous work. Future directions to improve our work include scaling to larger proteins and generalizing across different systems. Another interesting direction would be to combine the current force-agnostic training approach with an explicit force-matching objective if such force information is available.

■ ASSOCIATED CONTENT

Ⓢ Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.3c00702.

> Relation between score function and noise predicting network; connecting denoising diffusion and Brownian dynamics; ablation on conservative forces; ablation on number of hidden features; ablation on noise level; rotation equivariance through data augmentation; architecture details; coarse-graining operator; optimization objective; dataset details; implementation details; and extended results (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Marloes Arts − *Department of Computer Science, University of Copenhagen, Copenhagen 2100, Denmark;* ⓞ orcid.org/0000-0001-7918-8955; Email: ma@di.ku.dk

Victor Garcia Satorras − *AI4Science, Microsoft Research, Amsterdam 1118 CZ, The Netherlands;* Email: victorgar@microsoft.com

Authors

Chin-Wei Huang − *AI4Science, Microsoft Research, Amsterdam 1118 CZ, The Netherlands*

Daniel Zügner − *AI4Science, Microsoft Research, Berlin 10178, Germany*

Marco Federici − *Informatics Institute, University of Amsterdam, Amsterdam 1098 XH, The Netherlands*

Cecilia Clementi − *AI4Science, Microsoft Research, Berlin 10178, Germany; Department of Physics, Freie Universität Berlin, Berlin 14195, Germany;* ⓞ orcid.org/0000-0001-9221-2358

Frank Noé − *AI4Science, Microsoft Research, Berlin 10178, Germany*

Robert Pinsler − *AI4Science, Microsoft Research, Cambridge CB1 2FB, U.K.*

Rianne van den Berg − *AI4Science, Microsoft Research, Amsterdam 1118 CZ, The Netherlands;* ⓞ orcid.org/0000-0001-5076-2802

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jctc.3c00702

Author Contributions

¶M.A. and V.G.S. contributed equally.

Notes

The authors declare no competing financial interest.
∇M.A. and M.F.: Work done during an internship at Microsoft Research (Amsterdam).

■ REFERENCES

(1) Clementi, C. Coarse-grained models of protein folding: toy models or predictive tools? *Curr. Opin. Struct. Biol.* **2008**, *18*, 10−15.

(2) Noid, W. G. Perspective: Coarse-grained models for biomolecular systems. *J. Chem. Phys.* **2013**, *139*, 090901.

(3) Saunders, M. G.; Voth, G. A. Coarse-graining methods for computational biology. *Annu. Rev. Biophys.* **2013**, *42*, 73−93.

(4) Kmiecik, S.; Gront, D.; Kolinski, M.; Wieteska, L.; Dawid, A. E.; Kolinski, A. Coarse-grained protein models and their applications. *Chem. Rev.* **2016**, *116*, 7898−7936.

(5) Marrink, S. J.; Risselada, H. J.; Yefimov, S.; Tieleman, D. P.; De Vries, A. H. The MARTINI force field: coarse grained model for biomolecular simulations. *J. Phys. Chem. B* **2007**, *111*, 7812−7824.

(6) Davtyan, A.; Schafer, N. P.; Zheng, W.; Clementi, C.; Wolynes, P. G.; Papoian, G. A. AWSEM-MD: protein structure prediction using coarse-grained physical potentials and bioinformatically based local structure biasing. *J. Phys. Chem. B* **2012**, *116*, 8494−8503.

(7) Matysiak, S.; Clementi, C. Minimalist Protein Model as a Diagnostic Tool for Misfolding and Aggregation. *J. Mol. Biol.* **2006**, *363*, 297−308.

(8) Chen, J.; Chen, J.; Pinamonti, G.; Clementi, C. Learning Effective Molecular Models from Experimental Observables. *J. Chem. Theory Comput.* **2018**, *14*, 3849−3858.

(9) Noid, W. G.; Chu, J.-W.; Ayton, G. S.; Krishna, V.; Izvekov, S.; Voth, G. A.; Das, A.; Andersen, H. C. The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models. *J. Chem. Phys.* **2008**, *128*, 244114.

(10) Shell, M. S. The relative entropy is fundamental to multiscale and inverse thermodynamic problems. *J. Chem. Phys.* **2008**, *129*, 144108.

(11) Nüske, F.; Boninsegna, L.; Clementi, C. Coarse-graining molecular systems by spectral matching. *J. Chem. Phys.* **2019**, *151*, 044116.

(12) Mim, C.; Cui, H.; Gawronski-Salerno, J. A.; Frost, A.; Lyman, E.; Voth, G. A.; Unger, V. M. Structural Basis of Membrane Bending by the N-BAR Protein Endophilin. *Cell* **2012**, *149*, 137−145.

(13) Chu, J.-W.; Voth, G. A. Allostery of actin filaments: Molecular dynamics simulations and coarse-grained analysis. *Proc. Natl. Acad. Sci. U.S.A.* **2005**, *102*, 13111−13116.

(14) Yu, A.; Pak, A. J.; He, P.; Monje-Galvan, V.; Casalino, L.; Gaieb, Z.; Dommer, A. C.; Amaro, R. E.; Voth, G. A. A multiscale coarse-grained model of the SARS-CoV-2 virion. *Biophys. J.* **2021**, *120*, 1097−1104.

(15) Wang, J.; Olsson, S.; Wehmeyer, C.; Pérez, A.; Charron, N. E.; De Fabritiis, G.; Noé, F.; Clementi, C. Machine learning of coarse-grained molecular dynamics force fields. *ACS Cent. Sci.* **2019**, *5*, 755−767.

(16) Husic, B. E.; Charron, N. E.; Lemm, D.; Wang, J.; Pérez, A.; Majewski, M.; Krämer, A.; Chen, Y.; Olsson, S.; de Fabritiis, G.; et al. Coarse graining molecular dynamics with graph neural networks. *J. Chem. Phys.* **2020**, *153*, 194101.

(17) Song, Y.; Kingma, D. P. How to train your energy-based models. 2021, arXiv preprint arXiv:2101.03288. https://doi.org/10.48550/arXiv.2101.03288.

(18) Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural Comput.* **2002**, *14*, 1771−1800.

(19) Köhler, J.; Chen, Y.; Krämer, A.; Clementi, C.; Noé, F. Flow-Matching: Efficient Coarse-Graining of Molecular Dynamics without Forces. *J. Chem. Theory Comput.* **2023**, *19*, 942−952.

(20) Rezende, D.; Mohamed, S. Variational inference with normalizing flows. *Proceedings of the 32nd International Conference on Machine Learning*; PMLR, 2015; pp 1530−1538.

(21) Papamakarios, G.; Nalisnick, E. T.; Rezende, D. J.; Mohamed, S.; Lakshminarayanan, B. Normalizing Flows for Probabilistic Modeling and Inference. *J. Mach. Learn. Res.* **2021**, *22*, 1−64.

(22) Huang, C.-W.; Dinh, L.; Courville, A. Augmented normalizing flows: Bridging the gap between generative flows and latent variable models. 2020, arXiv preprint arXiv:2002.07101. https://doi.org/10.48550/arXiv.2002.07101.

(23) Chen, J.; Lu, C.; Chenli, B.; Zhu, J.; Tian, T. Vflow: More expressive generative flows with variational data augmentation. *Proceedings of the 37th International Conference on Machine Learning*; PMLR, 2020; pp 1660−1669.

(24) Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; Shaw, D. E. How fast-folding proteins fold. *Science* **2011**, *334*, 517−520.

(25) Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Info. Process. Syst.* **2020**, *33*, 6840−6851.

(26) Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; Ganguli, S. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. *Proceedings of the 32nd International Conference on Machine Learning*; PMLR, 2015; pp 2256−2265.

(27) Wu, K. E.; Yang, K. K.; Berg, R. v. d.; Zou, J. Y.; Lu, A. X.; Amini, A. P. Protein structure generation via folding diffusion. 2022, arXiv preprint arXiv:2209.15611. https://doi.org/10.48550/arXiv.2209.15611.

(28) Trippe, B. L.; Yim, J.; Tischer, D.; Broderick, T.; Baker, D.; Barzilay, R.; Jaakkola, T. Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem. 2022, arXiv preprint arXiv:2206.04119. https://doi.org/10.48550/arXiv.2206.04119.

(29) Watson, J. L.; Juergens, D.; Bennett, N. R.; Trippe, B. L.; Yim, J.; Eisenach, H. E.; Ahern, W.; Borst, A. J.; Ragotte, R. J.; Milles, L. F.; et al. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. *bioRxiv* **2022**, 12.

(30) Igashov, I.; Stärk, H.; Vignac, C.; Satorras, V. G.; Frossard, P.; Welling, M.; Bronstein, M.; Correia, B. Equivariant 3d-conditional diffusion models for molecular linker design. 2022, arXiv preprint arXiv:2210.05274. https://doi.org/10.48550/arXiv.2210.05274.

(31) Qiao, Z.; Nie, W.; Vahdat, A.; Miller, T. F., III; Anandkumar, A. Dynamic-Backbone Protein-Ligand Structure Prediction with Multi-scale Generative Diffusion Models. 2022, arXiv preprint arXiv:2209.15171. https://doi.org/10.48550/arXiv.2209.15171.

(32) Jing, B.; Corso, G.; Chang, J.; Barzilay, R.; Jaakkola, T. Torsional Diffusion for Molecular Conformer Generation. 2022, arXiv preprint arXiv:2206.01729. https://doi.org/10.48550/arXiv.2206.01729.

(33) Corso, G.; Stärk, H.; Jing, B.; Barzilay, R.; Jaakkola, T. Diffdock: Diffusion steps, twists, and turns for molecular docking. 2022, arXiv preprint arXiv:2210.01776. https://doi.org/10.48550/arXiv.2210.01776.

(34) Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; Poole, B. Score-based generative modeling through stochastic differential equations. 2020, arXiv preprint arXiv:2011.13456. https://doi.org/10.48550/arXiv.2011.13456.

(35) Ciccotti, G.; Kapral, R.; Vanden-Eijnden, E. Blue Moon sampling, vectorial reaction coordinates, and unbiased constrained dynamics. *ChemPhysChem* **2005**, *6*, 1809−1814.

(36) Thaler, S.; Stupp, M.; Zavadlav, J. Deep Coarse-grained Potentials via Relative Entropy Minimization. 2022, arXiv preprint arXiv:2208.10330. https://doi.org/10.48550/arXiv.2208.10330.

(37) Dinh, L.; Krueger, D.; Bengio, Y. Nice: Non-linear independent components estimation. 2014, arXiv preprint arXiv:1410.8516. https://doi.org/10.48550/arXiv.1410.8516.

(38) Dinh, L.; Sohl-Dickstein, J.; Bengio, S. Density estimation using real nvp. 2016, arXiv preprint arXiv:1605.08803. https://doi.org/10.48550/arXiv.1605.08803.

(39) Vincent, P. A connection between score matching and denoising autoencoders. *Neural Comput.* **2011**, *23*, 1661−1674.

(40) Zaidi, S.; Schaarschmidt, M.; Martens, J.; Kim, H.; Teh, Y. W.; Sanchez-Gonzalez, A.; Battaglia, P.; Pascanu, R.; Godwin, J. Pre-training via Denoising for Molecular Property Prediction. 2022, arXiv preprint arXiv:2206.00133. https://doi.org/10.48550/arXiv.2206.00133.

(41) Xie, T.; Fu, X.; Ganea, O.-E.; Barzilay, R.; Jaakkola, T. S. Crystal Diffusion Variational Autoencoder for Periodic Material Generation. 2022, arXiv preprint arXiv:2110.06197. https://doi.org/10.48550/arXiv.2110.06197.

(42) Salimans, T.; Ho, J. Should EBMs model the energy or the score? In *Energy Based Models Workshop*; ICLR, 2021.

(43) Satorras, V. G.; Hoogeboom, E.; Welling, M. E(n) equivariant graph neural networks. In *Proceedings of the 38th International Conference on Machine Learning*; ICLR, 2021; pp 9323−9332.

(44) Gruver, N.; Finzi, M.; Goldblum, M.; Wilson, A. G. The Lie Derivative for Measuring Learned Equivariance. 2022, arXiv preprint arXiv:2210.02984. https://doi.org/10.48550/arXiv.2210.02984.

(45) Naritomi, Y.; Fuchigami, S. Slow dynamics in protein fluctuations revealed by time-structure based independent component analysis: the case of domain motions. *J. Chem. Phys.* **2011**, *134*, 02B617.

(46) Pérez-Hernández, G.; Paul, F.; Giorgino, T.; De Fabritiis, G.; Noé, F. Identification of slow molecular order parameters for Markov model construction. *J. Chem. Phys.* **2013**, *139*, 07B604.

(47) Schwantes, C. R.; Pande, V. S. Improvements in Markov state model construction reveal many non-native interactions in the folding of NTL9. *J. Chem. Theory Comput.* **2013**, *9*, 2000−2009.

(48) Hoffmann, M.; Scherer, M.; Hempel, T.; Mardt, A.; de Silva, B.; Husic, B. E.; Klus, S.; Wu, H.; Kutz, N.; Brunton, S. L.; et al. Deeptime: a Python library for machine learning dynamical models from time series data. *Mach. Learn.: Sci. Technol.* **2021**, *3*, 015009.

(49) Prinz, J.-H.; Wu, H.; Sarich, M.; Keller, B.; Senne, M.; Held, M.; Chodera, J. D.; Schütte, C.; Noé, F. Markov models of molecular kinetics: Generation and validation. *J. Chem. Phys.* **2011**, *134*, 174105.

(50) Jin, J.; Schweizer, K. S.; Voth, G. A. Understanding dynamics in coarse-grained models: I. Universal excess entropy scaling relationship. *J. Chem. Phys.* **2022**, *158*, 034103.

(51) Schrödinger. *The PyMOL Molecular Graphics System*, version 2.5.2, 2021.