

Machine learning coarse-grained potentials of protein thermodynamics

Received: 2 June 2023

Accepted: 29 August 2023

Published online: 15 September 2023

 Check for updates

Maciej Majewski  ^{1,2,14}, Adrià Pérez  ^{1,2,14}, Philipp Thölke  ¹, Stefan Doerr ², Nicholas E. Charron  ^{3,4,5}, Toni Giorgino  ⁶, Brooke E. Husic  ^{7,8,9,10}, Cecilia Clementi  ^{3,4,5,11}  [✉], Frank Noé  ^{5,7,11,12}  [✉] & Gianni De Fabritiis  ^{1,2,13} 

A generalized understanding of protein dynamics is an unsolved scientific problem, the solution of which is critical to the interpretation of the structure-function relationships that govern essential biological processes. Here, we approach this problem by constructing coarse-grained molecular potentials based on artificial neural networks and grounded in statistical mechanics. For training, we build a unique dataset of unbiased all-atom molecular dynamics simulations of approximately 9 ms for twelve different proteins with multiple secondary structure arrangements. The coarse-grained models are capable of accelerating the dynamics by more than three orders of magnitude while preserving the thermodynamics of the systems. Coarse-grained simulations identify relevant structural states in the ensemble with comparable energetics to the all-atom systems. Furthermore, we show that a single coarse-grained potential can integrate all twelve proteins and can capture experimental structural features of mutated proteins. These results indicate that machine learning coarse-grained potentials could provide a feasible approach to simulate and understand protein dynamics.

Proteins are complex dynamical systems that exist in an equilibrium of distinct conformational states, and their multi-state behavior is critical for their biological functions^{1–5}. A complete description of the dynamics of a protein requires the determination of (1) its stable and metastable conformational states, (2) the relative probabilities of these states, and (3) the rates of interconversion among them. Here, we focus on addressing the first two problems by demonstrating how to learn coarse-grained potentials that preserve protein thermodynamics.

Due to the structural heterogeneity of proteins and the ranges of time and length scales over which their dynamics occur, there is no

single technique that is able to successfully model protein behavior across the whole spatiotemporal scale. Computationally, the main method to study protein dynamics has traditionally been molecular dynamics (MD). The first MD simulation ever made was carried out in 1977 on the BPTI protein in vacuum, and only accounted for 9.2 picoseconds of simulation time⁶. As remarked by Karplus & McCammon⁷, these simulations were pivotal towards the realization that proteins are dynamic systems and that those dynamics play a fundamental role in their biological function². When compared with experimental methods such as X-ray crystallography, MD simulations

¹Computational Science Laboratory, Universitat Pompeu Fabra, Barcelona Biomedical Research Park (PRBB), Carrer Dr. Aiguader 88, 08003 Barcelona, Spain.

²Acellera Labs, Doctor Trueta 183, 08005 Barcelona, Spain. ³Department of Physics, Rice University, Houston, TX 77005, USA. ⁴Center for Theoretical Biological Physics, Rice University, Houston, TX 77005, USA. ⁵Department of Physics, FU Berlin, Arnimallee 12, 14195 Berlin, Germany. ⁶Biophysics Institute, National Research Council (CNR-IBF), 20133 Milan, Italy. ⁷Department of Mathematics and Computer Science, FU Berlin, Arnimallee 12, 14195 Berlin, Germany.

⁸Lewis Sigler Institute for Integrative Genomics, Princeton University, Princeton, NJ 08540, USA. ⁹Princeton Center for Theoretical Science, Princeton University, Princeton, NJ 08540, USA. ¹⁰Center for the Physics of Biological Function, Princeton University, Princeton, NJ 08540, USA. ¹¹Department of Chemistry, Rice University, Houston, TX 77005, USA. ¹²Microsoft Research AI4Science, Karl-Liebknecht Str. 32, 10178 Berlin, Germany. ¹³Institució Catalana de Recerca i Estudis Avançats (ICREA), Passeig Lluís Companys 23, 08010 Barcelona, Spain. ¹⁴These authors contributed equally: Maciej Majewski, Adrià Pérez.

 e-mail: cecilia.clementi@fu-berlin.de; frank.noe@fu-berlin.de; gianni.defabritiis@upf.edu

may obtain a complete description of the dynamics in atomic resolution. This information can explain slow events at the millisecond or microsecond timescale, typically with a femtosecond time resolution.

In the last several decades, there have been many attempts to better understand protein dynamics by long unbiased MD. For example, Lindorff-Larsen et al.⁸ and Piana et al.⁹ simulated several proteins that undergo multiple folding events over the course of micro- to millisecond trajectories, yielding crucial insights into the hierarchy and timescales of the various structural rearrangements. With current technological limitations, unbiased MD is not capable of describing longer-timescale events, such as the dynamics of large proteins or the formation of multi-protein complexes. Due to the computational cost and timescales involved, there are just a few examples of modeling of such events, including folding of a dimeric protein Top7-CFr¹⁰ and all-atom computational reconstruction of protein-protein (Barnase-Barstar) recognition¹¹. Many methods have been developed to alleviate these sampling limitations, for instance, umbrella sampling¹², biased Monte Carlo methods¹³, and biased molecular dynamics like replica-exchange^{14,15}, steered MD^{16,17}, and metadynamics¹⁸. More recently, a new generative method based on normalizing flows has been proposed to sample structures from the Boltzmann distribution in one-shot, thereby avoiding the many steps needed in MD to sample different metastable states^{19,20}.

Another way to access the timescales of slow biological processes is through the use of coarse-graining (CG) approaches. Coarse-graining has a long history in the modeling of protein dynamics^{21,22} and since the pioneering work of Levitt and Warshel²³, many different approaches to CG have been proposed^{24–29}. Notably, the work by Hills et al.³⁰ has made significant strides towards creating a transferable bottom-up coarse-grained potential for the simulation of proteins, contributing valuable insights to the field. Popular CG approaches include structure-based models³¹, MARTINI^{32,33}, CABS³⁴, AWSEM³⁵, and Rosetta³⁶. In general, a CG model consists of two parts: the selection of the CG resolution (or mapping) and the design of an effective energy function for the model once the mapping has been assigned. Although recent work has attempted to combine these two points³⁷, they are in general kept distinct. The choice of an optimal mapping strategy is still an open research problem^{38–40} and we will assume in the following that the mapping is given, focusing instead on the second point, which is the choice of an energy function for the CG model that can reproduce relevant properties of the fine-grained system. Recently, our groups and others have used machine learning methods to extend the theoretical ideas of coarse-graining to systems of practical interest, which provides a systematic and general solution to reduce the degrees of freedom of a molecular system by building a potential of mean force over the coarse-grained system^{41–47}.

Machine learning models, in particular neural network potentials (NNPs), can learn fast, yet accurate, potential energy functions for use in MD simulations by training on large-scale databases obtained from more expensive approaches^{43,44,48–51}. One particularly interesting feature of machine learning potentials is that they can learn many-body atomic interactions⁵². A steady level of improvement of the methodology over the years has led to dozens of novel and better modeling architectures for predicting the energy of small molecules. The first important contributions are rooted in the seminal works by Behler and Parrinello⁵³ and Rupp et al.⁵⁴. One of the earliest transferable machine learning potentials for biomolecules, ANI-1⁵⁵, is based on Behler-Parrinello (BP) representation, while other models use more modern graph convolutions^{51,56,57}.

In this work, we investigate twelve non-trivial protein systems with a variety of secondary structural elements. We build a unique multi-millisecond dataset of unbiased all-atom MD simulations of studied proteins. We show the recovery of experimental conformations starting from disordered configurations through the classical Langevin simulations of a machine-learned CG force field. We demonstrate

transferability across macromolecular systems by using a single multi-protein machine learning potential for all the targets. Finally, we investigate the predictive capabilities of the NNP through simulation and analysis of selected mutants (i.e., sequences outside of the training set).

Results

Multi-millisecond all-atom molecular dynamics dataset

We created a large-scale dataset of all-atom MD simulations by selecting twelve fast-folding proteins, studied previously by Kubelka et al.⁵⁸ and Lindorff-Larsen et al.⁸ Supplementary Table 1. These proteins contain a variety of secondary structural elements, including α -helices and β -strands, as well as unique tertiary structures and various lengths from 10 to 80 amino acids. In the case of the shortest proteins, Chignolin and Trp-Cage (up to 20 amino acids), the secondary structure is quite simple. In general, the dataset contains a higher proportion of α -helical proteins. The exceptions are the β -turn present in Chignolin, the mostly β -sheet structure of WW-Domain, and the mixed $\alpha\beta$ structures of BBA, NTL9, and Protein G (Fig. 1). The dataset was generated by performing MD on each of the proteins starting from random coil conformations, simulating their whole dynamics and reaching the native structure. The total size of the dataset amounts to approximately 9 ms of simulation time across all proteins (Table 1). The dataset is available for download as a part of Supplementary Information.

Coarse-grained neural network potentials

A common approach to bottom-up coarse-graining is to seek thermodynamic consistency; i.e., the equilibrium distribution sampled by the CG model—and thus all thermodynamic quantities computable from it, such as folding free energies—should match those of the all-atom model³⁰. Popular approaches to train thermodynamically consistent CG models are relative entropy minimization⁵⁹ and variational force matching^{27,60,61}. The latter has recently been developed into a machine-learning approach to train NNPs to compute the CG energy^{43,44}.

Let \mathbb{D} be a dataset of M coordinate-force pairs obtained using an all-atom MD force field. Conformations are given by $\mathbf{r}_c \in \mathbb{R}^{3N_c}$, $c = 1, \dots, M$ and forces by $\mathbf{F}(\mathbf{r}_c) \in \mathbb{R}^{3N_c}$, where N_c is the number of atoms in the system. The number of atoms N_c depends on c as we wish to also have different protein systems in the dataset \mathbb{D} . We define a linear mapping Ξ which reduces the dimensionality of the atomistic system $\mathbf{x} = \Xi \mathbf{r} \in \mathbb{R}^{3n}$, where $3n$ are the remaining degrees of freedom. For example, Ξ could be a simple map to α -carbon atom coordinates for each amino acid, to backbone coordinates or to the center of mass. We seek to obtain $U(\mathbf{x}_c; \boldsymbol{\theta}) : \mathbb{R}^{3n} \rightarrow \mathbb{R}$ for any configuration c parameterized in $\boldsymbol{\theta}$, such that to minimize the loss

$$L(\mathbf{R}; \boldsymbol{\theta}) = \frac{1}{3nM} \sum_{c=1}^M \| \Xi \mathbf{F}(\mathbf{r}_c) + \nabla U(\Xi \mathbf{r}_c; \boldsymbol{\theta}) \|^2 \quad (1)$$

In order to reduce the conformational space accessible during the CG simulation and prevent the system from poor exploration, it is important to provide a prior potential^{44,62}. This also serves to reduce the complexity of the force field learning problem, and can equivalently be viewed as imposing physical biases from domain knowledge. The NNP is therefore performing a delta-learning between the all-atom forces and the prior forces. We applied bonded and repulsive terms to avoid rupture of the protein chain as well as clashing beads (Eqs. (11) and (12) in “Methods”). Furthermore, we enforce chirality by introducing a dihedral prior term (Eq. (13) in “Methods”). This prevents the CG proteins from exploring mirror images of the native structures. The functional forms and parameters of all prior terms are available in “Methods”.

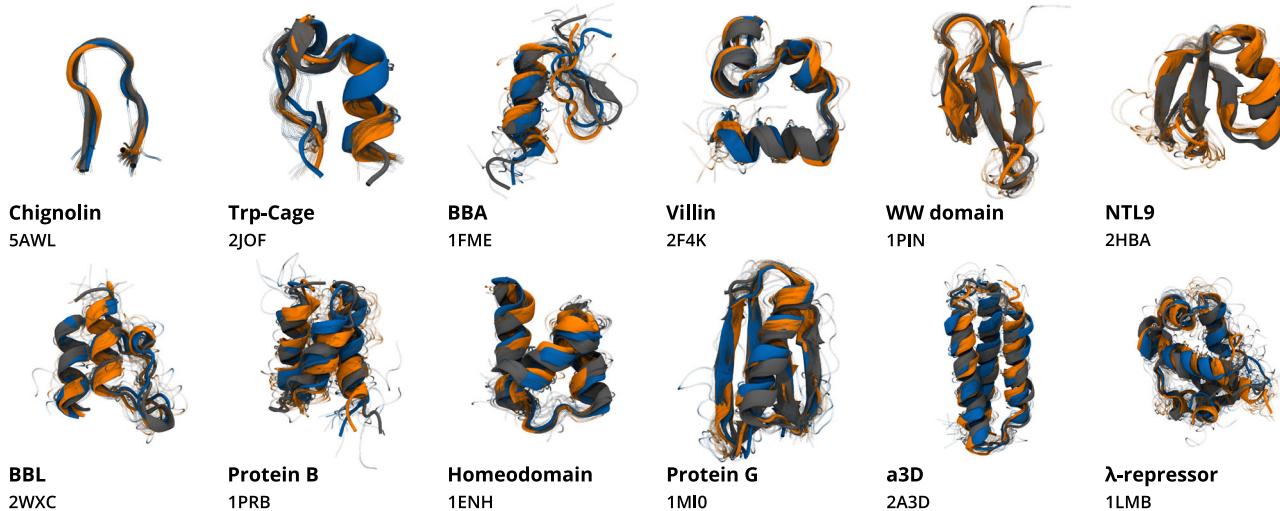


Fig. 1 | Comparison of simulated and experimental protein structures. Structures obtained from CG simulations of the protein-specific model (orange) and the multi-protein model (blue), compared to their respective experimental structures (gray). Structures were sampled from the native macrostate, which was identified as the macrostate containing the conformation with the minimum RMSD with respect to the experimental crystal structure. Ten conformations were sampled from each conformational state (visualized as transparent shadows) and the lowest RMSD

conformation of each macrostate is displayed in cartoon representation, reconstructing the backbone structure from α -carbon atoms. The native conformation of each protein, extracted from their corresponding crystal structure is shown in opaque gray. The text indicates the protein name and PDB ID for the experimental structure. WW-Domain and NTL9 results for the multi-protein model are not shown, as the model failed to recover the experimental structures. The statistics of native macrostates are included in Table 2.

CG representations were created by retaining only certain atoms of each protein's all-atom representation; the retained atoms are referred to as CG beads. NNPs were trained to predict forces based on the coordinates and identities of the beads, where the latter is represented as an embedding vector. Each CG bead comprises the α -carbon atom of its amino acid, and each amino acid was described by a unique bead type. In previous work, we experimented with both α -carbon and $\alpha\beta$ -carbon representation; however, the simpler α -carbon representation was sufficient to learn the dynamics of small proteins⁶³.

Coarse-grained molecular dynamics with neural network potentials reconstructs the dynamics of proteins

Initially, we carried out CG simulations of all twelve proteins using the models trained on individual all-atom MD datasets corresponding to each protein; that is, we trained twelve models, each one only using the corresponding data for one protein. To validate the models, we performed 32 parallel coarse-grained simulations for each target, starting

from conformations sampled across the reference free energy surface, built based on all-atom MD (Supplementary Fig. 1). The intent was to explore the conformational dynamics, sample the native structure and reconstruct the reference free energy surface.

A Markov state model (MSM)^{64–68} analysis of CG simulations shows that all of the individual protein models were able to recover the experimental structure of the corresponding target (Fig. 1), accurately predicting all the secondary structure elements and the tertiary structure, with loops and unstructured terminal regions being the most variable parts. For the simplest target, Chignolin, the average root-mean-square deviation (RMSD) value of the native macrostate was 0.7 Å. For less trivial structures, such as WW-Domain or NTL9, the values were below 2.5 Å. For even more complex arrangements of secondary elements, like Protein B and λ -repressor, the average RMSD of the native macrostate predicted by the network increased to 5.5 and 4.2 Å, respectively. In all cases, however, the network was able to sample conformations below 2.5 Å and global distance test (GDT)⁶⁹ scores above 60 (Table 2 and Supplementary Table 2).

For all protein-specific models, simulations were able to sample folding events, in which the protein goes from a random coil to a native conformation (Fig. 2 and Supplementary Fig. 2). The dynamics of transitions is accelerated more than three orders of magnitude, as the process happens in nanosecond timescale, in contrast to microseconds in the case of all-atom MD⁸. It is worth noting that, with current software, coarse-grained molecular dynamics with neural network potentials is 1–2 orders of magnitude slower than equivalent simulation with explicit solvent using classical force fields⁶³. However, we expect that this difference is going to reduce fast. In addition, individual trajectories were able to explore the conformational landscape and transition between different metastable states observed in the original all-atom trajectories. For each protein, a representative trajectory is shown in a video included in Supplementary Information (Supplementary Table 3). A few models, in particular Homeodomain, α 3D, and λ -repressor, failed to sample direct transitions from ordered to disordered conformations (Supplementary Fig. 2). This could have been caused partially by the model over-stabilizing the native structure.

Table 1 | All-atom MD simulation dataset generated for this work and used for training and testing of NNPs

Protein	Sequence length (#aa)	Aggregated time (μs)	Min. RMSD (Å)
Chignolin	10	186	0.15
Trp-Cage	20	195	0.45
BBA	28	362	1.13
WW-Domain	34	1362	0.73
Villin	35	234	0.47
NTL9	39	776	0.32
BBL	47	677	1.55
Protein B	47	608	1.19
Homeodomain	54	198	0.56
Protein G	56	2266	0.55
α 3D	73	768	1.81
λ -repressor	80	1422	0.82

Table 2 | Native macrostate statistics from all MSMs built with CG simulations from all protein-specific models and the multi-protein model

Protein	Protein-specific			Multi-protein			Reference		
	Macro prob. (%)	Mean RMSD (Å)	Min RMSD (Å)	Macro prob. (%)	Mean RMSD (Å)	Min RMSD (Å)	Macro prob. (%)	Mean RMSD (Å)	Min RMSD (Å)
Chignolin	19.7 ± 0.8	0.7 ± 0.4	0.2	33.4 ± 0.6	1.2 ± 0.6	0.2	57.5 ± 0.6	1.0 ± 0.4	0.1
Trp-Cage	93.2 ± 0.7	2.8 ± 0.5	1.0	81.1 ± 12.0	2.9 ± 0.5	1.0	30.1 ± 3.9	2.5 ± 0.8	0.4
BBA	41.1 ± 1.8	3.8 ± 1.0	1.6	17.5 ± 1.4	4.4 ± 1.0	1.6	5.24 ± 0.9	3.9 ± 1.3	1.1
WW-Domain	15.4 ± 2.5	2.5 ± 0.5	1.1	—	—	—	45.5 ± 1.1	2.7 ± 1.1	0.7
Villin	77.3 ± 8.9	2.7 ± 0.9	0.8	77.7 ± 13.0	2.9 ± 0.9	1.0	69.2 ± 1.4	3.4 ± 1.8	0.5
NTL9	32.0 ± 2.2	2.4 ± 0.9	0.6	—	—	—	15.3 ± 3.5	1.6 ± 0.9	0.3
BBL	95.0 ± 0.5	2.8 ± 1.2	1.0	47.8 ± 8.3	2.4 ± 0.6	0.9	30.5 ± 2.7	3.1 ± 1.3	0.7
Protein B	71.6 ± 1.6	5.6 ± 1.0	2.3	75.8 ± 6.4	3.3 ± 0.5	2.0	30.1 ± 0.4	4.4 ± 1.4	1.2
Homeodomain	77.6 ± 14.0	2.8 ± 0.4	1.8	98.5 ± 0.4	2.4 ± 0.3	1.5	53.5 ± 1.9	2.3 ± 1.5	0.3
Protein G	64.8 ± 3.9	2.7 ± 0.5	1.4	2.1 ± 0.9	2.2 ± 0.4	1.2	17.1 ± 1.6	2.9 ± 1.9	0.6
α3D	90.5 ± 6.9	3.2 ± 0.2	2.4	96.4 ± 2.4	3.4 ± 0.3	2.2	67.9 ± 1.2	3.5 ± 0.7	1.8
λ-repressor	77.4 ± 10.7	4.3 ± 0.5	2.1	79.1 ± 7.0	4.6 ± 0.7	2.8	21.9 ± 0.5	4.5 ± 1.2	0.8

The data describes the identified native macrostate for each protein, showing equilibrium probabilities in percentage (Macro prob.), average (with standard deviation), and minimum RMSD values with respect to the experimental structure.

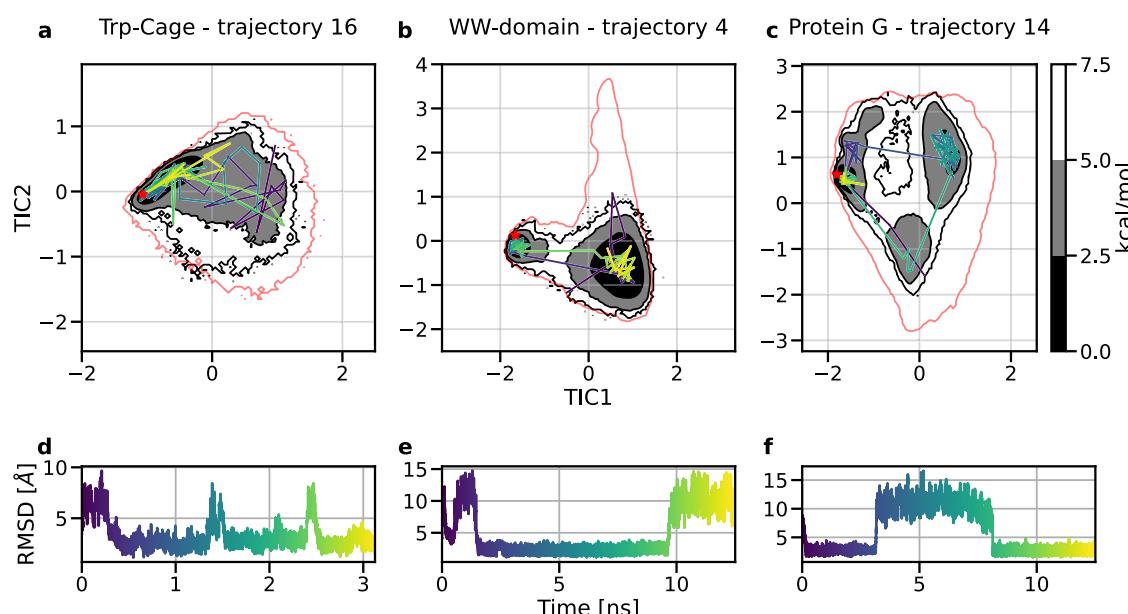


Fig. 2 | Trajectory analysis of protein dynamics. Three individual CG trajectories selected from validation MD of Trp-Cage, WW-Domain, and Protein G. Each visualization simulation, colored from purple to yellow, explores the free energy surface, accesses multiple major basins and transitions among conformations. Top panels: 100 states sampled uniformly from the trajectory plotted over CG free energy surface, projected over the first two time-lagged independent components

(TICs) for Trp-Cage (a), WW-Domain (b), and Protein G (c). The red line indicates the all-atom equilibrium density by showing the energy level above the free energy minimum with the value of 7.5 kcal/mol. The experimental structure is marked as a red star. Bottom panels: Ca-RMSD of the trajectory with reference to the experimental structure for Trp-Cage (d), WW-Domain (e), and Protein G (f). Source data are provided as a Source data file.

Coarse-grained potentials maintain the energetic landscape

In order to estimate the equilibrium distribution and approximate the free energy surfaces from the CG simulations, we built MSMs for each CG simulation set. Time-lagged independent component analysis (TICA)^{70,71} was used to project coarse-grained trajectories onto the first three components, using covariances computed from reference all-atom MD. Overall, the MSMs were able to recover the surface describing the dynamics, correctly locating the position of the global minimum in the free energy surface for all cases except Protein B (Fig. 3 and Supplementary Figs. 3 and 4). The most ill-defined regions of TIC space correspond to unstructured conformations, which are more difficult for the models to sample. In most of the models, simulations transition rapidly to the native structure, and only the surface around

the global minimum is sampled. This is particularly true for larger helical proteins, such as Homeodomain, α3D, and λ-repressor, where the space explored falls mostly around the native structure. Alternatively, in Chignolin, Trp-Cage, Villin, NTL9, and Protein G, the models are able to sample most of the free energy surface, locating all different metastable minima identified through TICA.

In the case of Protein G, the model was able to identify all the metastable states, sharing similar features as the reference all-atom MD simulations (Fig. 4). Furthermore, the model correctly replicates the main transition to the native structure and allows for a possible interpretation of the folding pathway. In the most probable folding pathway, the protein initially forms an intermediate, partially folded state containing the α-helix and the first hairpin. Next, the native

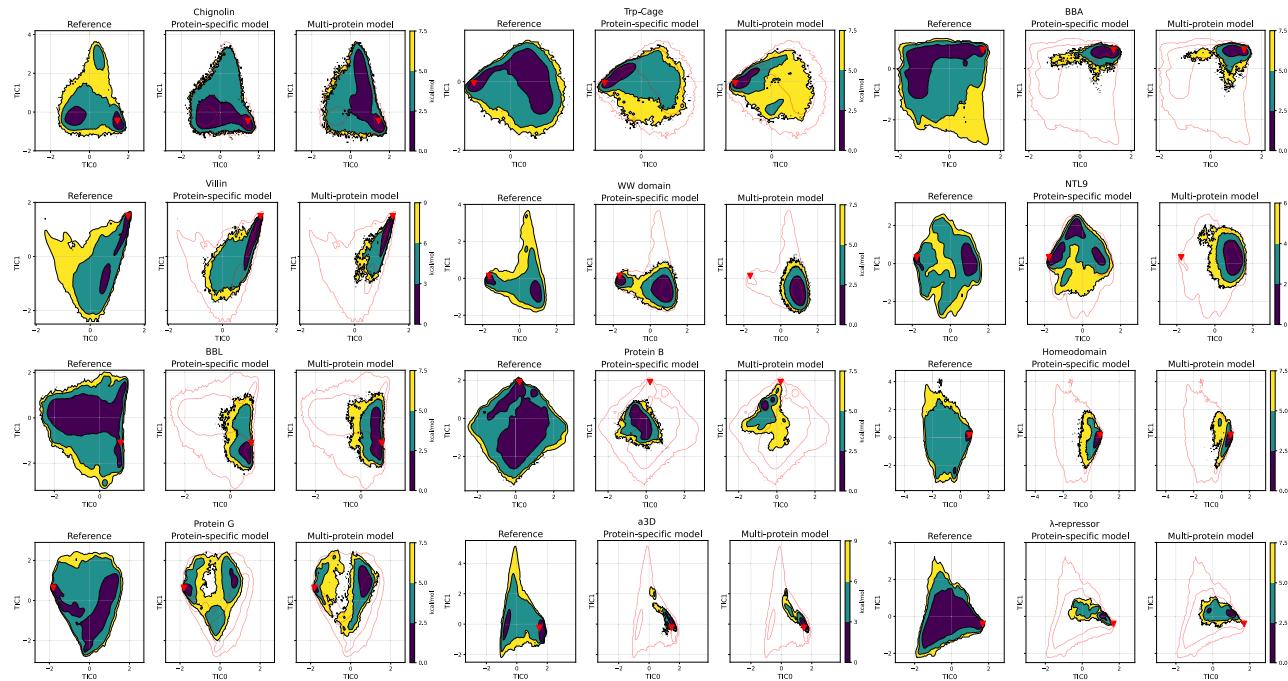


Fig. 3 | Free energy surface comparison across all-atom reference and coarse-grained models. Comparison between the reference MD (left), protein-specific model (center), and multi-protein model (right) coarse-grained simulations free energy surface across the first two TICA dimensions for each protein. The free energy surface for each simulation set was obtained by binning over the first two TICA dimensions, dividing them into a 80×80 grid, and averaging the weights of

the equilibrium probability in each bin computed by the Markov state model. The red triangles indicate the experimental structures. The red line indicates the all-atom equilibrium density by showing the energy level above free energy minimum with the values of 9 kcal/mol for Villin and α 3D, 6 kcal/mol for NTL9, and 7.5 kcal/mol for the remaining proteins. Source data are provided as a Source data file.

structure is completed by the formation of the second hairpin. Alternatively, a second pathway is possible where the structure goes through a misfolded state with an almost complete native structure except for the first hairpin, which shows increased flexibility. This replicates the results of all-atomistic MD simulations performed by Lindorff-Larsen et al.⁸. The variant simulated both there and in this study is intermediate in sequence between the wild type and redesigned NuG2 variant. Despite high similarities in the sequence, experiments show that these variants exhibit distinct folding pathways. The difference is in the order of formation of the elements of β -sheet; in the wild-type variant of Protein G, the second hairpin folds before the first hairpin^{72,73} while in the NuG2 variant the order is reversed⁷⁴. The CG simulation using NNP shows the majority of flow going into the NuG2 variant folding, which agrees with one of the possible folding pathways. In addition, the simulation correctly recovered the minima around the native conformation of Protein G, however, the position of the other minima on the free energy surface are less similar. In general, the force-matching method does not preserve kinetics^{27,60}, so the height of the energy barriers is not expected to be accurately captured, as shown in the free energy plot (Supplementary Figs. 3 and 4).

For NTL9, the model correctly replicates the transition to the native structure, allowing for a possible interpretation of its pathway (Supplementary Fig. 5). From the structural samples, we can see that the α -helix is the first secondary element formed that appears even in the unstructured macrostate. By identifying the intermediate state, where the β -sheet is not entirely formed, we can understand that β -sheet formation is the limiting step in the process.

The multi-protein model recovers the native structures of most reference proteins

The individual CG models recovered native structures of the proteins, demonstrating the success of our approach for complex structures.

These NNPs are, however, limited to the individual targets they were trained on. In the next step, we examined if it was possible to train a single, multi-protein model using the reference simulation data of all the protein targets (Supplementary Fig. 6). We then simulated all targets with the multi-protein model, in the same way we did for the protein-specific models. The main objective of the multi-protein model is to match the results of individual models using a single CG potential.

The CG simulations show that the multi-protein model is able to reproduce the native structure of most of the proteins, with the exception of NTL9 and WW-Domain (Fig. 1 and Supplementary Figs. 3 and 4). We identify each native macrostate based on its RMSD to the corresponding experimental structure. However, a simple criterion of minimal potential energy produced by the NNP is able to correctly identify all of the native macrostates for protein-specific models described in the previous section, and in nine out of ten cases (excluding NTL9 and WW-Domain) where the multi-protein model sampled the native structure. The only exception is BBA, where a quasi-folded macrostate is selected instead which has not fully stabilized the small β -sheet (Supplementary Fig. 7).

In general, the free energy landscapes produced by the multi-protein model resemble the protein-specific ones. However, the multi-protein model neglects energetic barriers and overestimates the global minima, which leads to some trajectories being stuck at the native structure (Fig. 3 and Supplementary Figs. 3, 4 and 8).

In the cases of NTL9 and WW-Domain, the native structure is sampled only as an artefact of starting positions being equally distributed on the reference free energy surface (Supplementary Fig. 1). The native structure is not stable as all simulations move quickly to unstructured conformations. For Protein G, simulations show that the native conformation is stable, but we could not sample any transitions into this conformation from random coil initial conditions, although we could capture unfolding events (Supplementary Fig. 9). In these

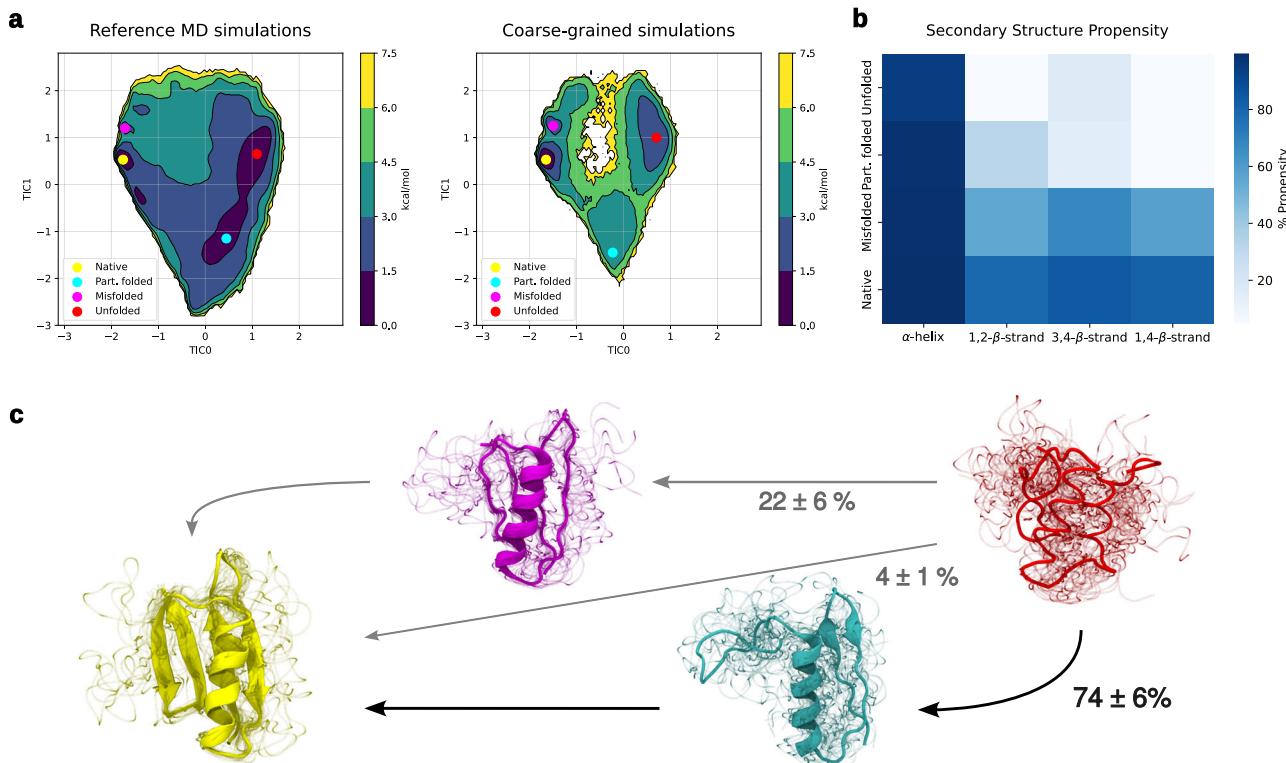


Fig. 4 | Free energy surface and structural analysis of Protein G simulations. **a** Free energy surface of Protein G over the first two TICs for the all-atom MD simulations (top) and the coarse-grained simulations (bottom) using the protein-specific model. The circles identify different relevant minima (yellow—native, magenta—misfolded, cyan—partially folded, red—random coil). **b** The propensity of all the secondary structural elements of Protein G across the different macrostates, estimated using an RMSD threshold of 2 Å for each structural element shown in the

x-axis. **c** Sampled conformations from the macrostates of coarse-grained simulations corresponding to the marked minima in the free energy surfaces in (a). Sampled structure colors correspond to the minima colors in the free energy surface plot, with blurry lines of the same color showing additional conformations from the same state. Arrows represent the main pathways leading from the random coil to the native structure with the corresponding percentages of the total flux of each pathway. Source data are provided as a Source data file.

cases, the native structure is identified as the lowest energy structure by the NNP. Therefore we can promote transitions to the low-energy states by lowering the temperature of the simulation. We simulated these systems at temperatures of 300 K and 250 K. This approach showed that the NNP recovers the native structure of NTL9 at 300 K. For Protein G and WW-Domain, lowering the temperature stabilizes conformations that resemble the experimental structures, but we have not observed transitions from fully disordered to ordered structures (Supplementary Fig. 10).

One aspect that the failed cases have in common is the presence of β -sheets, which could be the reason why the multi-protein models make the proteins' structured states unstable. Ten out of twelve proteins in the training set contain α -helices, with only Chignolin and WW-domain representing completely β -sheet proteins and BBA, NTL9 and Protein G containing a mix of secondary structure elements. Therefore, the multi-protein model might be biased towards helical structures. Another explanation could be that due to the locality of interactions α -helices may be easier to learn for the NNP. α -helices can be formed gradually with smaller energy barriers, while β -sheets arrange a full strand at a time. In addition, α -helices are stabilized by residues close in the sequence, which provides molecular context even in the random coil state. On the contrary, the stabilizing interactions of β -sheets occur between the residues distant in the sequence. Therefore, for random conformations, the beads are usually outside of the 12 Å upper cutoff of the NNP, reducing the number of examples to learn from. Extending the upper cutoff leads, however, to noisy potentials and an overall worse performance. Similar difficulties with β -sheet proteins were observed during hyperparameter optimization of single-protein NNPs.

For all the helical proteins, the multi-protein model performs similarly to the protein-specific models (Table 2). In some cases, the frequency of transitions between states is altered, as well as the stability of the macrostates, but both models successfully recover the native conformations.

In the case of Trp-Cage, the multi-protein potential outperforms the protein-specific model. The location and the shape of the global minimum match better the reference simulations as well as experimental data, which indicates that the model benefits from additional data from other proteins (Fig. 3 and Supplementary Figs. 3, 4 and 8). In the case of Protein B, the multi-protein model also outperforms the protein-specific one, as it is able to improve the average RMSD of the native macrostate and samples the correct location of the experimental structure, although it is not detected as a minimum (Table 2, Fig. 3).

The results obtained with the multi-protein model are in line with protein-specific models, which indicates that our approach could scale to create a general-use CG force field. This model was able to simulate the transition from random coil to the correct native conformation for almost all target proteins, with the exception of β -sheet proteins (WW-Domain, NTL9, and Protein G), which required simulations at lower temperatures to recover the native state.

The multi-protein NNP recovers the native structure of mutated proteins

To further test the multi-protein NNP and assess its predictive power we simulated mutants of the originally targeted proteins. All mutants were sourced from PDB and the mutations did not affect the native structure of the target. Supplementary Table 4 summarizes the

Table 3 | Native macrostate statistics of mutant variants of the proteins based on the CG simulations performed with the multi-protein model

Protein	PDB	Number of substitutions	Min RMSD (Å)	Mean RMSD (Å)	Eq. prob. (%)
BBL	1BAL	3	1.5	3.9±0.9	52.3±1.4
Protein B	1GAB	2	2.3	4.9±1.2	32.5±1.4
Protein B	2N35	10	4.0	9.3±1.4	19.4±0.8
Homeodomain	1DUO	1	1.6	2.6±0.4	57.0±3.2
Homeodomain	1P7I	1	1.4	2.5±0.3	92.2±13.3
Homeodomain	1P7J	1	3.8	4.6±0.3	16.6±6.6
Homeodomain	2HOS	4	1.6	3.1±0.8	65.4±5.7
Homeodomain	6M3D	2	1.5	2.5±0.3	24.5±4.0
α3D	2MTQ	3	2.8	4.4±0.6	96.2±0.9
λ-repressor	1LLI	3	3.1	5.1±0.9	97.0±0.8
λ-repressor	3KZ3	6	2.3	5.8±1.4	76.1±7.6

The table shows the protein name, PDB ID and the number of amino acid substitutions for each mutant. Results show the minimum RMSD with respect to the mutant experimental structure, as well as the mean RMSD and equilibrium probability of the native macrostate, obtained from an MSM built based on CG simulations of the mutant.

structures selected for the experiment. For each mutant, we initially performed a CG simulation of a single trajectory that started from the native structure using the multi-protein model. In the majority of cases, the structure immediately transitioned to a random coil. The mutants that kept the native conformation for 1 ns were further evaluated using the same protocol we used for the previous CG simulations.

The results show that the multi-protein CG model is able to recover the native conformation for all cases that succeeded in the initial validation, except one (Protein B mutant 2N35), with reasonably low RMSD values (Table 3, Supplementary Fig. 11). Although the NNP was able to simulate the protein dynamics, the exploration of conformational space was limited, as the simulations converge rapidly to the native structure or the conformations resembling it (Supplementary Fig. 12). These cases demonstrate some ability of the multi-protein model to generalize outside of the training set even with a narrow training set of only twelve proteins.

All the examples that recovered the native conformation had very few mutations and were solely helical structures, for which the multi-protein model performs well. In the case of a mutant of Protein B (PDB: 2N35), the NNP failed to obtain the native structure. Its sequence contains 10 mutated residues, which may exceed the capacity of the model to generalize outside of the training set. As shown in Supplementary Table 4, an increased number of mutations reduces the stability of the native macrostate. In the case of β -sheet containing proteins, even with a point mutation, the model failed to recover the native structure and the amino-acid chains immediately formed unstructured bundles. This observation is not surprising, given the difficulties encountered by the multi-protein model on the β -sheet containing targets.

Overall, the mutagenesis tests have shown limited but encouraging results for the predictive capabilities of the multi-protein model. Despite its failure to keep the native conformation stable for the sequences that are substantially altered or for proteins that contain β -sheets, the NNP recovered native macrostates of α -helical proteins with minor changes in the sequence. This shows some capacity of the model to generalize.

Discussion

In the previous work, we have shown that an NNP with a non-transferable coarse-grained model architecture can learn the

thermodynamics of a single protein⁴⁴. Then, in the following publication, we replicated the task using a model architecture that is in principle transferable⁴³. In this work, we apply a revised model architecture and show that we can effectively learn thermodynamics for twelve structurally diverse proteins at once, in a single model. This demonstrates for the first time that the model architecture is truly transferable and might generalize providing enough data. To achieve that, we generated a multi-millisecond dataset of MD simulations sampling the dynamical landscape of the proteins and used it to obtain machine-learned CG potentials for studying the protein dynamics. Results show that we were able to model protein dynamics in computationally accessible timescales, and recover the native structure of all twelve proteins through coarse-grained MD simulations using NNPs and an α -carbon CG representation, with a unique bead type corresponding to each amino-acid type. From the model-generated CG simulation data, we were able to reconstruct multiple metastable states, capturing the folding pathways and the formation of different types of secondary and tertiary structures. In contrast to novel deep-learning structure prediction methods^{75,76}, our method offers a substantial improvement and models protein dynamics, which is essential for understanding protein function.

The multi-protein model, trained over all proteins in the MD dataset, demonstrates that we were able to model multiple proteins with a single NNP. The following tests on mutants of the 12 proteins have shown the robustness of the multi-protein NNP to small differences in sequence. We highlight, however, that the current training set, while being one of the largest ever produced, only contains data for 12 small proteins. With such a small number of training examples, it is unrealistic to expect that the NNP will model sequences different from the training set. Therefore we do not provide a hold-out test set. While it is not a physical model, this work is a fundamental step in that direction.

There are a few limitations to the current approach. In general, machine learning potentials do not extrapolate well outside of the training set for atom positions that are never sampled in the training set. Therefore, unseen positions are assigned unrealistically low energies and often produce spikes in forces. This has been solved by limiting the physical sampled space with the use of basic prior energy terms⁴⁴. The network also relies on large datasets of all-atom molecular dynamics trajectories which are expensive to produce. Furthermore, the current accuracy of coarse-grained MD is limited by the accuracy of the underlying all-atom simulations. While all-atom force fields are reasonably good for proteins, improved approaches are required for coarse-grained small molecules⁷⁷. Ultimately, the ability to create a truly general model that is transferable from smaller to larger proteins would revolutionize the field⁷⁸. Some works suggest that transferability can be achieved by a sufficient sampling of various configurations and state variables⁷⁹. We think that, in order to learn transferable potentials, some key improvements need to be made, mainly: much larger molecular simulation datasets, alternative training strategies, improved coarse-grained mapping strategies, and more robust architectures that can deal with non-physical states. Current results indicate that this might be achievable.

Methods

All-atom molecular dynamics simulations and training data

All initial structures were solvated in a cubic box and ionized as described by Lindorff-Larsen et al.⁸. MD simulations were performed with ACEMD⁸⁰ on the GPUGRID.net distributed computing network⁸¹. The systems were simulated using the CHARMM22⁸² force field and TIP3P water model⁸³ at the temperature of 350 K. All the simulations were performed following a previously used adaptive sampling strategy⁸⁴, in order to explore efficiently as many conformations as possible. Homeodomain dataset also contains simulations that started from the native conformation, as low RMSD values (≤ 2 Å) with respect

to the native structure are difficult to sample when starting from random coil conformations. A Langevin integrator was used with a damping constant of 0.1 ps^{-1} . Integration time step was set to 4 fs, with heavy hydrogen atoms (scaled up to four times the hydrogen mass) and holonomic constraints on all hydrogen-heavy atom bond terms⁸⁵. Electrostatics were computed using Particle Mesh Ewald with a cutoff distance of 9 Å and grid spacing of 1 Å. Ten NVT simulations of 1 to 10 ns length were carried out for each protein, with a dielectric constant of 80 and temperature of 500 K to generate ten different starting random coil conformations for the production runs. Production simulations consisted of thousands of short trajectories of 20, 50, or 100 ns, distributed across different epochs using the adaptive sampling^{84,86} protocols implemented in HTMD⁸⁷. In adaptive sampling, multiple rounds of simulations are performed, and in each round the available trajectories are analyzed to select the initial coordinates for the next round of simulations. The MSM constructed during the analysis was done using atom distances, using TICA for dimensionality reduction and k-centers for clustering. From the trajectories, we extracted forces and coordinates with an interval of 100 ps. Total aggregate times used for training for all the proteins are summarized in Table 1.

Based on the MD dataset we built MSMs for each protein. The models were able to describe the conformational dynamics of each protein, sample the native conformation and identify intermediate and metastable states for some of them, such as Villin, NTL9, WW-Domain, or Protein G (Fig. 3).

Neural network training

To train NNPs we used TorchMD-Net⁷⁷. We performed an exhaustive hyperparameter search, which is described in Supplementary Table 5. The total number of parameters of the network is 294,565. The data was randomly split between training (85%), validation (5%), and testing (10%). An epoch for simulation was selected when the validation loss reached a minimum or a plateau. The training and validation loss reported as MSE loss, test loss reported as L1 loss and the learning rates of models selected for simulation are presented in Supplementary Figs. 6 and 13. The models were trained using Nvidia GeForce RTX 2080 graphics cards. The training of protein-specific models took from 7 min/epoch on a single GPU for Chignolin to 24 min/epoch on 2 GPUs for λ -repressor. The training of the multi-protein model took 46 min/epoch on 3 GPUs.

A lot of effort was dedicated to building a graph neural network architecture TorchMD-GN, inspired by SchNet^{56,88} and PhysNet⁵¹ and optimized to work optimally on noisy forces and energies proper of the reduced dimensionality of our coarse-graining. This scenario is different from the quantum case, where energy and forces are deterministic functions of the coordinates. In coarse-grained systems, the same coordinates generate stochastic energies and forces. The software was implemented using PyTorch Geometric⁸⁹ and PyTorch lightning framework⁹⁰ and is publicly available in TorchMD-Net⁷⁷. The SchNet architecture has several distinct components, each playing an important function in predicting system forces and energies for given input configurations. The formal inputs into the network are the Cartesian coordinates for a full configuration and a predetermined type for each coarse-grain bead. In the first network operation, a molecular graph, \mathcal{G} , is constructed, where each coarse grain bead represents a node. Each node is given an embedding feature vector, the set of which is grouped into a feature tensor. For SchNet, the embedding is produced by applying a learnable linear mapping. The edges of \mathcal{G} are used to define the network operations that update the features of each node. These updates are encompassed in so-called “interaction blocks”, which are a form of message-passing updates. The edges of \mathcal{G} are the set of pairwise distances for each bead from its nearest neighbors, the range of which is set uniformly for all beads by an upper cutoff distance. In this way, several interaction blocks can be stacked in

succession to give the network increased expressive power. After the final interaction block, an output network is used to contact the node feature dimension to a scalar for each node. This forms a set of scalar energy predictions, U from each node. By applying a gradient operation with respect to the network input coordinates, the curl-free Cartesian forces, \mathbf{F} , are predicted for each bead, representing the final network output.

The hyperparameters were selected based on the quality of the simulation produced using protein-specific models. An example of a training input file is presented in Supplementary Listing 1. The test loss was not a useful metric for hyperparameter selection because the value did not change much between successful and failed models^{91,92}. The only way to correctly validate the models was to use them in coarse-grained simulations. A high-quality model produces stable MD simulations, the trajectories explore the conformation landscape and the free energy surface is smooth. In addition, a good model will match the results of all-atom simulations and form energy minima around the relevant states, and we will observe multiple transitions between these states. The hyperparameter combination had the biggest influence on the stability of the MD simulation, the smoothness of the free energy landscape, and the visited areas of the conformational landscape. We found that reducing the number of radial base functions from 150, as in the previous work⁶³, to around 18 has a big impact on the stability of the MD simulations of proteins bigger than chignolin. With a higher number of RBFs, the forces become spiky for the conformations that are not present in the training set, which leads to the instability of MD runs. Further improvement can be made by replacing the Gaussian function that was used in previous works with expnorm. It is slightly elongated towards longer distances and this shape better suits modeling the properties of CG beads. The smoothness of the landscape was affected the most by the type of activation function. Hyperbolic tangent (tanh) makes the free energy surface smooth, while shifted soft-plus (ssp) caused the trajectories to collapse into many local minima, making the surface grainy. Other parameters that have a significant influence on the quality of the models are the number of interaction layers and the range of radial base functions. It is important to mention that in some cases even the random seed has an influence on the quality of the models, especially on the coverage of the free energy landscape. For that reason, to ensure reproducibility of the results we trained 2–4 replicas of each model, as mentioned in the main text. Based on the results for protein-specific models as well as the multi-protein model, we selected the following combination: 4 interaction layers, 128 filters used in continuous-filter convolution, 128 features to describe atomic environments, and 18 expnorm as radial base functions (RBF) span in the range from 3.0 to 12.0 Å. In general, we found that the models with hyperparameters similar to these tend to be good quality in terms of the metrics mentioned before.

Neural network architecture

The series of full network operations can be written as:

$$\xi^0 = \mathbf{W}^F \mathbf{z} \quad (2)$$

$$\xi^1 = \xi^0 + \mathbf{W}^0 \sigma(\text{Aggr}(\mathbf{W}^C * \xi^0)) \quad (3)$$

$$\xi^2 = \xi^1 + \mathbf{W}^1 \sigma(\text{Aggr}(\mathbf{W}^C * \xi^1)) \quad (4)$$

$$\vdots \quad (5)$$

$$\xi^N = \xi^{N-1} + \mathbf{W}^{N-1} \sigma(\text{Aggr}(\mathbf{W}^C * \xi^{N-1})) \quad (6)$$

$$U = H_{\text{out}}(\mathbf{\xi}^N) \quad (7)$$

$$\mathbf{F} = -\text{grad}(U, x) \quad (8)$$

for N interaction blocks. Note that for clarity, we have omitted learnable additive biases in all linear operations above, though they are easily incorporated. The first step of the message-passing update involves expanding the pairwise distances into a set of radial basis functions, ϕ . ϕ then comprises a filter generating network used to produce a set of continuous filters, \mathbf{W}^C :

$$\mathbf{W}^C = \mathbf{W}2(\sigma(\mathbf{W}1\phi)) \quad (9)$$

where $\mathbf{W}_1, \mathbf{W}_2$ are learnable linear weights and σ is an element-wise non-linearity. These filters are used in a continuous filter convolution through an element-wise multiplication with the current node features for \mathcal{G} . These convolved features are then passed through a non-linearity and added directly to the unconvolved node features through a residual connection:

$$\mathbf{\xi}^{i+1} = \mathbf{\xi}^i + \mathbf{W}^i \sigma(\text{Aggr}(\mathbf{W}^C * \mathbf{\xi}^i)) \quad (10)$$

where “Aggr” is a chosen pooling/aggregation function that reduces the convolution output (eg, sum, mean, max, etc.). This message-passing update, combined with the residual connection, forms the entirety of an interaction block, producing an updated set of node features for \mathcal{G} that can be used as input for another interaction block. Our implementation of this network architecture allows for training on multiple GPUs and more efficient utilization of GPU memory.

Coarse-grained simulations

Coarse-grained representations were created by filtering all-atom coordinates such that only certain atoms are retained. This mapping is a simple linear selection, wherein the mapping matrix that transforms the all-atom coordinates to the coarse-grained coordinates is a matrix where zero-entries filter out unwanted beads. The all-atom trajectories were filtered to retain the coordinates and forces of α -carbon atoms (CA). To speed up the training of protein-specific modes, trajectories were further reduced by selecting every 10th frame. However, for smaller proteins (Chignolin, Trp-Cage, BBA, and Villin), the training data was not sufficient to produce satisfactory models. Therefore all the frames were used in the training of these systems. To train the multi-protein model we combined the datasets for all the targets. Each CA bead was assigned a bead type based on the amino acid type. In the assignment, we ignored the protonation states and distinguished norleucine, a non-standard residue appearing in Villin, as a unique entity. The terminal residues were assigned the same embedding as the non-terminal residues, despite the charge. As a result, we obtained 21 unique bead types. To each bead type we assigned a unique integer, an embedding that will be used as an input for the network.

To perform the coarse-grained simulations using a trained NNP, we used TorchMD⁶³, an MD simulation code written entirely in PyTorch⁹³. The package allows for an easy simulation with a mix of classical force terms and NNPs. The parameters for the prior energy terms were enumerated and stored in YAML files. The NNP was introduced as an external force, as described in the previous work⁶³. We carried out CG simulations over all the proteins, both for each protein-specific model and for the multi-protein model, as well as selected mutants. Simulations were set up with a configuration file (an example in Supplementary Listing 2). We selected 32 conformations evenly distributed across the free energy surface of the reference simulations from where to start the coarse-grained simulations (Supplementary Fig. 1). For each system, 32 parallel, isolated trajectories were run at 350 K for the time necessary to observe transitions

between states with a 1 fs time step, saving the output every 100 fs. The length of each individual trajectory was 1.56 ns (accumulated time of 50 ns) for Chignolin and BBA, 3.12 ns (accumulated time of 100 ns) for Trp-Cage and Villin, 12.5 ns (accumulated time of 400 ns) for WW-Domain and Protein G, and 6.25 ns (accumulated time of 200 ns) for the remaining protein targets. For some systems and models, we were able to obtain stable trajectories with a time step as high as 10 fs. However, to make the results comparable we adapted identical parameters for all simulations, and thus we were limited by the highest possible time step where all types of simulations were stable (1 fs). We observed that for the conformations not represented in the training set, the forces tend to form spikes, which leads to the instability of the simulations. This can be reduced by applying prior force terms and applying cutoffs to radial base functions that limit the exploration of unphysical conformations. In addition, a reduced number of radial base functions has a positive impact on the overall smoothness of the force field. The coarse-grained simulations were performed using Nvidia GeForce RTX 2080 graphics cards.

Prior energy terms

The pairwise bonded term was represented with the following equation:

$$V_{\text{bonded}}(r) = k(r - r_0)^2 + V_0, \quad (11)$$

where r is the distance between the beads forming the bond, r_0 is the equilibrium distance, k is the spring constant and V_0 is a base potential. The nonbonded repulsive term was represented by the potential

$$V_{\text{repulsive}}(r) = 4\epsilon r^{-6} + V_0, \quad (12)$$

where ϵ is a constant that was fit to the data, r is the distance between the beads and V_0 is a base potential. The parameters were used as in TorchMD⁶³. The parameters for norleucine, a non-standard residue appearing in Villin, were adapted from leucine. In addition, we introduced a third prior dihedral term:

$$V_{\text{dihedral}}(\phi) = \sum_{n=1,2} k_n (1 + \cos(n\phi - \gamma_n)), \quad (13)$$

where ϕ is the dihedral angle between the four consecutive beads, k_n is the amplitude and γ_n is the phase offset of the harmonic component of periodicity n . The parameters for dihedral terms were fit to the data used for training, containing all the proteins. The extracted values of k_n were scaled by half to achieve a soft prior that will break the symmetry in the system but will not disturb the simulation in a major way. For simplicity, all combinations of four beads were treated equally, therefore all dihedral angles were characterized by the same set of parameters, in contrast to bonded and repulsive prior. The force field file with terms and associated parameters is available in the GitHub repository. To enable the simultaneous use of both Dihedral and RepulsionCG force terms in TorchMD, exclusions between pairs of beads for RepulsionCG term are defined by an additional parameter “exclusions”.

Markov state model estimation and structure selection

For the analysis of the CG simulations and their comparison with the all-atom MD simulations, we built MSMs for each protein, both for the all-atom MD simulations and the two sets of coarse-grained simulations (protein-specific and multi-protein models). The basic concept behind MSMs is that the dynamics of the system are modeled as a memory-less jump process, where future states are only conditioned on the current state, hence the dynamics are Markovian. MSM estimation of transition rates and probabilities requires partitioning the high-dimensional conformational space into discrete

states. In order to project the high-dimensional conformational space into an optimal low-dimensional space, we use TICA, a linear transformation method that projects simulation data into its slowest components by maximizing autocorrelation of transformed coordinates at a given lag time^{70,71}. The resulting low-dimensional projected space is then discretized using a clustering algorithm for the MSM construction.

For the all-atom MD simulations, we featurized the simulation data into pairwise C_α distances and applied TICA to project the featurized data into the first 4 components. Next, the components were clustered using a K-means algorithm and the discretized data was used to perform the MSM estimation. Although better reference MSM models could be obtained by using different featurizations, we are limited to only using pairwise C_α distances as it is transferable between systems and comparable with the coarse-grained simulations.

For the coarse-grained simulations, the same procedure was used. However, when projecting the featurized data into the main TICs, we used the covariance matrices computed with the all-atom MD simulations to project the first 3 components, in order to compare how well the coarse-grained simulations reproduce the free energy surface for each protein. For each MSM, we used the PCCA algorithm to cluster microstates into macrostates for better interpretability of the model and to define a native macrostate that we can use to evaluate the performance of the coarse-grained simulations.

The free energy surface plots used for comparison were obtained by binning over the first two TICA components, dividing them into an 80×80 grid, and averaging the weights of the equilibrium probability in each bin, obtained for each defined microstate through MSM analysis. To recover the native conformation from a set of coarse-graining simulations, we used the MSMs and sampled conformations from the native macrostate. The native macrostate was defined as the macrostate containing the frame with the minimum RMSD to the experimental structure.

Statistics and reproducibility

To ensure the reproducibility of the results, the training of each model was repeated 2 to 4 times with different random seeds. Each replica was then tested by performing a fast simulation of 4 parallel trajectories of a corresponding system, with the objective of a fast assessment of the model. The model that produced the best results was selected for the main validation.

All the statistics obtained using MSMs are reported with an average and standard deviation obtained from estimating 10 different models by bootstrapping the simulation data, taking 90% of the trajectories. This was performed for both reference MD and coarse-grained simulations. In addition, for coarse-grained trajectories, we removed 10% of the initial frames of each trajectory from the analysis to avoid biasing the model with starting conformations. All structures shown were obtained by sampling 10 conformations from the corresponding macrostates.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

All relevant data supporting the key findings of this study are available within the article and its Supplementary Information files. The models and data generated in this study are available at github.com/torchmd/torchmd-protein-thermodynamics (<https://doi.org/10.5281/zenodo.8155342>)⁹⁴. Starting structures for molecular dynamics were sourced from Protein Data Bank <https://www.rcsb.org/>, with PDBIDs: [5AWL](#), [2JOF](#), [1FME](#), [2F4K](#), [1PIN](#), [2HBA](#), [2WXC](#), [1PRB](#), [1ENH](#), [1M10](#), [2A3D](#), [1LMB](#), [1BAL](#), [1GAB](#), [2N35](#), [1DUO](#), [1P7I](#), [1P7J](#), [2HOS](#), [6M3D](#), [2MTQ](#), [1LLI](#), [3KZ3](#). Source data are provided with this paper.

Code availability

All codes are free and available in github.com/torchmd. The code to run molecular dynamics is available at github.com/torchmd/torchmd (<https://doi.org/10.5281/zenodo.8155115>)⁹⁵. The neural network architecture is available at github.com/torchmd/torchmd-net (<https://doi.org/10.5281/zenodo.8155330>)⁹⁶.

References

1. McCammon, J. Protein dynamics. *Rep. Prog. Phys.* **47**, 1 (1984).
2. Henzler-Wildman, K. & Kern, D. Dynamic personalities of proteins. *Nature* **450**, 964–972 (2007).
3. Frauenfelder, H., Sligar, S. G. & Wolynes, P. G. The energy landscapes and motions of proteins. *Science* **254**, 1598–1603 (1991).
4. Diez, M. et al. Proton-powered subunit rotation in single membrane-bound F_0F_1 -ATP synthase. *Nat. Struct. Mol. Biol.* **11**, 135–141 (2004).
5. Eisenmesser, E. Z. et al. Intrinsic dynamics of an enzyme underlies catalysis. *Nature* **438**, 117–121 (2005).
6. McCammon, J. A., Gelin, B. R. & Karplus, M. Dynamics of folded proteins. *Nature* **267**, 585–590 (1977).
7. Karplus, M. & McCammon, J. A. Molecular dynamics simulations of biomolecules. *Nat. Struct. Biol.* **9**, 646–652 (2002).
8. Lindorff-Larsen, K., Piana, S., Dror, R. O. & Shaw, D. E. How fast-folding proteins fold. *Science* **334**, 517–20 (2011).
9. Piana, S., Lindorff-Larsen, K. & Shaw, D. E. Atomic-level description of ubiquitin folding. *Proc. Natl. Acad. Sci. USA* **110**, 5915–5920 (2013).
10. Piana, S., Lindorff-Larsen, K. & Shaw, D. E. Atomistic description of the folding of a dimeric protein. *J. Phys. Chem. B* **117**, 12935–12942 (2013).
11. Plattner, N., Doerr, S., De Fabritiis, G. & Noé, F. Complete protein–protein association kinetics in atomic detail revealed by molecular dynamics simulations and Markov modelling. *Nat. Chem.* **9**, 1005–1011 (2017).
12. Torrie, G. M. & Valleau, J. P. Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *J. Comput. Phys.* **23**, 187–199 (1977).
13. Frenkel, D., Smit, B. & Ratner, M. A. *Understanding Molecular Simulation: From Algorithms to Applications* Vol. 2 (Academic Press, 1996).
14. Sugita, Y. & Okamoto, Y. Replica-exchange molecular dynamics method for protein folding. *Chem. Phys. Lett.* **314**, 141–151 (1999).
15. Fukunishi, H., Watanabe, O. & Takada, S. On the Hamiltonian replica exchange method for efficient sampling of biomolecular systems: Application to protein structure prediction. *J. Chem. Phys.* **116**, 9058–9067 (2002).
16. Izrailev, S. et al. *Computational Molecular Dynamics: Challenges, Methods, Ideas* 39–65 (Springer, 1999).
17. Isralewitz, B., Gao, M. & Schulten, K. Steered molecular dynamics and mechanical functions of proteins. *Curr. Opin. Struct. Biol.* **11**, 224–230 (2001).
18. Laio, A. & Parrinello, M. Escaping free-energy minima. *Proc. Natl. Acad. Sci. USA* **99**, 12562–12566 (2002).
19. Rezende, D. & Mohamed, S. Variational inference with normalizing flows. in *International Conference on Machine Learning* 1530–1538 (2015).
20. Noé, F., Olsson, S., Köhler, J. & Wu, H. Boltzmann generators: sampling equilibrium states of many-body systems with deep learning. *Science* **365**, eaaw1147 (2019).
21. Chavez, L. L., Onuchic, J. N. & Clementi, C. Quantifying the roughness on the free energy landscape: entropic bottlenecks and protein folding rates. *J. Am. Chem. Soc.* **126**, 8426–8432 (2004).
22. Das, P., Matysiak, S. & Clementi, C. Balancing energy and entropy: a minimalist model for the characterization of protein folding landscapes. *Proc. Natl. Acad. Sci. USA* **102**, 10141–10146 (2005).
23. Levitt, M. & Warshel, A. Computer simulation of protein folding. *Nature* **253**, 694–698 (1975).

24. Marrink, S. J. & Tieleman, D. P. Perspective on the Martini model. *Chem. Soc. Rev.* **42**, 6801–6822 (2013).

25. Machado, M. R. et al. The SIRAH 2.0 force field: altius, fortius, citius. *J. Chem. Theory Comput.* **15**, 2719–2733 (2019).

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

27. Izvekov, S. & Voth, G. A. A multiscale coarse-graining method for biomolecular systems. *J. Phys. Chem. B* **109**, 2469–2473 (2005).

28. Noid, W. G. Perspective: Coarse-grained models for biomolecular systems. *J. Chem. Phys.* **139**, 09B201_1 (2013).

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

30. Hills Jr, R. D., Lu, L. & Voth, G. A. Multiscale coarse-graining of the protein energy landscape. *PLoS Comput. Biol.* **6**, e1000827 (2010).

31. Clementi, C., Nymeyer, H. & Onuchic, J. N. Topological and energetic factors: what determines the structural details of the transition state ensemble and “en-route” intermediates for protein folding? An investigation for small globular proteins. *J. Mol. Biol.* **298**, 937–953 (2000).

32. 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* **111**, 7812–7824 (2007).

33. Monticelli, L. et al. The MARTINI coarse-grained force field: extension to proteins. *J. Chem. Theory Comput.* **4**, 819–834 (2008).

34. Koliński, A. et al. Protein modeling and structure prediction with a reduced representation. *Acta Biochim. Pol.* **51**, 349–371 (2004).

35. Davtyan, A. et al. AWSEM-MD: protein structure prediction using coarse-grained physical potentials and bioinformatically based local structure biasing. *J. Phys. Chem. B* **116**, 8494–8503 (2012).

36. Das, R. & Baker, D. Macromolecular modeling with rosetta. *Annu. Rev. Biochem.* **77**, 363–382 (2008).

37. Wang, W. & Gómez-Bombarelli, R. Coarse-graining auto-encoders for molecular dynamics. *npj Comput. Mater.* **5**, 1–9 (2019).

38. Boninsegna, L., Banisch, R. & Clementi, C. A data-driven perspective on the hierarchical assembly of molecular structures. *J. Chem. Theory Comput.* **14**, 453–460 (2018).

39. Foley, T. T., Shell, M. S. & Noid, W. G. The impact of resolution upon entropy and information in coarse-grained models. *J. Chem. Phys.* **143**, 12B601_1 (2015).

40. Foley, T. T., Kidder, K. M., Shell, M. S. & Noid, W. Exploring the landscape of model representations. *Proc. Natl. Acad. Sci. USA* **117**, 24061–24068 (2020).

41. Ruza, J. et al. Temperature-transferable coarse-graining of ionic liquids with dual graph convolutional neural networks. *J. Chem. Phys.* **153**, 164501 (2020).

42. Duvenaud, D. K. et al. Convolutional networks on graphs for learning molecular fingerprints. in *Advances in Neural Information Processing Systems* Vol. 28 (2015).

43. Husic, B. E. et al. Coarse graining molecular dynamics with graph neural networks. *J. Chem. Phys.* **153**, 194101 (2020).

44. Wang, J. et al. Machine learning of coarse-grained molecular dynamics force fields. *ACS Cent. Sci.* **5**, 755–767 (2019).

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

46. Wang, J., Chmiela, S., Müller, K.-R., Noé, F. & Clementi, C. Ensemble learning of coarse-grained molecular dynamics force fields with a kernel approach. *J. Chem. Phys.* **152**, 194106 (2020).

47. Zhang, L., Han, J., Wang, H., Car, R. & E, W. DeePCG: constructing coarse-grained models via deep neural networks. *J. Chem. Phys.* **149**, 034101 (2018).

48. Chen, Y. et al. Machine learning implicit solvation for molecular dynamics. *J. Chem. Phys.* **155**, 084101 (2021).

49. Unke, O. T. et al. SpookyNet: learning force fields with electronic degrees of freedom and nonlocal effects. *Nat. Commun.* **12**, 1–14 (2021).

50. Unke, O. T. et al. Accurate machine learned quantum-mechanical force fields for biomolecular simulations. Preprint at <https://arxiv.org/abs/2205.08306> (2022).

51. Unke, O. T. & Meuwly, M. PhysNet: a neural network for predicting energies, forces, dipole moments, and partial charges. *J. Chem. Theory Comput.* **15**, 3678–3693 (2019).

52. Wang, J. et al. Multi-body effects in a coarse-grained protein force field. *J. Chem. Phys.* **154**, 164113 (2021).

53. Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **98**, 146401 (2007).

54. Rupp, M., Tkatchenko, A., Müller, K.-R. & Von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.* **108**, 058301 (2012).

55. Smith, J. S., Isayev, O. & Roitberg, A. E. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.* **8**, 3192–3203 (2017).

56. Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A. & Müller, K.-R. Schnet—a deep learning architecture for molecules and materials. *J. Chem. Phys.* **148**, 241722 (2018).

57. Qiao, Z., Welborn, M., Anandkumar, A., Manby, F. R. & Miller III, T. F. OrbNet: deep learning for quantum chemistry using symmetry-adapted atomic-orbital features. *J. Chem. Phys.* **153**, 124111 (2020).

58. Kubelka, J., Hofrichter, J. & Eaton, W. A. The protein folding ‘speed limit’. *Curr. Opin. Struct. Biol.* **14**, 76–88 (2004).

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

60. Noid, W. G. et al. The multiscale coarse-graining method. *J. Chem. Phys.* **128**, 244114 (2008).

61. Mullinax, J. & Noid, W. Extended ensemble approach for deriving transferable coarse-grained potentials. *J. Chem. Phys.* **131**, 104110 (2009).

62. Thaler, S. & Zavadlav, J. Learning neural network potentials from experimental data via Differentiable Trajectory Reweighting. *Nat. Commun.* **12**, 1–10 (2021).

63. Doerr, S. et al. Torchmd: a deep learning framework for molecular simulations. *J. Chem. Theory Comput.* **17**, 2355–2363 (2021).

64. Husic, B. E. & Pande, V. S. Markov state models: from an art to a science. *J. Am. Chem. Soc.* **140**, 2386–2396 (2018).

65. Prinz, J. H. et al. Markov models of molecular kinetics: generation and validation. *J. Chem. Phys.* **134**, 174105 (2011).

66. Singhal, N., Snow, C. D. & Pande, V. S. Using path sampling to build better Markovian state models: predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. *J. Chem. Phys.* **121**, 415–425 (2004).

67. Noé, F. & Fischer, S. Transition networks for modeling the kinetics of conformational change in macromolecules. *Curr. Opin. Struct. Biol.* **18**, 154–162 (2008).

68. Pan, A. C. & Roux, B. Building Markov state models along pathways to determine free energies and rates of transitions. *J. Chem. Phys.* **129**, 064107 (2008).

69. Zemla, A. LGA: a method for finding 3D similarities in protein structures. *Nucleic Acids Res.* **31**, 3370–3374 (2003).

70. 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.* **139**, 07B604_1 (2013).

71. 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.* **9**, 2000–2009 (2013).

72. McCallister, E. L., Alm, E. & Baker, D. Critical role of β -hairpin formation in protein G folding. *Nat. Struct. Biol.* **7**, 669–673 (2000).

73. Kmiecik, S. & Kolinski, A. Folding pathway of the B1 domain of protein G explored by multiscale modeling. *Biophys. J.* **94**, 726–736 (2008).

74. Kuhlman, B. & Baker, D. Exploring folding free energy landscapes using computational protein design. *Curr. Opin. Struct. Biol.* **14**, 89–95 (2004).

75. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).

76. Baek, M. et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871–876 (2021).

77. Thölke, P. & De Fabritiis, G. TorchMD-NET: equivariant transformers for neural network based molecular potentials. Preprint at <https://arxiv.org/abs/2202.02541> (2022).

78. Jin, J., Pak, A. J., Durumeric, A. E., Loose, T. D. & Voth, G. A. Bottom-up coarse-graining: principles and perspectives. *J. Chem. Theory Comput.* **18**, 5759–5791 (2022).

79. Kanekal, K. H., Rudzinski, J. F. & Bereau, T. Broad chemical transferability in structure-based coarse-graining. *J. Chem. Phys.* **157**, 104102 (2022).

80. Harvey, M. J., Giupponi, G. & De Fabritiis, G. ACEMD: accelerating biomolecular dynamics in the microsecond time scale. *J. Chem. Theory Comput.* **5**, 1632–1639 (2009).

81. Buch, I., Harvey, M. J., Giorgino, T., Anderson, D. P. & De Fabritiis, G. High-throughput all-atom molecular dynamics simulations using distributed computing. *J. Chem. Inf. Model.* **50**, 397–403 (2010).

82. Piana, S., Lindorff-Larsen, K. & Shaw, D. E. How robust are protein folding simulations with respect to force field parameterization? *Biophys. J.* **100**, L47–49 (2011).

83. Jorgensen, W. L., Chandrasekhar, J., Madura, J. D., Impey, R. W. & Klein, M. L. Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.* **79**, 926–935 (1983).

84. Doerr, S. & De Fabritiis, G. On-the-fly learning and sampling of ligand binding by high-throughput molecular simulations. *J. Chem. Theory Comput.* **10**, 2064–2069 (2014).

85. Feenstra, K. A., Hess, B. & Berendsen, H. J. Improving efficiency of large time-scale molecular dynamics simulations of hydrogen-rich systems. *J. Comput. Chem.* **20**, 786–798 (1999).

86. Pérez, A., Herrera-Nieto, P., Doerr, S. & De Fabritiis, G. Adaptive-Bandit: a multi-armed bandit framework for adaptive sampling in molecular simulations. *J. Chem. Theory Comput.* **16**, 4685–4693 (2020).

87. Doerr, S., Harvey, M. J., Noé, F. & De Fabritiis, G. HTMD: high-throughput molecular dynamics for molecular discovery. *J. Chem. Theory Comput.* **12**, 1845–1852 (2016).

88. Schütt, K. et al. SchNetPack: a deep learning toolbox for atomistic systems. *J. Chem. Theory Comput.* **15**, 448–455 (2018).

89. Fey, M. & Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. Preprint at <https://arxiv.org/abs/1903.02428> (2019).

90. Falcon, W. A. et al. PyTorch Lightning. GitHub repository. <https://github.com/PyTorchLightning/pytorch-lightning> (2019).

91. Durumeric, A. E. et al. Machine learned coarse-grained protein force-fields: are we there yet? *Curr. Opin. Struct. Biol.* **79**, 102533 (2023).

92. Fu, X. et al. Forces are not enough: benchmark and critical evaluation for machine learning force fields with molecular simulations. Preprint at <https://arxiv.org/abs/2210.07237> (2022).

93. Paszke, A. et al. Pytorch: an imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32**, 8026–8037 (2019).

94. Majewski, M. et al. Machine learning coarse-grained potentials of protein thermodynamics. GitHub repository. <https://doi.org/10.5281/zenodo.8155343> (2023).

95. Doerr, S. et al. TorchMD. GitHub repository. <https://doi.org/10.5281/zenodo.8155115> (2020).

96. Thölke, P. & Fabritiis, G. D. TorchMD-NET. GitHub repository. <https://doi.org/10.5281/zenodo.8155330> (2022).

Acknowledgements

The project PID2020-116564GB-I00 has been funded by MCIN/AEI/10.13039/501100011033 (G.D.F.) This project has received funding from the Torres-Quevedo Program from the Spanish National Agency for Research (PTQ2020-011145/AEI/10.13039/501100011033) (M.M.); the Torres-Quevedo Program from the Spanish National Agency for Research (PTQ2021-011669/AEI/10.13039/501100011033) (A.P.); the European Union's Horizon 2020 research and innovation program under grant agreement No. 823712 (G.D.F.); NLM Training Program in Biomedical Informatics and Data Science (grant no. 5T15LM007093-27) (N.E.C.); Deutsche Forschungsgemeinschaft (DFG, GRK DAEDALUS, RTG 2433, Project Q05) (N.E.C.); National Science Foundation (CHE-1900374 and PHY-2019745) (C.C.); Einstein Foundation Berlin (Project 0420815101) (C.C.); Deutsche Forschungsgemeinschaft (DFG) SFB 1114 projects A04, B03, and B08, SFB/TRR 186 project A12, and SFB 1078 project C7 (C.C.); Deutsche Forschungsgemeinschaft (DFG) projects CRC1114/A04, CRC1114/C03 (F.N.); European Research Council (ERC) project ERG CoG 772230 (F.N.); Berlin Mathematics Center MATH+ project AA1-6 (F.N.); the National Institute of General Medical Sciences (NIGMS) of the National Institutes of Health under award number GM140090 (G.D.F.). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. The authors thank the volunteers of GPUGRID.net for donating computing time. T.G. acknowledges the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support. This project has received funding from the Spoke 7 of the National Centre for HPC, Big Data and Quantum Computing (CN00000013) of the NextGenerationEU initiative (T.G.).

Author contributions

M.M. and A.P. contributed equally to this work. G.D.F., F.N. and C.C. conceived the presented idea. G.D.F., M.M., and A.P. designed the study. M.M. and A.P. carried out the simulations and analyzed the data. N.E.C., T.G. and B.E.H. helped to analyze and interpret the data. S.D., P.T. contributed software used in the work. M.M., A.P. and G.D.F. wrote the manuscript. All authors provided critical feedback and helped shape the research, analysis and manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41467-023-41343-1>.

Correspondence and requests for materials should be addressed to Cecilia Clementi, Frank Noé or Gianni De Fabritiis.

Peer review information *Nature Communications* thanks the anonymous reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023