Scalable emulation of protein equilibrium ensembles with generative deep learning

Sarah Lewis[1†], Tim Hempel[1†], José Jiménez-Luna[1†], Michael Gastegger[1†], Yu Xie[1†], Andrew Y. K. Foong[1†], Victor García Satorras[1†], Osama Abdin[1†], Bastiaan S. Veeling[1†], Iryna Zaporozhets[1,2], Yaoyi Chen[1,2], Soojung Yang[1], Adam E. Foster[1], Arne Schneuing[1], Jigyasa Nigam[1], Federico Barbero[1], Vincent Stimper[1], Andrew Campbell[1], Jason Yim[1], Marten Lienen[1], Yu Shi[1], Shuxin Zheng[1], Hannes Schulz[1], Usman Munir[1], Roberto Sordillo[1], Ryota Tomioka[1], Cecilia Clementi[1,2,3], Frank Noé[1,2,3,*]

[1]AI for Science, Microsoft Research [2]Freie Universität Berlin, Berlin 14195, Germany

[3]Rice University, Houston, TX 77005, USA

[*]Corresponding author. Email: franknoe@microsoft.com

[†]These authors contributed equally to this work.

Following the sequence and structure revolutions, predicting functionally relevant protein structure changes at scale remains an outstanding challenge. We introduce BioEmu, a deep learning system that emulates protein equilibrium ensembles by generating thousands of statistically independent structures per hour on a single GPU. BioEmu integrates over 200 milliseconds of molecular dynamics (MD) simulations, static structures and experimental protein stabilities using novel training algorithms. It captures diverse functional motions – including cryptic pocket formation, local unfolding, and domain rearrangements – and predicts relative free energies with 1 kcal/mol accuracy compared to millisecond-scale

MD and experimental data. BioEmu provides mechanistic insights by jointly modelling structural ensembles and thermodynamic properties. This approach amortizes the cost of MD and experimental data generation, demonstrating a scalable path towards understanding and designing protein function.

Proteins and protein complexes are the functional building blocks of life and play a central role in drug development and biotechnology. While next-generation sequencing and deep learning-based structure prediction tools (*1–4*) have revolutionized access to sequence and structure, scalable methods for exploring protein function remain elusive. A key driver of protein function is the ability to transition between distinct conformational states (i.e., sets of different structures), often coupled to the binding of ligands or other proteins. For example, actin's ability to form muscle fibers arises from its ATP/ADP-regulated conformational dynamics (Fig. 1A).

Current technologies that quantitatively probe such conformational transitions and their coupling with binding states are not scalable. Single-molecule experiments can provide the full equilibrium distributions of observables such as intramolecular distances (*5*), but require bespoke molecular constructs and time-consuming data collection. Cryo-electron microscopy can resolve multiple conformational states of biomolecular complexes and their probabilities (*6*), but these experiments are time-consuming and costly. Molecular Dynamics (MD) simulation is, in principle, a universal tool that allows both structure and dynamics of biomolecules to be explored at all-atom resolution. However, biomolecular forcefields are far from perfect and the sampling problem makes studying protein folding or association via MD a feat of epic computational costs – even with special-purpose supercomputers or enhanced sampling methods (*7, 8*). Machine-learned (ML) coarse-grained MD models have an opportunity to achieve similar accuracy as all-atom MD at 2-3 orders of magnitude lower computational cost (*9, 10*) but are still under development.

The grand challenge to complete our understanding of protein function thus motivates the development of a technology that can elucidate protein conformational states and binding states, as well as their associated probabilities. This technology should ideally achieve an accuracy comparable to a converged MD simulation, or a cryo-EM experiment with multi-conformation analysis, but it should only require a few hours of wall-clock time and cost no more than a few dollars per experiment. Boltzmann Generators (*11*) (BGs) have demonstrated that physics-based generative ML models can sample equilibrium distributions of arbitrary molecular energy functions, however scaling such approaches to large macromolecules while maintaining high sample efficiency is challenging. Concurrently, data-based generative ML models, such as diffusion models are now widely used in protein structure prediction and design (*2, 4*). Such models (*12–14*), as well as perturbation-based derivatives of AlphaFold (*15, 16*) have also been shown to be capable of generating distinct protein structures and can be combined with MD simulation to alleviate the sampling problem (*17*). As yet, generative ML systems have mainly demonstrated an ability to qualitatively sample distinct protein conformational states. A demonstration that generative ML can quantitatively match equilibrium ensembles and predict experimental observables is critical going forward (*18*).

Model

Here we develop a biomolecular emulator, BioEmu – a generative deep learning system designed to approximately sample from the equilibrium distribution of protein conformations. BioEmu uses a similar model architecture as Distributional Graphormer (DiG) (*12*), but employs a distinct training approach. Starting from the input protein sequence, single and pair representations of the sequence are computed using the AlphaFold2 evoformer (*1*). This sequence representation serves as input to a denoising diffusion model that generates protein structures (Fig. 1B,C; materials and methods). Sequence encoding is performed only once per protein, and an efficient integration scheme enables structure generation in as few as 30-50 denoising steps (Fig. S11, materials and methods). As a result, 10,000 independent protein structures from the learned equilibrium distribution can be sampled within minutes to a few hours on a single GPU, depending on their size.

A major challenge in training BioEmu is the absence of a single high-quality data source for protein equilibrium distributions, due to the aforementioned challenges with experimental methods and MD (*19*). We therefore integrate training data from different, complementary sources. BioEmu is pre-trained on a clustered version of the AlphaFold database (AFDB) using a data augmentation strategy that encourages it to sample diverse conformations (Fig. 1D,E, materials and methods). In a second stage, we continue training the model on over 200 milliseconds of all-atom MD data of thousands of small-to-medium proteins (Fig. 1D, Table S1, materials and methods). To mitigate the sampling problem, MD data was reweighed towards equilibrium using either Markov State Models (*20*), or weights from experimental data, when possible (materials and methods).

Finally, we fine-tune the model on 500,000 sequences of the MEGAscale dataset (*21*), a large-scale, homogeneous collection of *in vitro* protein stability measurements (Fig. 1D). As the MEGAscale dataset does not contain structures, we developed a new algorithm called property-prediction fine-tuning (PPFT) that can efficiently incorporate experimental measurements into diffusion model training (Fig. 4A, materials and methods). To ensure generalization, we filter our training set such that no protein has more than 40% sequence similarity to a pre-defined holdout set including all reported test proteins of at least 20 residues or longer. The term "BioEmu" refers to the fine-tuned model, trained on AFDB, MD simulations and experimental protein stability measurements. Subsequent results use this model unless otherwise described.

Sampling conformational changes related to protein function

We consider the ability to qualitatively sample distinct, biologically relevant conformations as a foundation for building a quantitative equilibrium sampler. Therefore, we first test whether BioEmu can predict known conformational changes and compare this capability with AFCluster (*15*), AlphaFlow (*13*), DiG (*12*), and uniform MSA subsampling (*22*) as representative baseline methods (materials and methods). To benchmark BioEmu's ability to capture functionally relevant structural variability, we curated four benchmark sets comprising around 100 proteins with experimentally validated transitions (Fig. S1-S4). The first set, OOD60, assesses sequence generalization. The remaining three target specific types of conformational change: domain motions, local unfolding transitions, and cryptic pocket formations.

OOD60 is designed to tests strong generalization: its proteins were deposited in the PDB after the AlphaFold2 cutoff date, and share no more than 60% and 40% sequence similarity with the AlphaFold2 monomer model and BioEmu training sets, respectively. OOD60 includes various challenging cases such as large-scale conformational changes induced by binding with other biomolecules (Fig. S1). BioEmu significantly outperforms all baselines on this benchmark (Fig. S7). For the other benchmarks, BioEmu matches or exceeds baseline performance, except in predicting apo states of cryptic pockets, where AFCluster performs best. The performance gap is especially pronounced for proteins outside the AlphaFold2 training set (Fig. S7).

The domain motion benchmark consists of proteins that undergo large-scale motions as part of their functional cycle (Fig. 2A). In the open–close transition of Adenylate Kinase, the closed state brings the substrates together to catalyze the $\mathrm{ATP} + \mathrm{AMP} \rightleftharpoons 2\mathrm{ADP}$ reaction. Single-molecule experiments have confirmed that opening and closing occurs reversibly on timescales of tens of microseconds when the substrates are bound (*23*). BioEmu predicts a range of open and closed states, including close matches with crystallographic structures. A second example is the open-close transition of LAO-binding protein which is required to bind and release lysine, arginine and ornithine for transport across membranes as part of the ATP-binding cassette protein family. Another interesting example of domain motions is that of the receptor module which regulates the concentration of cyclic di-GMP in bacteria. In this case one domain undergoes a large-scale rotation and repacks to the other domain with a completely different contact pattern. See Fig. S2 for 15 further examples. Overall, BioEmu predicts 83% of the reference experimental structures with ≤ 3 Å RMSD (Fig. 2A), indicating the model's ability to predict which protein regions are rigid and which are flexible, as well as which resulting motions can occur.

Next we consider local unfolding transitions, in which part of a protein chain unfolds or detaches from its main structure as part of a signaling pathway (Fig. 2B). Predicting local unfolding challenges the model to correctly rank the relative stabilities of a protein's fold. A famous example of local unfolding is Ras p21, a conformational switch which signals cell growth and whose mutants are often linked to cancer development (*24*). In its active state, stabilized by binding GTP, the Switch II region forms a short alpha-helix, which partially unfolds in the inactive GDP-bound state. Rhomboid intramembrane protease is a more complex case involving domain swapping. Its monomeric form features a globular conformation, while in its dimeric form the central beta-sheet unfolds and the helices of the two monomers bind to each other. Finally, CaM Kinase II presents an autoinhibition mechanism, wherein the N-terminus binds into the active site. BioEmu correctly predicts the local unfolding of these structure elements, and samples 70% of the folded and 81% of locally unfolded states across 20 protein examples (Fig. 2B, Fig. S3).

As a final class of conformational changes we consider the formation of pockets that are absent in the apo PDB structure but can form to bind a small molecule (Fig. 2C). Such "cryptic" binding pockets can be discovered with high-performance MD simulation (*25, 26*), but the millisecond timescales often involved in the spontaneous opening of such pockets make MD on commercial hardware rarely viable for *in silico* drug discovery pipelines. We have curated 34 cases of experimentally-validated formation of cryptic binding pockets from the literature (Fig. S4). The sialic acid binding factor presents a case where a large opening in the apo state can partially close and form a binding site for the ligand. Fascin is a four-domain protein where two domains can rotate with respect to each other, to reveal a binding site. In Glu PRPP amidotransferase, part of the chain is unfolded in the apo state and folds into a structure that completes the binding site for the ligand. To ensure capturing subtle changes, we define success by a very strict 1.5Å RMSD threshold to the apo and holo reference structures. Surprisingly the model has a strong preference for holo states, successfully predicting the cryptic pocket in 86% of cases, while it only succeeds in predicting 56% of the apo structures, indicating further room for improvement (Fig. 2C). We hypothesize that the model may be picking up a bias implicit in the embeddings - proteins may have a few apo structures deposited in the PDB, but it is common to find multiple structures of the same protein with different small molecules bound.

To conclude, we conducted several tests to confirm that BioEmu's multi-conformation prediction is a hallmark of generalization rather than memorization of sequence-structure pairs. BioEmu's ability to predict multiple conformations depends only weakly on the sequence similarity between the query molecule and the training set, with the clearest trend seen for domain motions where the final performance is reached at 30% sequence similarity (Fig. S5, materials and methods). As our test proteins contain examples that overlap with the AlphaFold2 training set, we ablated whether multi-conformation prediction performance stems from trivial extraction of the information already present in the AlphaFold2 evoformer embeddings. To this end we trained an end-to-end version of the model that uses MSA information directly instead of pre-trained embeddings, on a training set that is at most 40% sequence similar to any protein belonging to the the multi-conformation benchmarks. The resulting model shows similar performance in the previously mentioned multi-conformation benchmarks to the fine-tuned BioEmu (Fig. S6, materials and methods).

Emulating MD equilibrium distributions

A major motivation for developing BioEmu is to overcome the well-known sampling problem in molecular dynamics (MD): simulating the full range of protein conformations and estimating their equilibrium probabilities often requires extensive MD simulations – on the order of 100 microseconds to 10 milliseconds (*7, 8, 27*). These timescales are necessary to capture rare but functionally important transitions, yet achieving them is computationally demanding – if not prohibitive, even with specialized supercomputers (*28*) or large-scale distributed simulations integrated via statistical models (*8, 29*). Here, we assess BioEmu's ability to emulate the equilibrium distribution that would be sampled with extensive MD simulations. To this end, we have amassed all-atom simulations of proteins with a total aggregated simulation time of over 200 ms (Table S1), which are used for fine-tuning BioEmu (Fig. 1D).

Before analyzing the model trained on the full dataset, we first tested whether our machine learning architecture and training protocol permit learning to emulate long-timescale MD equilibrium distributions. To this end, we used D. E. Shaw Research (DESRES) simulations of 12 fast-folding proteins generated on the Anton supercomputer (*7*). For each protein, we fine-tuned a separate model on the other 11 proteins and evaluated it on the held-out 12th – an approach known as leave-one-out cross-validation (see materials and methods). This setup ensures that each test case is evaluated independently of its training data and avoids bias from arbitrary train–test splits in this small dataset. As expected, the AFDB-pretrained model predicts the native state but performs poorly in sampling the full free energy surface (Fig. S9). Surprisingly, however, the "DESRES-finetuned models", each trained on only 11 fast folders, predict free energy surfaces on the held-out proteins that closely match the MD ground truth (Fig. 3A, Fig. S9). For all proteins, the model predicts both native as well as the unfolded states with similar shapes on the free energy landscape. In many cases, several or all folding intermediates visible on the two-dimensional free energy surface are predicted (Fig. 3A, Fig. S9): For beta-beta-alpha protein (BBA), both MD and the DESRES-finetuned models predict the existence of an intermediate with the alpha-helix formed and the beta-sheet broken. For protein G, both MD and the DESRES-finetuned models sample intermediates with half of the beta-sheet still formed, while the other half and most of the helix are broken. For homeodomain, MD and model agree in the prediction of an intermediate with only one helix turn unwound, while the unfolded states still feature some degree of helical propensity. There is an excellent agreement of the predicted secondary structure propensities with the MD data (Fig. 3A). Quantitatively, the mean absolute error between the MD and model 2D free energy landscapes is only 0.74 kcal/mol, ranging from 0.30 kcal/mol for BBA to 1.63 kcal/mol for λ-repressor, which is on the order of differences expected from two different classical MD force fields (*30, 31*).

We compare the computational costs of MD data generation and BioEmu in GPU-hours (on an NVIDIA Titan V; Fig. 3A, top right). For all BioEmu results shown here, we draw 10k samples, which incurs computational costs of under one GPU-minute for Chignolin to around one GPU-hour for λ-repressor. For MD we consider the cost for generating the DESRES simulations, whose lengths have been chosen to include roughly 10 folding-unfolding transitions. The MD costs then range from 2,000 GPU-hours for Chignolin to more than 100,000 GPU-hours for NTL9, resulting in a speedup of BioEmu over MD of four to five orders of magnitude. We also note that for most proteins shown here, performing sufficiently long MD simulations to directly observe folding and unfolding in single trajectories is still not possible on consumer-grade hardware but instead requires a much more complex methodological framework (*29, 32*).

The main BioEmu model is fine-tuned on more than 200 ms MD simulations with Amber force fields at or near a temperature of 300K (Table S1). We chose to combine data from slightly different simulation conditions as each of these MD models is inherently imperfect, and we regarded experimental data as being more reliable for weighing between conformations (Fig. 1D). Differences in the simulation conditions of our own generated data are intentional, e.g. AMBER ff99sb-disp (*33*) was chosen to avoid spuriously misfolded states produced by other force fields in the context of protein folding (materials and methods). A large fraction of training data, 46 ms, is dedicated to 1100 CATH domains, common building blocks of protein structure (*34*) (materials and methods). We designate 17 CATH systems with more than 100 μs simulation time as test set and report statistics comparing MD and model distributions (Fig. 3B, Fig. S10, materials and methods). Similar as for DESRES simulations, BioEmu predicts the native state with local fluctuations and typically several other substates and structures sampled by MD. Most secondary structure propensities match well (Fig. 3B). We observed a free energy mean absolute error over the converged test set of 0.9 kcal/mol, again comparable to the differences expected between different MD force fields.

To understand whether our model's ability to sample accurate equilibrium distributions is limited by training data or model expressivity, we trained 3 models with the same architecture as BioEmu from scratch, using only 1%, 10% and 100% of the CATH systems in the training dataset. We observed decreased free energy errors and an increased coverage of the conformations sampled by MD as the amount of training proteins increased (Fig. 3B, bottom right), suggesting that the model can be further improved by adding more training data. Notably, the finetuned model's error on the same test set is further reduced to 0.9 kcal/mol, demonstrating the potential benefit of pre-training and integrating multiple datasets even if they do not use identical simulation conditions.

Finally, we have evaluated BioEmu for two case studies that involve larger proteins: Complexin II (134 aa) and tetraspanin CD9 (225 aa). Complexin II is an intrinsically disordered protein (IDP) from the neurotransmitter release apparatus (*35*). IDPs tend to be difficult to sample with MD, however, BioEmu can efficiently emulate a flexible ensemble of complexin II structures (Fig. 3C) while reproducing known secondary structure elements such as the central and accessory helices (*35, 36*). Achieving convergence of IDPs of this size with all-atom MD is unpractical. At an orders of magnitude higher computational cost than with BioEmu, we have conducted $\sim 5\ \mu s$ of MD simulations with all-atom MD, which are most likely not converged but already display qualitatively different behavior: The AMBER ff14sb force field produces a very rigid compact structure with a small radius of gyration and little to no variation in secondary structure content, whereas AMBER ff99sb-disp tends to destabilize known secondary structure elements (Fig. 3C).

The second case-study, tetraspanin CD9, plays a role in cell adhesion and fusion (Fig. 3D). The large extracellular loop (LEL) of CD9 is part of our OOD60 test set (Fig. S1) in which our pre-trained model samples both crystallographic reference structures (6rlo, 6rlr), whereas the BioEmu model finetuned on MD data samples 6rlo but discards 6rlr. This is consistent with the observation that both structures exist in crystal environments, however, 6rlr cannot be realized in a folded monomeric protein and is therefore correctly discarded when fine-tuning BioEmu (Fig. S12A-C). We also sample the full-length CD9 structure, which has less than 40% sequence similarity to both BioEmu and AlphaFold training sets (Fig. 3D). In agreement with MD simulations of (*37*), BioEmu predicts the widely open state 1 and closed state 2 and similar contact distributions between the small and large extracellular loops (SEL, LEL) as reported in (*37*). A principal component analysis reveals that BioEmu and MD sample similar sets of conformations (Fig. 3D). MD predicts an experimentally-unknown metastable closed state 3 which is unstable in BioEmu. BioEmu's samples closed states 2 very similar with the experimental structure 6k4j (1.9 Å RMSD), whereas the closest MD sample has an RMSD of 4.6 Å to the crystal structure (Fig. S12D).

Predicting protein stabilities

Understanding protein stability is crucial for various applications in molecular biology, drug design, and biotechnology. From a modeling point of view, predicting a protein's stability is a specific case of predicting the equilibrium probabilities of its different conformational states, and these all arise from the same underlying biophysics. We therefore desire to train BioEmu so that the proportion of samples in folded and unfolded states matches the experimentally-measured protein stability. We classify protein structures as folded or unfolded based on their fraction of native contacts, and define the folding free energy as $\Delta G = G_{\text{folded}} - G_{\text{unfolded}}$ (materials and methods).

To facilitate protein stability prediction, BioEmu is trained on 502,442 mutant sequences generated from 361 wild types, a subset of the over 674,000 experimental measurements in the MEGAscale dataset (materials and methods) (*21*). We evaluate BioEmu on a test set of randomly chosen mutants from 95 wild types. For a subset of 271 wildtype proteins and 21458 mutants, we have additionally conducted a total of 25 ms of all-atom MD simulations of the folded and unfolded states (materials and methods). To address MD sampling and force field issues, we weigh the folded and unfolded samples based on the experimentally-measured protein stabilities (materials and methods). To accelerate training convergence and leverage the large number of MEGAscale measurements, we developed the Property Prediction Fine-Tuning algorithm (PPFT; Fig. 4A, materials and methods) that integrates experimental expectation values, such as protein stabilities, into diffusion model training without requiring protein structures. PPFT uses fast approximate sampling with only 8 denoising steps, which we observed to be sufficient to confidently predict whether each sampled structure will be classified as folded or unfolded. By comparing the mean foldedness of sampled structures with experimental measurements and backpropagating the error, our model can be efficiently trained to match experimental protein stabilities.

BioEmu achieves a mean absolute error below 0.9 kcal/mol and a Spearman correlation coefficient of approximately 0.6 for the MEGAscale test proteins (Fig. 4B). The BioEmu ensembles also correlate well with stability changes of point mutants, $\Delta\Delta G$, achieving mean absolute errors below 0.8 kcal/mol and a Spearman correlation coefficient above 0.6 (Fig. 4C). Errors of approximately 1 kcal/mol are achieved for test proteins that have 40% sequence similarity with the training set, but the best performance is obtained for test sets that include sequences with 50% or greater similarity

(Fig. 4B,C). As there are only 361 distinct wild-type sequences in the MEGAscale training data, it is likely generalization performance can be further improved with a training dataset that is more diverse in protein sequence space.

To check whether BioEmu makes physically reasonable predictions outside the MEGAscale set of proteins, we tested it on proteins that are known to be very stable or unstable. We first selected stable proteins from ProThermDB (*38*) with $\Delta G < -8$ kcal/mol (materials and methods) Our model consistently samples these proteins in their folded states with a fraction of native contacts always exceeding 0.65 (Fig. 4D). To test whether our model systematically predicts intrinsically disordered proteins (IDPs) as unfolded, we used the CALVADOS test set (*39*). Most proteins sampled displayed a radii of gyration (R_g) similar to random coil structures and mostly larger than typical folded proteins (Fig. 4E). Unlike other models (*40, 41*), ours has not been directly trained on IDPs; nonetheless, it provides zero-shot predictions of R_g that correlate well with experimental measurements (Fig. 4E).

In comparison with black box methods that predict protein stability directly from sequences (*42–45*), BioEmu has competitive or superior prediction accuracy. However, in contrast to a black-box prediction of ΔG, we can analyze the structure ensemble generated by our model to reveal insights on mutation-caused stability changes. For illustration, we show mutants of the design protein HHH_rd1_0335 and PDB entry 2JWS (Fig. 4F). In HHH_rd1_0335, the mutation I7P leads to a destabilization of the first helix, as indicated by the model's prediction of a $\Delta\Delta G$ of 1.8 kcal/mol compared to the experimental 2.1 kcal/mol. The analysis shows a decrease in average helicity which particularly affects the helix where the mutation is located. For 2JWS, the mutation I24D in the middle helix results in partial unfolding, with the model predicting a $\Delta\Delta G$ of 2.1 kcal/mol, closely matching the experimental value of 2.9 kcal/mol. This mutation replaces a hydrophobic residue with a negatively charged aspartate, disrupting core stability and leading to a localized structural change. These analyses highlight BioEmu's ability to correlate predictions of thermodynamics with structural causes, which is not possible with black-box prediction models.

Discussion

We have introduced BioEmu, a generative machine learning system to approximately sample the equilibrium distributions of proteins and through that explore two key aspects of molecular function: protein conformations and their equilibrium probabilities. The system has been demonstrated to sample experimentally known structures of proteins undergoing a variety of conformational changes, to approximate the equilibrium distributions of extensive MD simulations, and to predict experimentally-measured protein stabilities within errors of 1 kcal/mol. The cost of running inference is on the order of one GPU-hour per computational experiment — many orders of magnitude less than running MD simulations even if enhanced sampling methods are invoked, and orders of magnitude cheaper than experiments that can provide detailed structure-function relationships. Nonetheless, there are further opportunities to reduce BioEmu's inference cost. Conditional flow-matching (*46*) can be used to generate protein structures using even fewer integration steps (*47*). The computational cost of evaluating the transformer network in the score model (Fig. 1C) can potentially be reduced by leveraging sparse or low-rank attention mechanisms.

BioEmu and MD simulation are complementary: our system was trained on large amounts of MD simulation data for soluble proteins, and within this scope, it has shown to be able to approximate MD distributions at a tiny fraction of the MD simulation costs. However, BioEmu is not designed to generalize beyond this scope — for example membrane environments and small molecule ligands are neither represented in the model nor in the training data, and reliable predictions cannot be assumed when such factors play a key role in the process. In contrast, MD can be readily generalizing to such conditions, though it remains constrained by the sampling problem. Our system can be used to generate a guess for the equilibrium distribution, and MD trajectories can be launched from a BioEmu ensemble in order to obtain chemically accurate all-atom structures, refine the distribution, and even compute dynamical properties. We therefore do not expect emulators such as BioEmu to make MD simulation obsolete; rather, we anticipate we anticipate that MD will increasingly serve as a data generation and validation tool. A similar shift of roles is already in progress for other simulator-emulator pairs such as quantum chemistry methods and machine-learned forcefields.

An important limitation of BioEmu is that it generates distributions entirely empirically, whereas MD simulation uses potential energy functions which are connected to equilibrium distributions and expectation values by statistical mechanics. If direct access to a reduced potential energy function $u(x)$ was available that is consistent with the generated distribution by $p(x) \propto \mathrm{e}^{-u(x)}$, it could be used for reweighting and making rigorous enhanced sampling simulations available through the emulator. BioEmu can potentially also be improved by going beyond score matching and using energy or forces information from MD force fields at training time, as considered in Boltzmann Generators (*11*) and variational force matching (*9, 48*).

While BioEmu samples approximate equilibrium distribution, it does not model protein dynamics which is done by MD and other methods (*49*), nor does its training incorporate dynamical information as it does for Markov State Models (*20*). A pragmatic approach to generate a dynamic ensemble is to predict starting points with BioEmu and launch MD simulations from them. On the other hand, a starting point for a future model architecture that can both model dynamics and exploit dynamical information in the training data is the Implicit Transfer Operator approach (*50*).

We have demonstrated that using the PPFT method developed here, BioEmu can be efficiently fine-tuned on experimental data. Here, we have chosen to do that for protein stabilities using the MEGAscale dataset, because it presents a very favorable tradeoff of large data scale and quantitative reliability. However, in principle, PPFT can be used to fine-tune BioEmu and other diffusion models to match any set of experimental observables, including Nuclear Magnetic Resonance data, small-angle X-ray scattering, fluorescence measurements etc. Being able to fine-tune the generated ensemble to arbitrary experimental data is an important advantage compared to MD forcefields: these can also be tuned to fit experimental data (*51*), but the processes that give rise to the experimental observables must be sampled during the training process — a task that is tedious or even unfeasible for observables that involve complex rare events, such as folding free energies.

A widely used feature of AlphaFold is its ability to predict confidence in an output structure, and a similar confidence module for BioEmu would be highly desirable. Training a confidence module is relatively straightforward in AlphaFold, which serves a single prediction task (structure) and relies on a single ground truth dataset (the PDB), whereas equilibrium structure ensembles serve multiple downstream tasks and observables and no universal ground truth dataset is available. Confidence prediction or uncertainty quantification of arbitrary observables thus remains an important future research direction and may leverage ongoing research in the deep learning community (*52*). Even a rough notion of model confidence in observables, such as free energy differences, could be exploited to improve training data efficiency: In the MD community, Markov State Models and other kinetic models have been used to guide MD data production in an active learning loop (*8, 53*), and a similar approach could be implemented to have BioEmu request new MD or experimental data that are most likely to increase model confidence.

Another limitation of the current system is that it emulates single protein chains under a fixed thermodynamic condition of 300K. A proper emulator for proteins requires conditioning on experimentally and biologically relevant parameters such as temperature and pH, and needs to be able to model multiple interacting molecules, as proteins rarely have a function on their own. We envision two ways of achieving this: (i) training on additional MD simulation data at relevant thermodynamic ranges, and (ii) the incorporation of relevant additional experimental data (e.g., melting curves for temperature) into finetuning strategies.

An important future direction is to extend BioEmu's modalities by incorporating multiple protein chains and ligands, which are already included in recent biomolecular structure prediction systems (*2, 4*). Currently, oligomer and ligand binding state are implicit, which may cause biases in the training data to show up in the sampled distribution. A hallmark of this may be BioEmu's preference to predict holo over apo structures in the cryptic pocket benchmark (Fig. 2C). Such biases in the sampling distribution can be avoided by explicitly conditioning the prediction of the structure ensemble to ligands, which is however hampered by the lack of training data. While we have shown that the ability to accurately emulate the equilibrium distributions of small proteins increases with more training data, the sampling problem limits MD as a data generation engine. For learning changes of conformation and binding state of large biomolecular complexes, as well as learning the subtle binding affinity differences between binding partners, and ultimately tackle the quest for reliably predicting protein function, highly scalable experimental techniques that can be incorporated as training data will become key.

These results demonstrate that the large upfront costs of MD simulation and experimental data generation can be amortized by a deep learning emulator whose prediction error decreases with an increasing amounts of high-quality training data. This indicates a path forward for predicting biomolecular function at genomic scale.

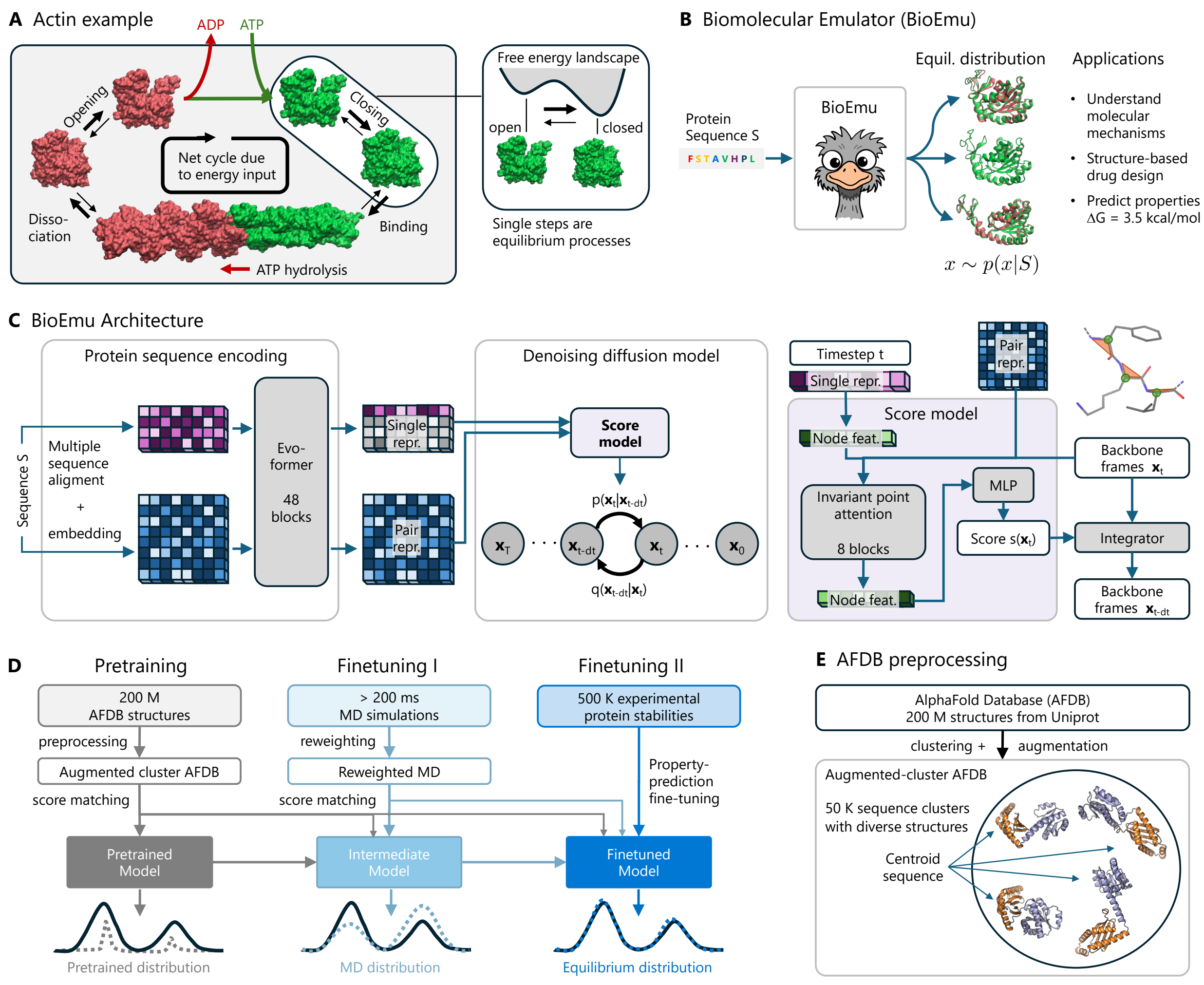

Figure 1: **Overview of model and architecture.** (**A**) Actin as a representative example of protein function driven by conformational changes. Actin filament formation depends on open/close transition of monomers, which is controlled by ADP/ATP binding. (**B**) Given a protein sequence, BioEmu samples protein structures from an approximate equilibrium distribution, from which properties such as free energy differences can be computed. (**C**) ML model architecture consisting of protein sequence encoder, denoising diffusion model and score model. The protein structure is represented using coarse-grained backbone frames. (**D**) Data integration and model training pipeline. (**E**) Data processing pipeline for pre-training. Abbreviations: Adenosine diphosphate (ADP), adenosine triphosphate (ATP), molecular dynamics (MD), multi-layer perceptron (MLP), representation (repr.), features (feat.).

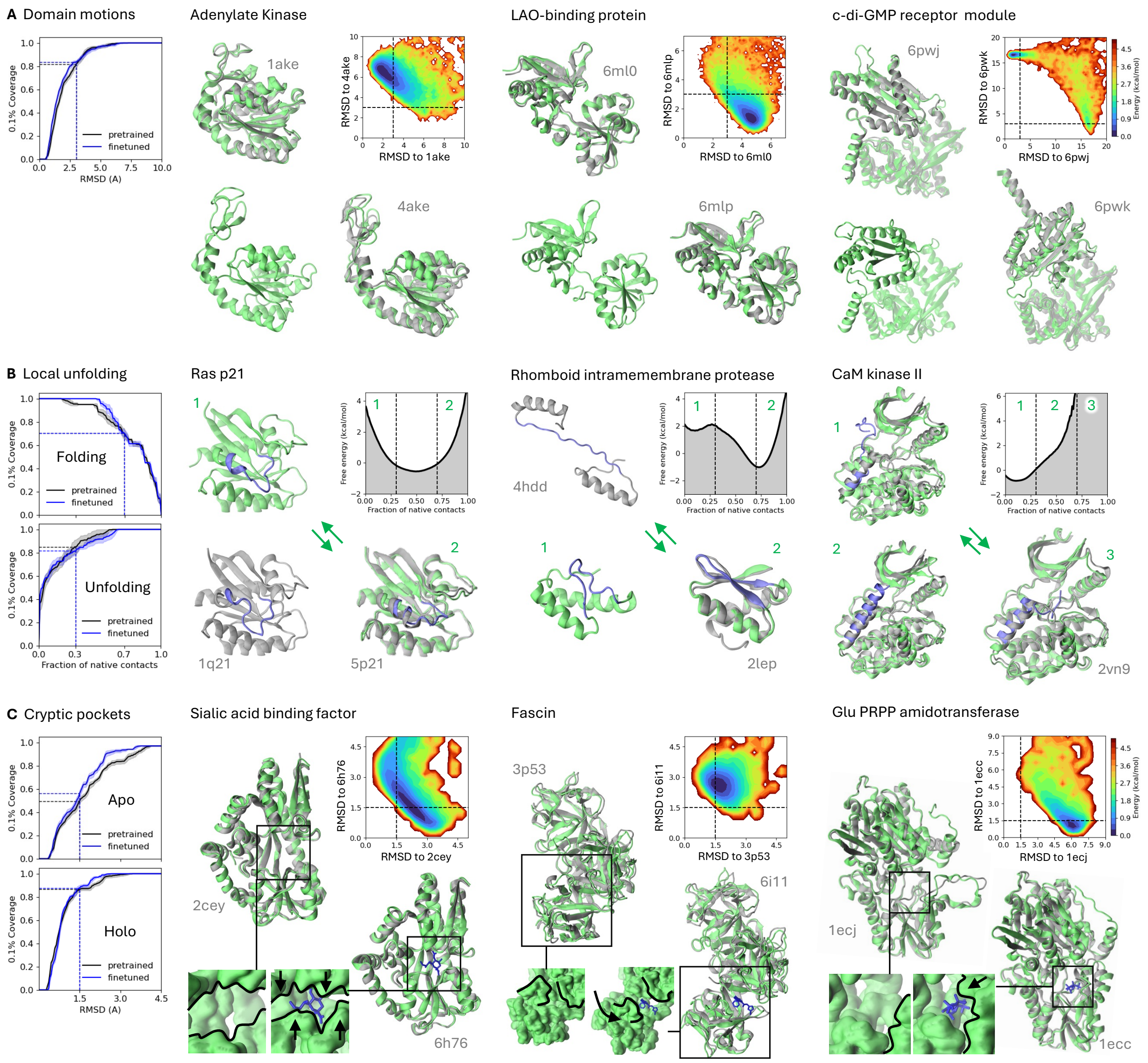

Figure 2: **BioEmu samples functionally distinct protein conformations.** (**A**) Large-scale domain motions. (**B**) Local unfolding or unbinding of parts of the protein. (**C**) Formation of cryptic binding pockets that are not present in the apo ground state. Left column: coverage of pretrained and fine-tuned BioEmu models, defined as the percentage of reference structures that are sampled by at least 0.1% of samples within a given distance. Global and local root mean square deviation (RMSD) are used for domain motions and cryptic pocket formation benchmarks, respectively, and fraction of native contacts (FNC) for local unfolding. Successful coverage of reference states is defined by probability density left of and below the dashed lines in A,C and outside the dashed lines in B. BioEmu sampled structures: green, PDB structures: grey, key secondary structure elements: blue. Abbreviations: Lysine arginine ornithine (LAO), guanosine monophosphate (GMP), calcium-calmodulin dependent (CaM). See Table S4 for 12-letter PDB codes and original citations.

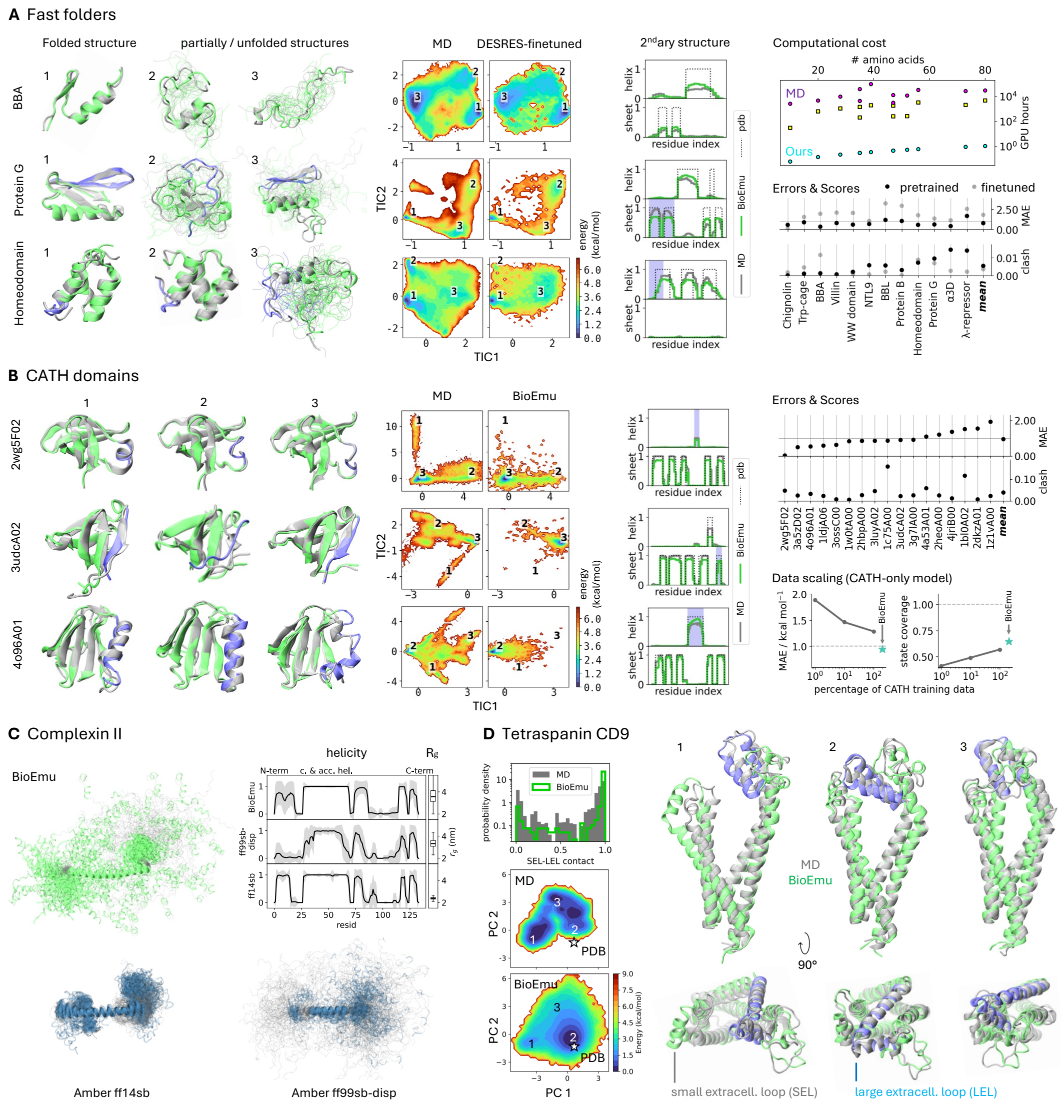

Figure 3: **BioEmu achieves fast emulation of all-atom molecular dynamics (MD) equilibrium distributions.** (**A**) DESRES fast-folding proteins. Left to right: Representative structures (green: model, grey: MD, blue: regions of interest). Free energy surfaces over slowest time-lagged independent components (TIC) (*54*). Secondary structure propensities. Computational cost for MD (magenta: full DESRES data; yellow: single folding-unfolding roundtrip) versus model (cyan: 10k samples). Mean absolute error (MAE) of free energy differences and fraction of unphysical model samples. (**B**) CATH domains. Structures, free energy surfaces and errors as in A. Bottom right: data scaling for specialized CATH-only model with free energy MAE and state coverage as function of training data. Cyan star: BioEmu. (**C**) Complexin II: Structures, helix content and radius of gyration (R_g) compared between BioEmu and two all-atom force fields. (**D**) Tetraspanin CD9 results from BioEmu and MD (*37*). Open-close transition, as histogram in log scale of small and large extracellular loop (SEL-LEL) contacts, as defined in (*37*). 2D-principal component (PC) analysis of $\exp(-d_{ij})$ of Ca-Ca distances d_{ij} between SEL and LEL. Star marks experimental structure (6k4j).

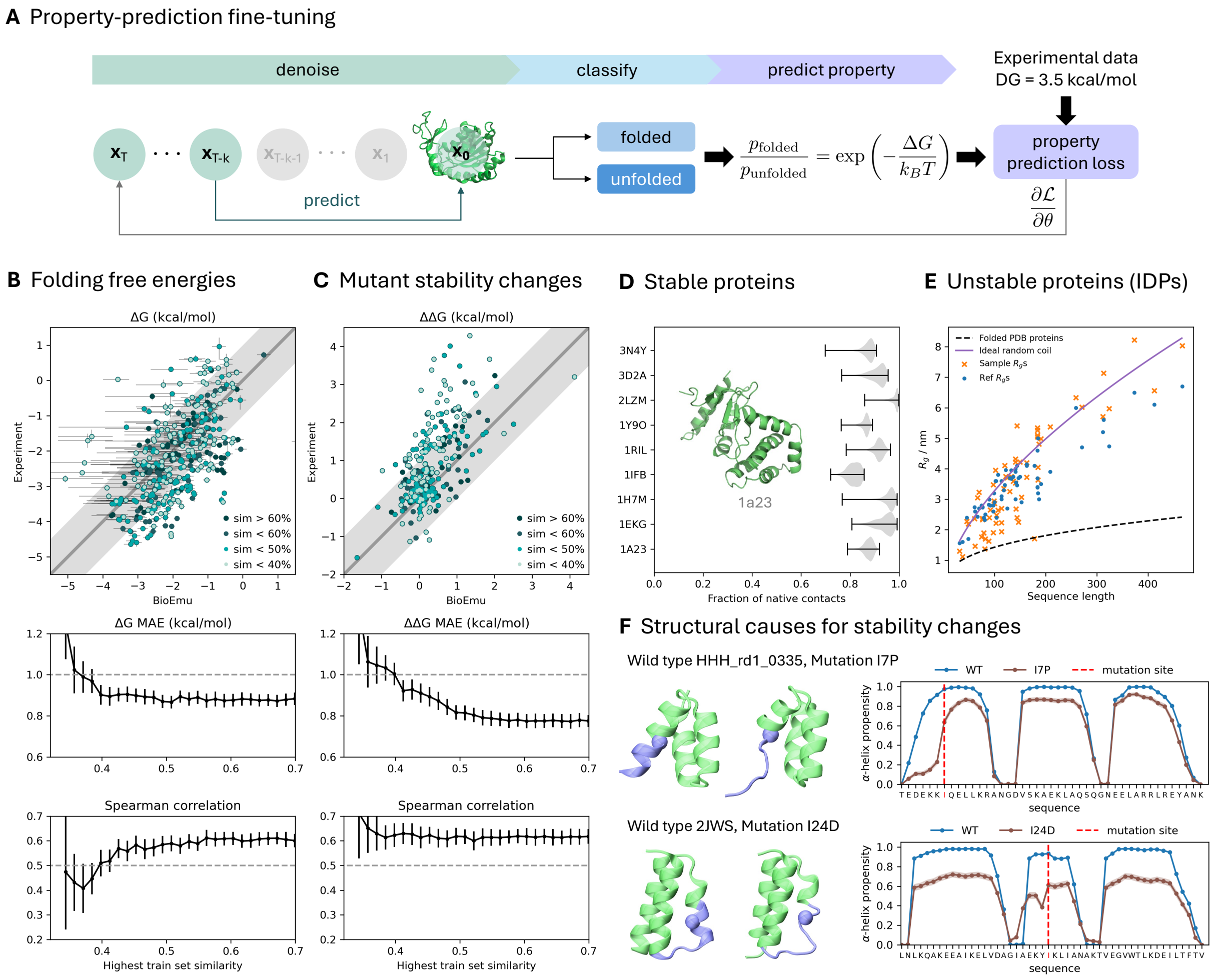

Figure 4: **Prediction of experimentally measured protein stabilities.** (**A**) Property-prediction fine-tuning algorithm for fine-tuning a pre-trained diffusion model to match experimentally measurable properties such as the protein stability. (**B**) Comparison of experimental measurements of folding free energies (*21*) with model predictions, generated by direct sampling and counting of folded and unfolded states for test proteins, the mean absolute error (MAE) and the Spearman correlation coefficient as a function of the sequence similarity between test and train proteins. (**C**) Same as B for the change in folding stability upon point mutation. (**D**) Validation that very stable proteins that are not included in the MEGAscale experimental dataset are consistently predicted as folded. (**E**) Validation that intrinsically disordered proteins (IDPs) reported in (*39*) and (*41*) are predicted as unfolded. Radius of gyration (R_g) is compared between model (orange crosses) experimental measurement (blue dots) and Flory scaling (*55*). (**F**) Analysis of the effect of two destabilizing mutants on the folded structures as predicted by the model: HHH_rd1_0335 with mutation I7P and 2JWS with mutation I24D. See Table S4 for 12-letter PDB codes and original citations.

References and Notes

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

2. J. Abramson, *et al.*, Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* **630**, 493–500 (2024).

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

4. R. Krishna, *et al.*, Generalized biomolecular modeling and design with RoseTTAFold All-Atom. *Science* **384**, eadl2528 (2024).

5. F. Ritort, Single-molecule experiments in biological physics: Methods and applications. *J. Phys.: Condens. Matter* **18**, R531–R583 (2006).

6. X.-C. Bai, G. McMullan, S. H. Scheres, How cryo-EM is revolutionizing structural biology. *Trends Biochem. Sci.* **40**, 49–57 (2015).

7. K. Lindorff-Larsen, S. Piana, R. O. Dror, D. E. Shaw, How Fast-Folding Proteins Fold. *Science* **334**, 517–520 (2011).

8. N. Plattner, S. Doerr, G. D. Fabritiis, F. Noé, Complete Protein–Protein Association Kinetics in Atomic Detail Revealed by Molecular Dynamics Simulations and Markov Modelling. *Nat. Chem.* **9**, 1005 (2017).

9. J. Wang, *et al.*, Machine Learning of coarse-grained Molecular Dynamics Force Fields. *ACS Cent. Sci.* **5**, 755–767 (2019).

10. N. E. Charron, *et al.*, Navigating protein landscapes with a machine-learned transferable coarse-grained model. *arXiv:2310.18278* (2023).

11. F. Noé, S. Olsson, J. Köhler, H. Wu, Boltzmann Generators - Sampling Equilibrium States of Many-Body Systems with Deep Learning. *Science* **365**, eaaw1147 (2019).

12. S. Zheng, *et al.*, Predicting equilibrium distributions for molecular systems with deep learning. *Nat. Mach. Intell.* **6**, 558–567 (2024).

13. B. Jing, B. Berger, T. Jaakkola, AlphaFold meets flow matching for generating protein ensembles. *arXiv:2402.04845* (2024).

14. Z. Qiao, W. Nie, A. Vahdat, T. F. M. III, A. Anandkumar, State-specific protein–ligand complex structure prediction with a multiscale deep generative model. *Nat. Mach. Intell.* **6**, 195–208 (2024).

15. H. K. Wayment-Steele, *et al.*, Predicting multiple conformations via sequence clustering and AlphaFold2. *Nature* **625**, 832–839 (2024).

16. P. Bryant, F. Noé, Structure prediction of alternative protein conformations. *Nat. Commun.* **15**, 7328 (2024).

17. B. P. Vani, A. Aranganathan, D. Wang, P. Tiwary, AlphaFold2-RAVE: From Sequence to Boltzmann Ranking. *J. Chem. Theory Comput.* **19**, 4351–4354 (2023).

18. A. Aranganathan, X. Gu, D. Wang, B. Vani, P. Tiwary, Modeling Boltzmann weighted structural ensembles of proteins using AI based methods. *Curr. Opin. Struct. Biol.* **91**, 103000 (2025).

19. R. Amaro, J. Åqvist, I. e. a. Bahar, The need to implement FAIR principles in biomolecular simulations. *Nat. Methods* **22**, 641–645 (2025).

20. J.-H. Prinz, *et al.*, Markov models of molecular kinetics: Generation and Validation. *J. Chem. Phys.* **134**, 174105 (2011).

21. K. Tsuboyama, *et al.*, Mega-scale experimental analysis of protein folding stability in biology and design. *Nature* **620**, 434–444 (2023).

22. D. Del Alamo, D. Sala, H. S. Mchaourab, J. Meiler, Sampling alternative conformational states of transporters and receptors with AlphaFold2. *Elife* **11**, e75751 (2022).

23. H. Y. Aviram, M. Pirchi, H. Mazal, G. Haran, Direct observation of ultrafast large-scale dynamics of an enzyme under turnover conditions. *Proc. Natl. Acad. Sci. USA* **115**, 3243–3248 (2018).

24. F. B. Fromowitz, *et al.*, Ras p21 expression in the progression of breast cancer. *Hum. Path.* **18**, 1268–1275 (1987).

25. J. B. Greisman, *et al.*, Discovery and Validation of the Binding Poses of Allosteric Fragment Hits to Protein Tyrosine Phosphatase 1b: From Molecular Dynamics Simulations to X-ray Crystallography. *J. Chem. Inf. Model.* **63**, 2644–2650 (2023).

26. M. I. Zimmerman, *et al.*, SARS-CoV-2 Simulations Go Exascale to Predict Dramatic Spike Opening and Cryptic Pockets across the Proteome. *Nat. Chem.* **13**, 651–659 (2021).

27. T. J. Lane, D. Shukla, K. A. Beauchamp, V. S. Pande, To Milliseconds and beyond: Challenges in the Simulation of Protein Folding. *Curr. Opin. Struct. Biol.* **23**, 58–65 (2013).

28. D. E. Shaw, *et al.*, Atomic-Level Characterization of the Structural Dynamics of Proteins. *Science* **330**, 341–346 (2010).

29. J. D. Chodera, F. Noé, Markov State Models of Biomolecular Conformational Dynamics. *Curr. Opin. Struct. Biol.* **25**, 135–144 (2014).

30. R. B. Best, J. Mittal, Free-energy landscape of the GB1 hairpin in all-atom explicit solvent simulations with different force fields: Similarities and differences. *Proteins* **79**, 1318–1328 (2011).

31. D. F. Hahn, V. Gapsys, B. L. de Groot, D. L. Mobley, G. Tresadern, Current state of open source force fields in protein-ligand binding affinity predictions. *J. Chem. Inf. Model.* **64**, 5063–5076 (2024).

32. A. Laio, M. Parrinello, Escaping Free-Energy Minima. *Proc. Natl. Acad. Sci. USA* **99** (20), 12562–12566 (2002).

33. P. Robustelli, S. Piana, D. E. Shaw, Developing a Molecular Dynamics Force Field for Both Folded and Disordered Protein States. *Proc. Natl. Acad. Sci. USA* **115**, E4758–E4766 (2018).

34. I. Sillitoe, *et al.*, CATH: Increased Structural Coverage of Functional Space. *Nucleic Acids Res.* **49**, D266–D273 (2021).

35. J. Malsam, *et al.*, Complexin Suppresses Spontaneous Exocytosis by Capturing the Membrane-Proximal Regions of VAMP2 and SNAP25. *Cell Rep.* **32**, 107926 (2020).

36. Q. Zhou, *et al.*, The Primed SNARE–Complexin–Synaptotagmin Complex for Neuronal Exocytosis. *Nature* **548**, 420–425 (2017).

37. R. Umeda, *et al.*, Structural insights into tetraspanin CD9 function. *Nat. Commun.* **11**, 1606 (2020).

38. R. Nikam, A. Kulandaisamy, K. Harini, D. Sharma, M. M. Gromiha, ProThermDB: Thermodynamic database for proteins and mutants revisited after 15 years. *Nucleic Acids Res.* **49**, D420–D424 (2021).

39. G. Tesei, *et al.*, Conformational ensembles of the human intrinsically disordered proteome. *Nature* **626**, 897–904 (2024).

40. G. Tesei, K. Lindorff-Larsen, Improved predictions of phase behaviour of intrinsically disordered proteins by tuning the interaction range. *Open Research Europe* **2**, 94 (2023).

41. J. Zhu, *et al.*, Precise Generation of Conformational Ensembles for Intrinsically Disordered Proteins via Fine-tuned Diffusion Models. *bioRxiv* (2024), doi:10.1101/2024.05.05.592611.

42. J. Ouyang-Zhang, D. Diaz, A. Klivans, P. Krähenbühl, Predicting a protein's stability under a million mutations. *Adv. Neural Inf. Process. Syst.* **36**, 76229–76247 (2024).

43. M. Cagiada, S. Ovchinnikov, K. Lindorff-Larsen, Predicting absolute protein folding stability using generative models. *Prot. Sci.* **34**, e5233 (2025).

44. P. Notin, *et al.*, ProteinGym: Large-scale benchmarks for protein fitness prediction and design. *Adv. Neural Inf. Process. Syst.* **36**, 64331–64379 (2024).

45. T. Widatalla, R. Rafailov, B. Hie, Aligning protein generative models with experimental fitness via Direct Preference Optimization. *bioRxiv* (2024), doi:10.1101/2024.05.20.595026.

46. Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, M. Le, Flow Matching for Generative Modeling. *arXiv:2210.02747* (2022).

47. J. Yim, *et al.*, Fast protein backbone generation with SE(3) flow matching. *arXiv:2310.05297* (2023).

48. W. G. Noid, *et al.*, The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models. *J. Chem. Phys.* **128**, 244114 (2008).

49. L. Klein, *et al.*, Timewarp: transferable acceleration of molecular dynamics by learning time-coarsened dynamics. *Adv. Neural Inf. Process. Syst.* **37**, 52863–52883 (2023).

50. M. Schreiner, O. Winther, S. Olsson, Implicit transfer operator learning: Multiple time-resolution models for molecular dynamics. *Adv. Neural Inf. Process. Syst.* **36**, 36449–36462 (2024).

51. T. Fröhlking, M. Bernetti, N. Calonaci, G. Bussi, Toward empirical force fields that match experimental observables. *J. Chem. Phys.* **152**, 230902 (2020).

52. J. Gawlikowski, *et al.*, A survey of uncertainty in deep neural networks. *Artif. Intell. Rev.* **56**, 1513–1589 (2023).

53. H. Sidky, W. Chen, A. L. Ferguson, Machine learning for collective variable discovery and enhanced sampling in biomolecular simulation. *Mol. Phys.* **118**, e1737742 (2020).

54. G. Perez-Hernandez, F. Paul, T. Giorgino, G. D Fabritiis, F. Noé, Identification of slow molecular order parameters for Markov model construction. *J. Chem. Phys.* **139**, 015102 (2013).

55. H. Hofmann, *et al.*, Polymer scaling laws of unfolded and intrinsically disordered proteins quantified with single-molecule spectroscopy. *Proc. Natl. Acad. Sci. USA* **109**, 16155–16160 (2012).

56. BioEmu training data: Octapeptides, `https://doi.org/10.5281/zenodo.15641199`.

57. BioEmu training data: CATH, `https://doi.org/10.5281/zenodo.15629740`.

58. BioEmu training data: MEGAsim, `https://doi.org/10.5281/zenodo.15641184`.

59. G. M. Visani, W. Galvin, M. Pun, A. Nourmohammad, H-Packer: Holographic Rotationally Equivariant Convolutional Neural Network for Protein Side-Chain Packing, in *Proceedings of the 18th Machine Learning in Computational Biology meeting*, D. A. Knowles, S. Mostafavi, Eds., vol. 240 of *Proc. Mach. Learn. Res.* (2024), pp. 230–249.

60. P. Eastman, *et al.*, OpenMM 7: Rapid Development of High Performance Algorithms for Molecular Dynamics. *PLOS Comput. Biol.* **13**, e1005659 (2017).

61. Colabfold Github, Github, `https://github.com/microsoft/bioemu-benchmarks`.

62. BioEmu-1 code, model and data, `https://doi.org/10.5281/zenodo.15672282`.

63. M. Steinegger, J. Söding, MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.* **35**, 1026–1028 (2017).

64. M. van Kempen, *et al.*, Fast and accurate protein structure search with Foldseek. *Nat. Biotechnol.* **42**, 243–246 (2024).

65. G. Huguet, *et al.*, Sequence-Augmented SE(3)-Flow Matching For Conditional Protein Backbone Generation. *arXiv:2405.20313* (2024).

66. M. J. Abraham, *et al.*, GROMACS: High Performance Molecular Simulations through Multi-Level Parallelism from Laptops to Supercomputers. *SoftwareX* **1–2**, 19–25 (2015).

67. W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, M. L. Klein, Comparison of Simple Potential Functions for Simulating Liquid Water. *J. Chem. Phys.* **79**, 926–935 (1983).

68. C. W. Hopkins, S. Le Grand, R. C. Walker, A. E. Roitberg, Long-Time-Step Molecular Dynamics through Hydrogen Mass Repartitioning. *J. Chem. Theory Comput.* **11**, 1864–1874 (2015).

69. K. Lindorff-Larsen, *et al.*, Improved Side-chain Torsion Potentials for the Amber ff99SB Protein Force Field. *Proteins* **78**, 1950–1958 (2010).

70. E. Hruska, J. R. Abella, F. Nüske, L. E. Kavraki, C. Clementi, Quantitative Comparison of Adaptive Sampling Methods for Protein Dynamics. *J. Chem. Phys.* **149**, 244119 (2018).

71. M. K. Scherer, *et al.*, PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models. *J. Chem. Theory Comput.* **11**, 5525–5542 (2015).

72. J. A. Maier, *et al.*, ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. *J. Chem. Theory Comput.* **11**, 3696–3713 (2015).

73. S. Piana, K. Lindorff-Larsen, D. E. Shaw, How Robust Are Protein Folding Simulations with Respect to Force Field Parameterization? *Biophys. J.* **100**, L47–L49 (2011).

74. S. M. Hanson, *et al.*, What Makes a Kinase Promiscuous for Inhibitors? *Cell Chem. Biol.* **26**, 390–399.e5 (2019).

75. S. Chen, *et al.*, The Dynamic Conformational Landscape of the Protein Methyltransferase SETD8. *eLife* **8**, e45403 (2019).

76. Y. Duan, *et al.*, A Point-charge Force Field for Molecular Mechanics Simulations of Proteins Based on Condensed-phase Quantum Mechanical Calculations. *J. Comput. Chem.* **24**, 1999–2012 (2003).

77. E. C. Thomson, *et al.*, Circulating SARS-CoV-2 Spike N439K Variants Maintain Fitness While Evading Antibody-Mediated Immunity. *Cell* **184**, 1171–1187.e20 (2021).

78. E. T. Abualrous, *et al.*, MHC-II Dynamics Are Maintained in HLA-DR Allotypes to Ensure Catalyzed Peptide Exchange. *Nat. Chem. Biol.* **19**, 1196–1204 (2023).

79. V. Hornak, *et al.*, Comparison of Multiple Amber Force Fields and Development of Improved Protein Backbone Parameters. *Proteins* **65**, 712–725 (2006).

80. M. Mirdita, *et al.*, ColabFold: Making protein folding accessible to all. *Nat. Meth.* **19**, 679–682 (2022).

81. J. Yim, *et al.*, SE(3) diffusion model with application to protein backbone generation. *arXiv:2302.02277* (2023).

82. G. Ahdritz, *et al.*, OpenFold: Retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization. *Nat. Meth.* **21**, 1–11 (2024).

83. A. Nichol, P. Dhariwal, Improved Denoising Diffusion Probabilistic Models. *arXiv:2102.09672* (2021).

84. V. D. Bortoli, *et al.*, Riemannian Score-Based Generative Modelling. *arXiv:2202.02763* (2022).

85. Y. Song, *et al.*, Score-based generative modeling through stochastic differential equations. *arXiv:2011.13456* (2020).

86. T. Karras, M. Aittala, T. Aila, S. Laine, Elucidating the design space of diffusion-based generative models. *Adv. Neural. Inf. Process. Syst.* **35**, 26565–26577 (2022).

87. C. Lu, *et al.*, DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. *Adv. Neural. Inf. Process. Syst.* **35**, 5775–5787 (2022).

88. I. Barrio-Hernandez, *et al.*, Clustering predicted structures at the scale of the known protein universe. *Nature* **622**, 637–645 (2023).

89. C. Wehmeyer, *et al.*, Introduction to Markov State Modeling with the PyEMMA Software [Article v1.0]. *LiveCoMS* **1**, 5965 (2018).

90. M. Hoffmann, *et al.*, Deeptime: A Python Library for Machine Learning Dynamical Models from Time Series Data. *Mach. Learn.: Sci. Technol.* **3**, 015009 (2022).

91. R. B. Best, G. Hummer, W. A. Eaton, Native contacts determine protein folding mechanisms in atomistic simulations. *Proc. Natl. Acad. Sci. USA* **110**, 17874–17879 (2013).

92. C. Luo, Understanding diffusion models: A unified perspective. *arXiv:2208.11970* (2022).

93. C. Domingo-Enrich, M. Drozdzal, B. Karrer, R. T. Chen, Adjoint Matching: Fine-tuning Flow and Diffusion Generative Models with Memoryless Stochastic Optimal Control. *arXiv:2409.08861* (2024).

94. J. Xu, *et al.*, Imagereward: Learning and evaluating human preferences for text-to-image generation. *Adv. Neural. Inf. Process. Syst.* **37**, 15903–15935 (2023).

95. K. Clark, P. Vicol, K. Swersky, D. J. Fleet, Directly fine-tuning diffusion models on differentiable rewards. *arXiv:2309.17400* (2023).

96. P. Cimermancic, *et al.*, CryptoSite: Expanding the druggable proteome by characterization and prediction of cryptic binding sites. *J. Mol. Biol.* **428**, 709–719 (2016).

97. A. Meller, S. Bhakat, S. Solieva, G. R. Bowman, Accelerating cryptic pocket discovery using AlphaFold. *J. Chem. Theory Comput.* **19**, 4355–4363 (2023).

98. D. Chakravarty, L. L. Porter, AlphaFold2 fails to predict protein fold switching. *Protein Sci.* **31**, e4353 (2022).

99. J. M. Dana, *et al.*, SIFTS: updated Structure Integration with Function, Taxonomy and Sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. *Nucleic Acids Res.* **47**, D482–D489 (2019).

100. C. W. Müller, G. E. Schulz, Structure of the complex between adenylate kinase from Escherichia coli and the inhibitor Ap5A refined at 1.9 Å resolution: A model for a catalytic transition state. *J. Mol. Biol.* **224**, 159–177 (1992).

101. C. H. Chan, C. C. Wilbanks, G. I. Makhatadze, K. B. Wong, Electrostatic contribution of surface charge residues to the stability of a thermophilic protein: benchmarking experimental and predicted pKa values. *PLoS One* **7**, e30296 (2012).

102. C. W. Müller, G. J. Schlauderer, J. Reinstein, G. E. Schulz, Adenylate kinase motions during catalysis: an energetic counterweight balancing substrate binding. *Structure* **4**, 147–156 (1996).

103. S. Ahmad, M. Z. Kamal, R. Sankaranarayanan, N. M. Rao, Thermostable Bacillus subtilis lipases: in vitro evolution and structural insight. *J. Mol. Biol.* **381**, 324–340 (2008).

104. R. Vergara, *et al.*, The interplay of protein-ligand and water-mediated interactions shape affinity and selectivity in the LAO binding protein. *FEBS J.* **287**, 763–782 (2020).

105. L. H. Weaver, B. W. Matthews, Structure of bacteriophage T4 lysozyme refined at 1.7 Å resolution. *J. Mol. Biol.* **193**, 189–199 (1987).

106. A. Corazza, *et al.*, Structure, conformational stability, and enzymatic properties of acylphosphatase from the hyperthermophile Sulfolobus solfataricus. *Proteins* **62**, 64–79 (2006).

107. G. Kitts, *et al.*, A Conserved Regulatory Circuit Controls Large Adhesins in Vibrio cholerae. *mBio* **10**, e02822–19 (2019).

108. K. Ishikawa, *et al.*, Crystal structure of ribonuclease H from Thermus thermophilus HB8 refined at 2.8 Å resolution. *J. Mol. Biol.* **230**, 529–542 (1993).

109. J. C. Sacchettini, J. I. Gordon, L. J. Banaszak, Refined apoprotein structure of rat intestinal fatty acid binding protein produced in Escherichia coli. *Proc. Natl. Acad. Sci. USA* **86**, 7736–7740 (1989).

110. L. A. Tong, A. M. de Vos, M. V. Milburn, S. H. Kim, Crystal structures at 2.2 Å resolution of the catalytic domains of normal ras protein and an oncogenic mutant complexed with GDP. *J. Mol. Biol.* **217**, 503–516 (1991).

111. Y.W., Chen, M. Bycroft, K. B. Wong, Crystal Structure of Ribosomal Protein L30E from the Extreme Thermophile Thermocccus Celer: Thermal Stability and RNA Binding. *Biochem.* **42**, 2857–2865 (2003).

112. E. F. Pai, *et al.*, Refined crystal structure of the triphosphate conformation of H-ras p21 at 1.35 Åresolution: implications for the mechanism of GTP hydrolysis. *EMBO J.* **9**, 2351–2359 (1990).

113. S. Dhe-Paganon, R. Shigeta, Y. I. Chi, M. Ristow, S. E. Shoelson, Crystal structure of human frataxin. *J. Biol. Chem.* **275**, 30753–30756 (2000).

114. C. Lazareno-Saez, E. Arutyunova, N. Coquelle, M. J. Lemieux, Domain swapping in the cytoplasmic domain of the Escherichia coli rhomboid protease. *J. Mol. Biol.* **425**, 1127–1142 (2013).

115. H. J. Schirra, *et al.*, Structure of reduced DsbA from Escherichia coli in solution. *Biochem.* **37**, 6263–6276 (1998).

116. A. R. Sherratt, D. R. Blais, H. Ghasriani, J. P. Pezacki, N. K. Goto, Activity-Based Protein Profiling of the Escherichia coli GlpG Rhomboid Protein Delineates the Catalytic Core. *Biochem.* **51**, 7794–7803 (2012).

117. Y. He, Y. Chen, P. Alexander, P. N. Bryan, J. Orban, NMR structures of two designed proteins with high sequence identity but different fold and function. *Proc. Natl. Acad. Sci. USA* **105**, 14412–14417 (2008).

118. P. Rellos, *et al.*, Structure of the Camkiidelta/Calmodulin Complex Reveals the Molecular Mechanism of Camkii Kinase Activation. *PLoS Biol.* **8** (426) (2010).

119. V. Neviani, *Unravelling the molecular mechanisms of tetraspanin CD9*, Ph.D. thesis, Utrecht University, the Netherlands (2019).

120. A. Muller, *et al.*, Conservation of Structure and Mechanism in Primary and Secondary Transporters Exemplified by Siap, a Sialic Acid Binding Virulence Factor from Haemophilus Influenzae. *J. Biol. Chem.* **281**, 22212 (2006).

121. W. Oosterheert, *et al.*, Implications for tetraspanin-enriched microdomain assembly based on structures of CD9 with EWI-F. *Life Sci. Alliance* **3**, e202000883 (2020).

122. J. F. Darby, *et al.*, Water Networks Can Determine the Affinity of Ligand Binding to Proteins. *J. Am. Chem. Soc.* **141**, 15818–15826 (2019).

123. S. Jansen, *et al.*, Mechanism of actin filament bundling by fascin. *J. Biol. Chem.* **286**, 30087–30096 (2011).

124. S. Francis, *et al.*, Structure-based design, synthesis and biological evaluation of a novel series of isoquinolone and pyrazolo[4,3-c]pyridine inhibitors of fascin 1 as potential anti-metastatic agents. *Bioorg. Med. Chem. Lett.* **29**, 1023–1029 (2019).

125. C. R. Muchmore, J. M. Krahn, J. H. Kim, H. Zalkin, J. L. Smith, Crystal structure of glutamine phosphoribosylpyrophosphate amidotransferase from Escherichia coli. *Protein Sci.* **7**, 39–51 (1998).

126. J. M. Krahn, *et al.*, Coupled formation of an amidotransferase interdomain ammonia channel and a phosphoribosyltransferase active site. *Biochem.* **36**, 11061–11068 (1997).

Acknowledgments

We are indebted to the entire Microsoft Research AI for Science team, and thank in particular the following individuals for valuable support and discussions: Chang Liu, Peiran Jin, Tie-Yan Liu, Christopher M. Bishop, Bonnie Kruft, Jonas Köhler, Thijs Vogels, Marwin Segler, Rianne van den Berg, Marco Federici, Stratis Markou, Maik Riechert, Lixin Sun. We thank Martin Steinegger (Seoul National University) for his invaluable help in publishing the BioEmu inference code on ColabFold. Furthermore, we thank colleagues from FU Berlin for valuable discussions and advice, in particular Nicholas E. Charron, Katarina Elez, and Aldo Sayeg Pasos Trejo. Additionally, we thank Gregory R. Bowman and Anthony M. Trent (University of Pennsylvania) and Sukrit Singh (MSKCC) for their help with Folding@Home data and Nuria Plattner (University of Basel) for her support.

Funding: All work was funded by Microsoft.

Author contributions: Data curation, generation and benchmarks: SL, TH, JJL, IZ, YC, CC, FN. Software: SL, TH, JJL, MG, YX, AYKF, VGS, OA, BV, YC, SY, AS, JN, FB, VS, AC, JY, ML, HS, RT. Methodology: SL, TH, JJL, MG, YX, AYKF, VGS, OA, BV, IZ, YC, SY, AEF, AS, JN, FB, VS, AC, JY, ML, YS, SZ, RT, FN. Results generation, analysis, validation and visualization: SL, TH, JJL, MG, YX, AYKF, VGS, OA, BV, IZ, YC, AEF, RT, CC, FN. Writing: SL, TH, JJL, MG, YX, AYKF, VGS, OA, IZ, YC, RT, CC, FN. Project administration: RS, UM. Software lead: SL, BV. Simulation data lead: TH. Project lead + conceptualization: FN.

Competing interests: A US patent application for the algorithms and training methods described in this paper has been filed under the title “Prediction of protein structure ensembles”. The following authors are listed as inventors in the patent application: Sarah Lewis, Tim Hempel, José Jiménez-Luna, Michael Gastegger, Yu Xie, Andrew Y. K. Foong, Victor García Satorras, Osama Abdin, Bastiaan Veeling, Soojung Yang, Arne Schneuing, and Frank Noé.

Data and materials availability: As currently no standard database for MD data exists (see (*19*) for a discussion), MD data was either downloaded from various sources or generated for this work. Please see Table S1 and materials and methods for a description of all datasets with download links. The MD training data produced as part of this study is available at Zenodo under the permissive Community Data License Agreement CDLA 2.0, see references (*56–58*). The D. E. SHAW Research (DESRES) data for fast-folding proteins, as well the model trained on them, cannot be redistributed according to the DESRES data license. Researchers are requested to contact DESRES directly for obtaining this dataset and a license. The BioEmu model and inference code are available under MIT license at `https://github.com/microsoft/bioemu` and can be installed using `pip install bioemu` for scalable execution on Linux desktops or clusters. Internal training code is not published as it is specialized to a Microsoft-internal Azure cloud environment. To facilitate the development of a community training code for BioEmu, the pretraining AFDB cluster definition and a sample implementation of PPFT are available in the same GitHub repository. A simple way to test the model is available through `https://github.com/sokrypton/ColabFold`. Sidechain reconstruction and relaxation of generated structures for downstream analyses are also provided via the hpacker (*59*) and OpenMM (*60*) packages, respectively. Code for benchmarking BioEmu or other models for their ability to sample multiple protein conformations, free energy surfaces and folding free energies (Figs. 2, 3B, 4A, and Suppl. Figs. S1, S2, S3, S4), is provided under MIT license at `https://github.com/microsoft/bioemu-benchmarks` (*61*). The version of the BioEmu code and model used in this paper, as well as the BioEmu model samples used for generating the figures, has been archived at (*62*).

Supplementary materials

Supplementary Materials for
Scalable emulation of protein equilibrium ensembles with generative deep learning

Sarah Lewis[1†], Tim Hempel[1†], José Jiménez-Luna[1†], Michael Gastegger[1†], Yu Xie[1†], Andrew Y. K. Foong[1†], Victor García Satorras[1†], Osama Abdin[1†], Bastiaan S. Veeling[1†], Iryna Zaporozhets[1,2], Yaoyi Chen[1,2], Soojung Yang[1], Adam E. Foster[1], Arne Schneuing[1], Jigyasa Nigam[1], Federico Barbero[1], Vincent Stimper[1], Andrew Campbell[1], Jason Yim[1], Marten Lienen[1], Yu Shi[1], Shuxin Zheng[1], Hannes Schulz[1], Usman Munir[1], Roberto Sordillo[1], Ryota Tomioka[1], Cecilia Clementi[1,2,3], Frank Noé[1,2,3,*]

[1]AI for Science, Microsoft Research

[2]Freie Universität Berlin, Berlin 14195, Germany

[3]Rice University, Houston, TX 77005, USA

[*]Corresponding author. Email: franknoe@microsoft.com

[†]These authors contributed equally to this work.

This PDF file includes:

Materials and Methods

Figures S1 to S12

Tables S1 to S3

Algorithms 1 to 2

Materials and Methods

Data

AlphaFoldDB processing An AlphaFold database (AFDB) snapshot was downloaded in July 2024 and preprocessed for model pre-training (Fig. 1D,E). The aim of this preprocessing is to identify sets of similar sequences with heterogeneous predicted structures, and this is accomplished through a series of steps:

1. We used `mmseqs` (*63*) to cluster all sequences at 80% sequence identity and 70% coverage, resulting in a set containing more than 93 million clusters.

2. To reduce the sequence clusters to a representative set, we clustered the centroids of these clusters at 30% sequence identity and discarded all but the largest 80%-sequence-identity cluster within each 30%-sequence-identity group. The result was a set of sequence clusters with 80% sequence similarity within each cluster and at most 30% sequence similarity between the centroids of different clusters.

3. We discarded sequence clusters with fewer than 10 members, leaving roughly 1.4 million sequence clusters.

4. We performed structure-based clustering within each sequence cluster, using FOLDSEEK (*64*) (version 9.427df8a) with a sequence identity threshold of 70% at 90% coverage.

5. We discarded everything except the representative member of each structure cluster, leaving a set of sequence clusters, each containing a few structure representatives.

6. We discarded sequence clusters with only one structure representative and those where all the structure representatives were disordered (defined as being composed of more than 50% coil in their secondary structure).

7. To account for structural heterogeneity that was incorrectly flagged due to missing regions in structure representatives, we performed structural alignments in sequence-aligned regions of proteins and discarded structure representatives with a TM-score greater than 0.9 to another structure representative, as computed by FOLDSEEK.

8. Similarly to (*65*), we removed sequence clusters lacking at least one structure with a pLDDT greater than 80, and a pLDDT standard deviation lower than 15 across residues

After running this pipeline, we had ~50k sequence clusters with structural diversity. We utilize the structural diversity to generate augmented data during the pre-training phase (see "Pre-training on AFDB").

Protein Data Bank processing A snapshot of the PDB was downloaded on Nov. 23rd, 2023, including all of the available asymmetric units in the mmCIF format. We use the `pdbecif` Python package (version 1.5) for mmCIF parsing and consider an entry for processing if the overall number of residues in the entry was below 2500 with a resolution below 9.5 Å, if a resolution value was available. All molecular entities per entry were separated according to their type (*i.e.*, polymer or non-polymer), discarding those associated with other nucleic acids (*e.g.*, RNAs, DNAs). Non-biologically relevant non-polymer entities (*e.g.*, solvents, ions) were further filtered out by a list provided in (*4*). Non-binding polymer chains were then kept on an entry basis if they contained standard amino acid types and depending on whether other non-binding chains corresponding to the same entity identifier had already been processed. So as to better capture ligand-binding conformational effects, all binding polymer chains with unique binders, defined by a distance threshold of 6 Å between any binder and protein atom, were kept.

Molecular Dynamics simulation data In the following, we list synthetic all-atom molecular dynamics (MD) data used in this article. An overview of all publicly available and in-house datasets is provided in Tab. S1. In-house datasets are described in detail in "In-house MD datasets", with our standard MD protocol specified in "MD simulation protocol". Publicly available datasets are listed under "Public MD datasets". As our model is for single protein chains and there were several MD simulations of multi-chain systems, we extracted individual protein chains and treated them independently. As a consequence, the effective cumulative simulation time for such multi-chain simulations is reported as a sum over all chains. For datasets curated from the literature, a reference is provided. Details for MD simulations generated specifically for this work are described in "In-house MD datasets".

MD simulation protocol We internally developed code specifically tailored towards running large MD production campaigns on Azure compute resources. Our code is based on OpenMM (*60*) as its compute engine, albeit setups are generated using OpenMM or GROMACS (*66*) as a backend, depending on the specific case. Unless specified otherwise, all MD simulations followed the protocol described below: We use explicit solvent and the tip3p water model (*67*), solvate the structures in a cubic box with 1 nm padding and 0.1 M NaCl buffer. The solvent is equilibrated with a harmonic constraint force on the solute heavy atoms for 0.1 ns under constant temperature and volume (NVT) followed by 0.9 ns under constant temperature and pressure (NPT). The constraint force is gradually removed over an additional 0.1 ns of simulation. During the equilibration phase, the integration time-step is set to 2 fs. Production runs are conducted in the NPT ensemble, using hydrogen mass repartitioning with a hydrogen mass of 4 amu (*68*), with hydrogen bond constraints, and an integration time-step of 4 fs. The temperature is set to 300 K and the pressure to 1 bar, unless noted otherwise.

In-house MD datasets Most simulation data described in this work was generated using T4-based (`NC4as_T4_v3`) Azure compute instances. Model training data is provided under the links below.

Octapeptides The octapeptide dataset consists of 1100 peptides of 8 amino acids length. The selection of systems had been previously described (see Ref. (*10*) for details about system selection and initial structure seeding procedures). We extend this dataset with longer trajectories to better represent equilibrium. For each system, 5 new trajectories with 1 μs length were generated using our in-house protocol ("MD simulation protocol"), using the same force field as in the original dataset (Amber ff99SB-ildn (*69*)), at 300K. The total simulation time amounts to 8 ms. Download link: `https://doi.org/10.5281/zenodo.15641199`.

CATH1 This data consists of 50 CATH domains as previously described in Ref. (*10*). We also extend this dataset, previously consisting of 4 x 0.5 μs trajectories per system, with longer MD trajectories to better represent long-timescale dynamic behavior of different protein domains. Trajectories between 1 and 5 μs in length using the amber ff99SB-ildn forcefield (*69*) were produced, totaling 100 μs per CATH domain. Data production was conducted using an adaptive sampling (*70*) scheme, where the first trajectory epoch was seeded from a reference PDB structure, while the following 2 epochs were seeded by extracting frames from previous epochs via a minRMSD clustering (*71*) approach. The total simulation time of the combined dataset (original and in-house) amounts to 5.2 ms. Download link: `https://doi.org/10.5281/zenodo.15629740`.

CATH2 The CATH2 dataset focuses on sequence coverage rather than overall simulation length. Similar to CATH1 and to the procedure described in (*10*), systems were selected from the CATH database (*34*) (version 4.3.0) by filtering out non-contiguous structures or sequences, non-standard amino acids, proteins with disulfide bonds, coil fractions above 50%, or a relative shape anisotropy ≥ 0.05. Only domains containing between 50 and 200 amino acids were selected, forming a set of $\sim$1100 domains. Out of those, for 1040 we could generate valid MD setups using our in-house protocol ("MD simulation protocol"). 1 μs trajectories for these domains were generated using the Amber99SB-ildn (*69*) forcefield, producing a total of approximately 39 μs per CATH domain, with the exact amount varying due to compute availability reasons. We used the same adaptive sampling strategy as in the CATH1 dataset, and 2 epochs of reseeding. The cumulative simulation time for the whole dataset is 41 ms.
Download link: `https://doi.org/10.5281/zenodo.15629740`.

MEGAsim The MEGAscale domain simulation dataset ("MEGAsim" in short) is dedicated to including folding-unfolding transitions in the training data. In total, it consists of extended simulations of 271 wild-types, and 1 μs simulations for each of the 22,118 point mutants, including single-residue insertion/deletions. The systems and mutants in our dataset were taken from the megascale measurements of protein domain stability via cDNA display proteolysis (*21*). To ensure that every sampled system had a corresponding experimental measurement of the folding free energy ΔG during the folding process, we focused on wild-types and mutants within the curated set ("`Dataset2_Dataset3`") of the reference publication. Our final dataset consists of a smaller subset of systems due to finite computational resources and several applied filters, detailed below.

For the wildtype dataset, we adapted the general simulation setup to efficiently sample in both the folded and unfolded states. The seeding structures included both the folded state as well as less structured decoys. Folded structures were obtained from the AF2 predictions available on the

Zenodo repository of Ref. (*21*). Unfolded (decoy) starting structures were obtained by simulated thermal denaturization in implicit solvent at 400K, followed by several rounds of adaptive sampling in explicit solvent at an elevated temperature. For both the equilibration and production phases, two force fields were used: amber ff14sb (*72*) and amber ff99SB-disp (*33*). In comparison to more traditional force fields like ff14sb, ff99sb-disp is specifically designed to model disordered proteins and does not overstabilize globular decoy structures (*33*). Even though generally reliable, we noticed that a99sb-disp can destabilize the native fold of a protein after extensive simulations. For those cases we relied on ff14sb to generate samples of the folded state.

To optimize compute efficiency, we chose a rhombic dodecahedron simulation box with a 1.5 nm padding for each individual seed. Equilibration was performed with 0.2 ns NVT and 0.6 ns NPT simulations, targeting 295K and 1 bar with a Langevin integrator and a 4 fs time step. Production simulations were performed for 1.5 μs per starting structure at 295K in the NVT ensemble and a 4 fs time step. Bond constraints and hydrogen mass were kept identical to section "MD simulation protocol"), and we discarded the first 500 ns of each trajectory to only consider the last 1 μs in the subsequent analysis. Post processing was carried out with the goal of obtaining a clear separation between folded and unfolded samples as well as minimizing the effect of mixing samples from two force fields. We used the fraction of native contacts (FNC) to define the relative foldedness of each MD frame, and built FNC histograms for all samples from each force field. While, theoretically, for two-state folding-unfolding transitions one can expect the a bimodal distribution, in practice it can be multimodal. However, we observed that the folded and unfolded states have well-defined density peaks in the FNC distribution, and thus performed a kernel density estimation. For each forcefield, we used the FNC with the lowest estimated density as the folding threshold.

For the unfolded state, we picked the samples below the FNC threshold from trajectories simulated with the ff99sb-disp forcefield, whereas for the folded ones we used samples from the same forcefield, i.e., ff99sb-disp by default. However, some cases remained where the samples above the FNC threshold from the ff99sb-disp forcefield were multimodal, that spread over a large range, or that had lower FNCs than the samples from ff14sb. For those systems, we selected ff14sb for the folded state. We discarded systems where neither force field resulted in a clear density peak in the FNC distribution above 0.8, or where either the folded or unfolded samples consisted of less than 10% of the entire dataset. In difficult cases, we checked several sample structures as well as the

FNC and RMSD time series and made a decision based on visual inspection. After processing, the "`MSR-megasim-merge`" dataset consisted of 271 wild-type systems, out of which 77 had folded states from amber ff14sb and unfolded states from amber ff99sb-disp, while the rest 194 featured both folded and unfolded samples from amber ff99sb-disp.

Due to the large number of sequences present in the mutant dataset, we could not afford to conduct sampling as thorough as for the wild-types. Instead, we relied on the presumption that point mutations or single insertion/deletion mainly affect local interactions in the folded state, and only weakly perturb the sample distribution in the unfolded state. Since our model only considers the protein backbone, we reused the unfolded samples from the wild-types for all point mutants, and generated MD simulations for all mutants in the folded state. Here we also assumed that the mutant folded state does not deviate completely from the native state of its wild-type, but would only be involved in local rearrangements, such as side-chain repacking. This assumption allowed MD simulations of mutant structures to be seeded from their wild-type folded conformation, as well as the use of the wild-type FNC to probe mutant foldedness. It further means that we can use the FNC defined by the wild-type native contacts to probe the foldedness of the mutants. In practice, we generated the mutant starting structures from their corresponding wildtype reference structure by exchanging the side-chain accordingly and by performing energy minimization. This is followed by a 1 μs simulation for each of the mutants using the amber ff99sb-disp force field. Since we expect the starting structure to be not the exact native structure of the mutant, we anticipated the need for a burn-in period, to allow the system to relax to a more stable native folded state. To select the length of such period, we split the trajectory into two parts so that the difference of the mean FNCs of each part would be maximized. The part after the burn-in period was then kept for the folded samples, except for situations where the FNC decreased monotonically throughout the simulation.

To validate the combination of samples of each wild-type with its mutants, we considered the impact of including mutant folded samples on the folding threshold for FNC computation. In cases where the previously classified unfolded samples had surpassed the folding threshold, it was no longer possible to define foldedness for the mutant based on its wild-type native contacts. In all other cases, samples were combined, since those coming from wild-type simulations only contributed to the unfolded population. After excluding cases violating the previous two assumptions, we obtained a set containing samples for 21,458 mutants, which we named the "`MSR-megasim-mutants-mosaic-disp`" dataset.

Download link: `https://doi.org/10.5281/zenodo.15641184`.

Complexin We have generated a small MD dataset for complexin-2 (Uniprot ID Q6PUV4), which is solely used for qualitative comparison of model predictions in Fig. 3C. The simulations were seeded using the AlphaFold2 predicted structure deposited in Uniprot. First, we produced a 5 μs trajectory with the Amber ff14sb force field (*72*) using our standard MD simulation protocol ("MD simulation protocol"). Subsequently, we generated dynamics with the Amber ff99sb-disp (*33*) force field. Here, the setup and equilibration were conducted in GROMACS (*66*) with initial structures being solvated in a cubic box with 1.2 nm padding and 0.135 molar KCL buffer and the custom ff99sb-disp TIP4P water model. After local energy minimization, the system was equilibrated in 0.1 ns (NVT) and 0.1 ns (NPT). Four production simulations of 1.5 μs were conducted in OpenMM following "MD simulation protocol"

We evaluated the simulation speed on an NVIDIA TitanV: 200 ns/day for ff14sb and 60 ns/day for ff99sb-disp. The latter throughput is reduced due to the more expensive 4-point water model.

Public MD datasets The following MD datasets were curated from public sources, as specified below.

- **DESRES fast-folding proteins** We use the fast folding protein simulations described in Ref. (*7*) under license. The dataset consists of 12 systems simulated with the charmm22* force field (*73*), with a cumulative simulation time of 8.2 ms. This dataset has only been used for a separate model whose results are shown in Fig. 3A and S9, but it is not used to obtain any of the other results presented throughout the manuscript. Data can be obtained from DESRES upon request.

- **DDR1** Simulations of 9 DDR1 kinases published in Ref. (*74*) with the amber ff99sb-ildn (*69*) forcefield and featuring a cumulative simulation time of 6.8 ms. Downloaded from: `https://osf.io/4r8x2/`

- **SETD8** Simulations of methyltransferase SETD8 (*75*) excluding trajectories involving small molecules. The dataset consists of 26 systems, has a total simulation length of 5.9 ms and uses the amber ff99sb-ildn force field. Downloaded from: `https://osf.io/2h6p4/`

- **SARS-CoV-2 exascale** We use the publicly available subset of the data published with Ref. (*26*), which consists of simulations for 24 systems and uses the Amber ff03 force field (*76*). The cumulative simulation time is 56.5 ms (when counting by trajectory), or an effective 81 ms (when treating chains independently). Downloaded from: `https://registry.opendata.aws/foldingathome-covid19/`. AWS resource name: `arn:aws:s3:::fah-public-data-covid19-cryptic-pockets`.

- **SARS-CoV-2 non-exascale** Non-glycosylated SARS-CoV-2 RBD data by Ref. (*77*). The dataset consists of a single system with 1.9 ms cumulative simulation time and uses the amber ff14sb (*72*) forcefield. Downloaded from: `https://registry.opendata.aws/foldingathome-covid19/`. AWS resource name: `arn:aws:s3:::fah-public-data-covid19-antibodies`.

- **MHC2 peptide simulations** We use the dataset of MHC2 in complex with peptides as published by Ref. (*78*). It consists of 68 systems with multiple chains and uses the amber ff99sb forcefield (*79*). The cumulative simulation time is 9 ms (when counting by trajectory) or effectively 27 ms (when treating protein chains independently). Downloaded from: `https://zenodo.org/records/15436451`.

- **Barnase-Barstar** Simulations provided by Ref. (*8*), consisting of one system with two chains using Amber ff99sb (*79*). The cumulative simulation time is 2.0 ms (when counting by trajectory) or 4.0 ms effective (when treating chains independently). Downloaded from: `https://zenodo.org/records/8252423`.

Experimental thermodynamics data High-throughput experimental measurements of protein stability at ambient temperature were used to fine-tune the model in combination with in-house generated datasets. Specifically, we extracted the ΔG (“`dG_ML`”) and $\Delta\Delta G$ values (“`ddG_ML`”) and the corresponding amino acid sequences for wild-types and mutants within the curated set (“`Dataset2_Dataset3`”) from the associated data in Ref. (*21*), resulting in approximately 674,000 cleaned entries. We randomly selected 43 wild types and a few mutants per wild type to add to the test proteins. They are added together with test proteins from other benchmarks and analysis for generating train/valid/test split, as described in “Data splitting procedure”. After the split, 502,442 sequences including 361 wild types and their mutants are in the training set. In Fig. 4, we present results for 95 wild-type proteins and their randomly sampled mutants, none of which were included in the training set. This evaluation set comprises proteins from the 43 initially selected wild types, along with additional proteins chosen to span a range of sequence similarities to the training data. This design allows us to assess how prediction accuracy varies with sequence similarity.

Model architecture

In this section, we describe the architecture of the proposed model and how it is trained. BioEmu is defined as a conditional generative model. BioEmu receives as input a protein sequence and generates independent and identically distributed (iid) samples from the approximate equilibrium distribution of that protein’s conformations. The iid generation of samples can be parallelized across a batch of random seeds, enabling efficient exploration of the equilibrium distribution of protein conformations orders of magnitude faster than standard sequential, correlated molecular dynamics simulations.

Protein sequence encoder As in prior work (*12*), the protein sequence S is encoded through the protein sequence encoder (Fig. 1C) to compute single and pair representations using pre-trained AlphaFold2 (*1*). We used mmseqs (*63*) from the Colabfold (*80*) interface `colabfold_search` with default parameters for efficient and large-scale multiple sequence alignment search, excluded templates entirely, and disabled AlphaFold2 recycling iterations. For generation, we fix the random seed to 0 and use the single and pair embeddings generated by AlphaFold2 model 3.

As the protein sequence encoder depends on no other variables than the protein sequence S, the single and pair embeddings for all proteins used in training and inference are pre-computed once and stored for fast retrieval.

Coarse-grained protein structure representation BioEmu represents 3D protein structures using a coarse-grained approach, following (*81*) (Fig. 1C). Only the backbone heavy atoms of the protein are represented via the *backbone frame* representation introduced in (*1*). As in (*81*), but unlike (*1*), BioEmu does not explicitly model side-chains or hydrogen atoms.

To convert an all-atom protein conformation to its backbone frame representation for a given residue, we use its C_α atom coordinate $\mathbf{r} \in \mathbb{R}^3$ and apply the Gram-Schmidt orthogonolization on the displacement vectors $C_\alpha \rightarrow N$ and $C_\alpha \rightarrow C$. This yields an orthonormal basis which can be represented as a rotation matrix $\mathbf{Q} \in SO(3)$. Repeating this for each residue, we obtain a sequence of position-orientation tuples, $\mathbf{x} := \{(\mathbf{r}_i, \mathbf{Q}_i)\}_{i=1}^{N}$, for all N protein residues. To recover the Cartesian backbone atom positions from the frame representation, we start with a reference backbone heavy-atom frame per residue type, with idealized atom positions, similarly to AlphaFold2 (*1*) or OpenFold (*82*). For example, for alanine, the idealized frame atom positions are:

$$\begin{array}{c} \mathrm{N} \\ \mathrm{C}_\alpha \\ \mathrm{C} \\ \mathrm{C}_\beta \\ \mathrm{O} \end{array} \begin{pmatrix} -0.525 & 1.363 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ 1.526 & 0.000 & 0.000 \\ -0.529 & -0.774 & -1.205 \\ 0.627 & 1.062 & 0.000 \end{pmatrix}.$$

We then apply the rotation matrix $\mathbf{Q}_n$ to obtain the rotated frame, and add the position vector $\mathbf{r}_n$ to the coordinates of all the atoms in the frame. Note that since the C_α is at the origin of the idealized frame, it will be at exactly location $\mathbf{r}_n$ upon applying this transformation.

Diffusion conditional generative model BioEmu functions as a sequence-conditional generative model: given a protein amino acid sequence, it parameterizes a distribution over backbone conformations. Formally, let $S = (a_1, a_2, \ldots, a_N)$ be a protein sequence with N residues $a_i \in \mathcal{R}$ from the set of 20 standard amino acids. BioEmu is a conditional diffusion model that can be used to sample 3D protein conformations $\mathbf{x}$ from a conditional distribution

$$\mathbf{x}_0 \sim p_\theta(\mathbf{x}|S), \tag{S1}$$

where θ are the learnable weights that parameterize the neural network that acts as a score model $s_\theta(\mathbf{x}|S)$. Since the dimensionality of $\mathbf{x}$ depends on the protein sequence length N, the dimensionality of the space that BioEmu defines a distribution over depends on the length of S. The sampling procedure that characterizes $p_\theta(\mathbf{x}|S)$ is obtained by simulating the reverse of a *forward diffusion* *process*, defined by a stochastic differential equation on the space of backbone frame representations $\mathbf{x}$:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{G}(\mathbf{x}, t)d\mathbf{w}, \tag{S2}$$

where $\mathbf{w}$ is a standard Wiener process, and $\mathbf{f}$ and $\mathbf{G}$, drift and diffusion coefficients respectively, are functional hyper-parameters. We choose $\mathbf{f}$ and $\mathbf{G}$ such that all residues as well as their positions $\mathbf{r}$ and orientations $\mathbf{Q}$ are corrupted independently. Specifically, the positions are corrupted with a variance-preserving stochastic differential equation (SDE) and a cosine noise schedule as described in (*83*). We refer the reader to (*84*) for further details on diffusing over the space of orientations, SO(3). The orientations are corrupted with a geometric noise schedule so that the marginal distribution of the change in orientation after time t is:

$$\mathcal{IG}_{\mathrm{SO}(3)}(\omega, \sigma^2) = \frac{1-\cos(\omega)}{\pi} \sum_{l=0}^{\infty} (2l+1) e^{-l(l+1)\frac{\sigma^2}{2}} \frac{\sin((l+\frac{1}{2})\omega)}{\sin(\frac{\omega}{2})}, \tag{S3}$$

where ω is the angle between rotations $\mathbf{Q}_t$ and $\mathbf{Q}_0$ computed as:

$$\omega = \arccos(\mathrm{trace}(\mathbf{Q}_t^\top \mathbf{Q}_0)/2 - 1/2). \tag{S4}$$

We use $p(\mathbf{x}, t)$ to denote the probability distribution of $\mathbf{x}$ at diffusion time t when $\mathbf{x}$ is corrupted in the above way, with the boundary condition that $p(\mathbf{x}, 0) = p(\mathbf{x})$ (the target distribution). If the initial positions $\mathbf{r}_0$ are bounded (a reasonable assumption for physical protein structures centered at the origin), then $p(\mathbf{x}, 1)$ is close to a simple *prior distribution* under which positions have a standard isotropic Gaussian distribution and orientations are uniformly distributed.

It has been shown that by training on samples $\mathbf{x}(0)$ from $p(\mathbf{x})$ together with corresponding samples from the conditional distribution of $\mathbf{x}(t)$ given $\mathbf{x}(0)$, a model can approximate the score $\nabla_{\mathbf{x}} p(\mathbf{x}, t)$; furthermore, if we know the score, we can construct SDEs under which the evolution of the probability density $\frac{\partial}{\partial t} p(\mathbf{x}, t)$ is reversed (*85*). Starting by sampling positions $\mathbf{r}$ and orientations $\mathbf{Q}$ from the prior and gradually 'denoising' by simulating one of these SDEs from $t = 1$ to $t = 0$, we can approximately sample from the target distribution.

Model training details are further described in section "Training methodology". For inference purposes, we smoothen the model weights using an exponential moving average. To sample structures with the trained model, we have investigated the Heun sampler (*86*) and the DPM solver (*87*), both leading to high-quality samples with fewer function evaluations compared to a traditional

Euler-Maruyama sampler. See Fig. S11 for a comparison of distributions with different samplers and number of steps. All results in the paper were obtained using the DPM solver.

Score model The score model (Fig. 1C) receives single and pair representations of the protein sequence $\mathbf{h} := \{\mathbf{h}_i\}_{i=1}^{N}$ and $\mathbf{z} := \{\mathbf{z}_{ij}\}_{i,j=1}^{N}$, corrupted frames $\mathbf{x} := \{\mathbf{r}_i, \mathbf{Q}_i\}_{i=1}^{N}$, relative sequence positions $\mathbf{p} := \{p_i\}_{i=1}^{N}$, and a diffusion timestep t, and predicts the score $s_\theta(\mathbf{x}, \mathbf{h}, \mathbf{z}, t)$. It resembles the structure modules of the AlphaFold2 (*1*) and Distributional Graphormer (*12*) models, and uses the invariant point attention (IPA) transformer architecture. See Fig. 1C for an overview of the architecture and Algorithm 1 for a detailed description. The translation and rotation scores produced by the score model in Algorithm 1 are defined in the local coordinate frame of each residue, and are invariant under rotation or translation of the entire structure. During denoising, the updates to backbone atom positions are therefore equivariant under rotation and translation of the whole structure.

Training methodology

Training begins with a pre-trained sequence encoder from AlphaFold2 (*1*), whose weights are frozen while a custom structure module is trained from scratch. We first train on a synthetic dataset derived from AFDB, with high sequence diversity and varied conformations for each sequence ("AFDB pretraining" in Table S3, see section "Pre-training on AFDB" for details). The pretrained model can predict diverse conformations for the same protein sequence, but it does not accurately model the probabilities of different conformational states. We then fine-tune on MD simulation data mixed with AFDB structures with the corresponding fractions of 95% and 5% ("Amber MD finetuning" in Table S3, see section "Fine-tuning on Amber MD data" for details). After fine-tuning on MD data, we perform one more stage of fine-tuning with the megascale experimental folding free energy measurements with PPFT, while still keeping AFDB and MD data with a small fraction ("Property prediction fine-tuning" in Table S3, see section "Training on folding free energies via property prediction fine-tuning (PPFT)" for details). This training procedure results in the main model BioEmu reported in this paper. We also separately fine-tuned the pre-trained model on DESRES fast-folders data (section "Fine-tuning on CHARMM MD data of fast-folding proteins"), which was used to produce the results in Figures 3A, S9 but not for any other results.

In all stages, we use the standard denoising score-matching loss as in (*81*). To train on experimental thermodynamic data, we add a novel loss term described in “Training on folding free energies via property prediction fine-tuning (PPFT)”. Table S3 provides a summary of all training settings. We define a training epoch as the model processing 500,000 protein structures.

Data splitting procedure We first defined a list of test proteins for benchmarking and analysis, then excluded from the training and validation sets any protein with a sequence similar to those test proteins. Specifically, we used the `mmseqs2` software (*63*) (version 15.6f452) and removed proteins if they have 40% or higher sequence similarity with any test protein of at least 20 residues in size, using `mmseqs2` with its highest sensitivity setting (8.0).

Pre-training on AFDB We initially train our model using a dataset derived from AFDB to encourage protein conformational variability (section “AlphaFoldDB processing” for details). Training examples are drawn by randomly selecting a sequence cluster and then a structure from within that cluster. While the structure is randomly selected, we always use the sequence associated with the highest pLDDT structure in the cluster as input to the model. This procedure effectively maps a single sequence to multiple structural conformations (Fig. 1E). In this stage of training, we use the standard denoising diffusion loss (*81*), defined as a sum over residues. We set the loss to zero in positions where there are insertions or deletions in the sampled structures relative to the representative sequence. The final model checkpoint was chosen based on the performance obtained on our curated OODVal benchmark (section “Multi-conformation benchmark sets”). For exact training parameters, please refer to Table S3.

We compared our pre-training strategy to the more straightforward approach of training the model on the PDB. Additionally, to assess whether the performance of the model was due to an increased diversity in sequence space, we additionally trained on foldseek (*64*) cluster representatives of AFDB with a pLDDT greater than 90 (*88*). This constituted a set of ~250k sequences distinct in both sequence and structure space. We found that models trained on the PDB and the high pLDDT subset of AFDB are worse in their ability to sample diverse conformations (Fig. S8), indicating that our curated subset of AFDB is an important contribution to facilitate multi-conformational learning.

Fine-tuning on CHARMM MD data of fast-folding proteins For the results shown in Fig. 3A and Fig. S9, we fine-tuned our best pre-trained model on a set of 12 fast-folding proteins (*7*) simulated with the charmm22* force field featuring sequence lengths ranging from 10 to 80 residues. For evaluation, this set is split into training, validation, and test subsets following a 10:1:1 ratio for each protein (leave-one-out cross-validation). PRB is used as the validation system in all splits except in the one where PRB is the test system. In that case, UVF was used instead. Specific training settings are reported in Table S3. All fast-folders results are obtained by evaluating the final model checkpoint from the last training epoch.

Fine-tuning on Amber MD data Starting from the best model identified during the pre-training stage, we perform two stages of fine tuning using the Amber MD datasets listed in Table S1, plus high-throughput experimental measurements of folding free energies from (*21*). In each fine-tuning stage, the training data is augmented with a small percentage of all datasets from the previous training stage, to prevent catastrophic forgetting. The hyper-parameters for finetuning are shown in Table S3. In the first stage of fine-tuning, we use standard denoising diffusion loss for all the MD simulation data and a small fraction of AFDB structures. In the second stage of fine-tuning, we include a large scale of 502,442 sequences (including 361 wild types and their mutants) with experimental folding free energy measurements ("`MegaExp`" dataset in Table S2), and use a novel loss to match experimental folding free energies by backpropagating through the sampling procedure (section "Fine-tuning on CHARMM MD data of fast-folding proteins"). We increase the proportion of MD simulations for which experimental folding free energies are available ("`MSR-megasim-merge`" and "`MSR-megasim-mutants-mosaic-disp`"). We have also included all the other MD simulations and AFDB structures with a smaller fraction. In this stage, we only fine-tune the model parameters of layer 1 and 8 of our score model (Algorithm 1). We find the inclusion of previous training data and freezing part of the model parameters help prevent catastophic forgetting and mode collapse.

During each training epoch, 500,000 training and 50,000 validation frames are sampled at with a weighted sampler. The probability weight of each frame is the product of a *MD dataset percentage* and a *normalized frame weight*. The dataset percentages, given in Tab. S2, were determined based on accumulated simulation time, sequence diversity and degree of convergence of the different datasets.

Most MD datasets use a frame weight of 1, but specific weights are applied for systems belonging to the following MD datasets:

- Systems in the `ONE-octapeptides` dataset are reweighted via Markov State Models (MSMs) so that each state (a region of configuration space) is sampled with the frequency determined by the MSM equilibrium probability (section "MSM reweighting for small peptide datasets").
- Systems in the `MSR-megasim-merge` and `MSR-megasim-mutants-mosaic-disp` datasets are reweighted based on their foldedness so that the training distribution recovers the experimental folding free energies (section "Connectivity filtering for post-hoc analyses").

In order to deal with systems of varying size, we define batches based on the total number of protein residues in a batch, up 1440 or 2048 depending on the training stage (Table S3). In order to reduce overhead caused by zero-padding, systems of similar size and same loss type (PPFT loss vs score matching loss) are grouped together when generating batches.

MSM reweighting for small peptide datasets The data distribution generated by MD is often biased towards the seeding structure since simulations are usually run in parallel, often starting from the same or a small number of seed structures. MSMs are a common approach to remedy this problem (*20*). In short, the classical approach first projects the $3N$-dimensional protein system into a low-dimensional representation, discretizes this projection using a clustering algorithm such as k-means, and estimates a transition matrix on these discrete states (*89*). This approach gives access to the equilibrium probabilities via the eigenvector of that matrix that corresponds to eigenvalue 1. Such eigenvector is then used as a probability distribution to draw samples from the MD simulation accordingly.

This analysis was applied to the ONE-octapeptide dataset, using two-dimensional TICA projections based on C_α-C_α distances and dihedral angles. We used a lag time of 1 ns for both TICA and MSM estimation, and 100 discrete states via k-means discretization for the MSM. These equilibrium probabilities are then used to reweight the sampling of training frames.

Connectivity filtering for post-hoc analyses It is common practice to perform MSM analyses on sets of states that are reversibly connected. A connected set of states is here referred to as one where each state is reachable from each other state via a sequence of trajectory transitions. Since multiple connected sets may exist, we select the one containing the most MD samples. As obtaining a connected set from data is numerically more stable than estimating a converged equilibrium distribution, this filter can be applied in situations where a converged MSM estimate could not be obtained.

This analysis was conducted for the ONE-cath1 dataset, using a linear VAMP projection (*90*) using a lag time of 5 ns and residue-residue minimal distances on heavy backbone and C_β atoms, excluding one residue at each terminus and two neighboring residues. Subsequent connectivity analysis was conducted by counting transitions between discretized states at a lag time of 500 ns based on the first 5 VAMP dimensions and a k-means clustering approach to obtain 200 states. Data outside of the largest connected set was discarded from subsequent analyses, which roughly translated to keeping 90-95% of the data on average. Free energy plots of ONE-cath1 (i.e., CATH domains presented in Fig. 3 and Fig. S10) were based on a secondary TICA projection obtained from trajectories inside the connected set of states.

Reweighting MD with experimental folding free energies As described above, we have generated MD simulation data for a subset of the sequences that are represented in the dataset of experimental folding free energies (ΔG) provided by (*21*). Since the MD simulations are too short to represent a converged sample of folding and unfolding events, the folding free energies estimated from histogramming the raw simulation data do not match their corresponding experimental measurements but primarily reflect whether the trajectory was started in the folded or unfolded state. To account for this, we reweigh the MD simulation data of each MEGAscale protein system with the corresponding experimental ΔG. For each system, first we classify all the MD conformations into folded and unfolded states, and then sample the folded and unfolded structures with different frequencies during training, such that the ratio of folded versus unfolded states seen by the model during training matches the target ratio given by the experimental ΔG. Specifically, the folding free energy is related to the probability that a protein will be found in a folded state in the equilibrium distribution.

The folding free energy ΔG can be defined in either of two directions, here we choose the convention to define it as the change in free energy when folding, i.e. $\Delta G = G_{\text{folded}} - G_{\text{unfolded}}$.

Then the probability of being in the folded state, p_{folded}, is related to the folding free energy is ΔG by:

$$\frac{p_{\text{folded}}}{1 - p_{\text{folded}}} = \exp\left(-\frac{\Delta G}{k_B T}\right), \tag{S5}$$

where T is the temperature and k_B is the Boltzmann constant. The probability of being in the folded state can be expressed as the expectation value of foldedness $p_{\text{folded}} = \mathbb{E}_{\mathbf{x}}[f(\mathbf{x})]$, where f ranges from 0 (unfolded) to 1 (folded). For both our reweighting and model evaluation, we define f as:

$$f_{\text{FNC}}(\mathbf{x}) = H\left(Q(\mathbf{x}) - Q_{\text{threshold}}\right), \tag{S6}$$

where H is the Heaviside step function, $Q(\mathbf{x})$ the fraction of native contacts, and $Q_{\text{threshold}}$ a system-dependent threshold. For a given protein structure $\mathbf{x}$, the fraction of native contacts (FNC) is defined from pairs of residues that are at least 3 residues apart in the amino acid sequence but which are physically within 10 Å of each other in a reference folded structure. Specifically, we follow notations as in (*91*):

$$Q(\mathbf{x}) = \frac{1}{N}\sum_{(i,j)} \frac{1}{1 + \exp\left[\beta\left(r_{ij}(\mathbf{x}) - \lambda(r_{ij}^0 + \delta)\right)\right]}, \tag{S7}$$

where $r_{ij}(\mathbf{x})$ and r_{ij}^0 are the contact distances between i and j in the configuration x and the reference conformation (native state). $\beta = 5$, $\lambda = 1.2$, and $\delta = 0$ are constants, representing the softness of the switching function, the reference distance tolerance and offset, respectively. For each simulated MEGAscale system, we use its PDB structure as the reference conformation. Any given sampled structure can then be classified as folded or unfolded by setting a threshold on the calculated FNC value.

To account for differences in the observed FNC distributions, we set the FNC threshold in a system-dependent but unsupervised manner. Specifically, considering that we initialized multiple MD trajectories separately starting from folded and unfolded states for every protein, and those are not long enough to observe transitions, the distribution of FNC for each system is generally separated into peaks near 1 and 0, representing folded and unfolded states, respectively. In order to obtain a smoother distribution of FNC values for each system, we use a kernel density estimate and then use its minimum within the range of 0.45-0.9.

Training on folding free energies via property prediction fine-tuning (PPFT) Although the reweighting method encourages the model to learn the correct experimental folding free energies with MD simulation data alone, we have empirically found that this convergence is slow, especially for systems where unfolded states are rare (large negative ΔG). More importantly, experimental observables such as ΔG can only be used in a standard diffusion model training approach if both folded and unfolded structures are available, e.g. obtained via MD simulation, whose computational costs would limit us to rather few training systems. Here we conducted a large number of MD simulations for 22,389 protein sequences from the MEGAscale dataset, and yet this only corresponds to about 2% of the entire experimental dataset. On the other hand, directly training diffusion models to sample distributions that match a given set of expectation values via generation and backpropagation is computationally prohibitive. Training costs would increase roughly in proportion to the number of denoising diffusion steps compared to regular score matching – in our case that would be a factor of 50 to 100.

To avoid these limitations and take advantage of high-throughput experiments such as the ones in (*21*), we have developed a novel and efficient method that trains diffusion models to generated distributions that respect a given set of properties of these distributions, e.g. experimental expectation values. As the method is most likely effective with a pre-trained diffusion model, we call it property-prediction fine-tuning (PPFT).

PPFT leverages that many low-dimensional properties of the distribution can be accurately predicted without performing a complete rollout of the diffusion model. Nonetheless, the training principle follows a simple prediction and backpropagation scheme. For a given sequence with an associated experimental ΔG, we can roll out the denoising process to generate a clean sample and compute its foldedness. We rewrite Eq. S5 to relate the sample expectation value of foldedness to the folding free energy:

$$\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})] = \frac{\exp\left(-\frac{\Delta G}{k_B T}\right)}{1 + \exp\left(-\frac{\Delta G}{k_B T}\right)}. \tag{S8}$$

where we use $T = 295\text{K}$ because the experimental measurements are performed at room temperature. Then, for each protein sequence s in a PPFT training batch, we define its loss term as:

$$L_{\text{ppft}}(s) = \frac{2}{M(M-1)} \sum_{m=1}^{M} \sum_{n>m} \left(f(\mathbf{x}^m) - f_{\text{target}}\right)\left(f(\mathbf{x}^n) - f_{\text{target}}\right), \tag{S9}$$

where we generate M iid samples with the same protein sequence, and compute f_{target} from the right hand side of Eq. S8 using the corresponding experimental ΔG. The cross term in Eq. S9 is used to minimize the expectation:

$$\mathbb{E}_{\mathbf{x}^m,\mathbf{x}^n}\left[(f(\mathbf{x}^m) - f_{\text{target}})(f(\mathbf{x}^n) - f_{\text{target}})\right] = \left(\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})] - f_{\text{target}}\right)^2. \tag{S10}$$

If a standard mean squared error loss were to be used instead,

$$\mathbb{E}_{\mathbf{x}}\left[(f(\mathbf{x}) - f_{\text{target}})^2\right] = \left(\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})] - f_{\text{target}}\right)^2 + \text{Var}[f(\mathbf{x})] \tag{S11}$$

would be minimized, which contains an additional variance term that would encourage mode collapse. After that, we average over the sequences in the batch to get the PPFT loss term L_{ppft}.

We notice that the definition of foldedness $f(\mathbf{x})$ by Eq. S6 is non-differentiable due to the Heaviside step function, and the system-dependent threshold adds additional complication. In PPFT, we replace the step function with a sigmoid function:

$$f_{\text{FNC}}(\mathbf{x}) = \sigma\left(k\left(Q(\mathbf{x}) - Q_{\text{threshold}}\right)\right), \tag{S12}$$

We choose $k = -20$, $Q_{\text{threshold}} = 0.5$ for all protein systems. The sigmoid function σ is differentiable and thus enables backpropagation. It also approaches a Heaviside step function when the slope k is sufficiently large.

In the model finetuning stage, for those systems with both simulation and experimental ΔG data, we combined L_{ppft} with the usual score matching loss L_{sm}, i.e.:

$$L = L_{\text{sm}} + wL_{\text{ppft}}, \tag{S13}$$

with a weight $w = 10$. Even though the simulation data was reweighted using the experimental ΔG based on the method described in section "Reweighting MD with experimental folding free energies", we find that the inclusion of the L_{ppft} loss notably sped up ΔG model convergence.

As described above, a key requirement for PPFT to be computationally efficient is to avoid executing full diffusion model denoising with hundreds of denoising steps. To mitigate this cost, and considering that folding/unfolding are changes easily recognizable at a coarsed-grained level at earlier denoising levels, we considered reducing the number of integration timesteps, which sacrifices sample quality, but still predicts the foldedness accurately. In practice, we find that 35 timesteps are sufficient when used alongside the DPMsampler. To further reduce cost, we denoised to a specified intermediate noise level $\hat{t}$ to then perform clean sample extrapolation $\hat{\mathbf{x}}_0$ using the reparameterization trick (*92*), with

$$\hat{\mathbf{x}}_0 = (\mathbf{x}_{\hat{t}} - \sqrt{1 - \bar{\alpha}_{\hat{t}}}\epsilon_0)/\sqrt{\bar{\alpha}_{\hat{t}}}. \tag{S14}$$

The foldedness is then predicted from $\hat{\mathbf{x}}_0$ after denoising 8 out of 35 timesteps. We remark that while the coarse-grained nature of foldedness enables us to greatly reduce the number of rollout steps and model evaluations, this may not be applicable for every properties of interest. In such scenarios the adjoint method may be needed for computationally-affordable and numerically-stable training (*93*).

As a final measure towards increasing efficiency we leverage partial backpropagation, which has shown to effectively reduce computational costs in image-related tasks (*94, 95*). Here we apply backpropagation only through the final extrapolation and denoising step 3, 4, 5. In other denoising steps the score function is detached from the computational graph.

We summarize the PPFT method in Algorithm 2, and a combination of techniques contribute to its performance:

1. Definition of a differentiable target function,
2. A base model that is able to qualitatively sample multiple conformations,
3. Partially freeze model parameters,
4. Cross-target matching loss term,
5. Joint training with regular score matching,
6. Use of a higher order sampler to reduce integration timesteps
7. Extrapolation, and
8. Partial backpropagation.

We find that 2-5) are particularly helpful for reducing the mode collapse problem that results from overoptimization of the property prediction loss function, while 6-8) help to greatly reduce the rollout and backpropagation steps, such that direct backpropagation is feasible with current compute requirements.

Multi-conformation benchmarking

We begin by describing the benchmark sets, their rationale for inclusion, and curation criteria in "Multi-conformation benchmark sets". Additional details about how we constructed an uncontaminated benchmark for evaluation is provided in "Curation of the OOD family of benchmarks". In "Multi-conformation metrics" we provide details on the summary metrics we use in order to evaluate multi-conformation sampling capabilities of BioEmu and other competing models. The influence of sequence similarity and AlphaFold2 evoformer embeddings on model performance is investigated in "Influence of sequence similarity on multi-conformation performance" and "Influence of AlphaFold2 evoformer training on multi-conformation performance", respectively. Finally, in section "Multi-conformation baseline methods" we provide insights into the baseline methods used for comparison with our models as well as the parameters that were chosen for them.

Multi-conformation benchmark sets In order to evaluate the multi-conformation sampling capabilities of the pre-trained and fine-tuned models, we manually curated several several example sets of structural biology interest. For some of the benchmarks, this curation includes manual labeling of residues involved in specific conformational changes. Full lists of PDB and chain identifiers (`label_asym_id`), along with residue labels for alignment and metric computation, are provided in the benchmark repository (*61*). Details on each individual benchmark are provided below:

- **OOD60**: A collection of 19 examples from the PDB deposited after the AlphaFold2 monomer model cutoff date (April 30, 2018). A 60% sequence similarity cutoff is used to remove anything from this benchmark that is similar to any chain in the PDB prior to the specified cutoff date. This benchmark represents an unbiased evaluation of multi-conformation sampling, and contains several representative examples of the type of conformational changes present in other benchmarks. Metric-wise, we use RMSD as defined on either a local region, or the entirety of the protein, depending on each case.
- **Domain motion**: 22 examples representing large-scale hinge motions. Only global RMSD is used for evaluation in this benchmark.

- **Cryptic pocket**: 34 example pairs featuring a conformational change characterized by the formation of a binding site that is induced in a *holo* (bound) structure, but not in the *apo* (unbound) structure. Many of these examples were further curated from the CryptoSite benchmark (*96*) or other related works (*97*). The binding site and other parts involved in the conformational changes were manually-defined, and local RMSD was used as a metric.

- **Local unfolding**: A set of 21 examples, where a segment of at least 8 residues undergoes an unfolding transition, including some examples from the benchmark proposed in (*98*). For this benchmark we defined the segment of the protein that can unfold or detach and measure the fraction of native contacts between this segment and the entire protein to track whether a sample was folded or unfolded.

- **OODVal**: A manually curated set of 11 examples picked after the AlphaFold 2 monomer model cutoff date but has no overlap with the OOD60 set described above, which we use for pre-trained model selection purposes. Only global RMSD is used as a metric in this benchmark.

Curation of the OOD family of benchmarks As mentioned in "Multi-conformation benchmark sets", we selected pairs of reference PDB structures deposited after the AlphaFold2 monomer model cutoff date to avoid potential dataset contamination during evaluation. We first extracted and separated all chains for all PDB entries after the mentioned cutoff date. Each protein entity inside each entry is then associated with a unique UniProt segment using SIFTS annotations (*99*). A sequence clustering procedure using `mmseqs2` is then applied on all Uniprot segments, using a minimum sequence identity threshold of 0.99. Within each sequence cluster, we perform a structure clustering procedure on the associated PDB entities, similar to the one reported in (*13*). We used TM-score as the primary structural similarity metric within each sequence cluster. An agglomerative clustering procedure, implemented in `scikit-learn`, was then applied with a maximum allowed TM-score of 0.7 between clusters. Benchmark pairs were selected from cluster representatives based on the following criteria, which are applied to the resolved structures: (i) Minimum sequence length of 50 residues, (ii) Maximum coil content of 40%, (iii) Minimum sequence identity of 80% between structures, (iv) Maximum difference in sequence lengths of 50 residues.

For the OOD60 benchmark, care was taken that the remaining pairs were at most 60% sequence-similar to the AFDB training set. OODVal was selected as the set difference between the whole OOD and OOD60 sets. Both sets underwent thorough manual curation to ensure unphysical or unrealistic examples were excluded. Some of the criteria applied for curation included checking whether an intra-domain conformational transition was present, whether that occurred in a region that is resolved in both references, or filtering for chains that formed a single long helix, as we deemed their stability outside a complex unlikely.

Multi-conformation metrics For most benchmarks we used RMSD of backbone atoms as the primary metric. For the local unfolding benchmark, we used a contact map based solely on C_α atoms to measure interactions between the unfolding region and the rest of the protein, since the unfolded state lacks a well-defined reference structure. For most multi-conformation benchmarks, we used the experimental sequence as defined by the `_entity_poly.pdbx_seq_one_letter_code_can` field in the mmCIF dictionary entry. If the experimental sequences differed between the two reference structures, we sampled both sequences in equal proportion. For the cryptic pocket benchmark, only the apo conformation's experimental sequence was used, as the apo conformation is more difficult to sample. Global pairwise sequence alignments were used to compute metrics as needed when the sampled sequences differed from the experimentally-resolved reference sequences in the mmCIF files. For this, we mostly used the default parameters of BioPython's `PairwiseAligner`, apart from manually setting an open gap penalty of 0.5.

We computed two key summary statistics to evaluate the multi-conformation capabilities of our model, as well as to compare it against other approaches:

- **Coverage**: the fraction of reference conformations sampled, evaluated using a chosen metric across various thresholds. We consider a conformation as covered if at least 0.1% of samples are within a specific threshold the corresponding reference structure.
- ***k*-recall**: the average metric value (e.g., RMSD or FNC) for the top 0.1% of samples closest to each reference structure.

Before computing metrics, we filtered out unphysical samples exhibiting chain breaks or atomic clashes. Specifically, we looked at $C\alpha$-$C\alpha$ and C-N distances between sequence-adjacent residues and ensured that these do not surpassed 4.5 Å and 2.0 Å thresholds, respectively. Additionally, distances were computed between any two backbone atoms of different residues and we ensured that samples did not contain any such distances below a threshold of 1.0 Å.

For a detailed definition of the metrics and benchmarks please refer to our benchmarks repository at `https://github.com/microsoft/bioemu-benchmarks`.

Influence of sequence similarity on multi-conformation performance We analyzed how BioEmu's performance depends on sequence similarity to the training set to assess generalization and reduce the risk of memorization. We evaluated the fraction of successfully sampled reference conformations (coverage) using predefined success thresholds: 3Å RMSD for OOD60 and domain motion, 1.5 Å RMSD for cryptic pockets, FNC values of < 0.3 and > 0.7 for unfolded and folded states, respectively, in the local unfolding benchmark. Multi-conformation coverage is computed for different subsets of the test set, as a function of sequence similarity to the entire training set (Fig. S5A). Additionally, we show the recall – defined as the mean RMSD or FNC values of the 0.1% best samples – plotted against train-test sequence similarity (Fig. S5B). Here, we define sequence similarity as the maximum similarity of a test sequence to the training dataset given at least 30% coverage of the test sequence. Sequence similarities below 25% are considered unreliable due to their sensitivity to alignment parameters. Between 25% and 40% sequence similarity values, the coverage of the domain-motion benchmark increases, and that of OOD60 slightly increases. Other benchmarks show little dependence on sequence similarity. In the local unfolding benchmark, the folded state coverage slightly decreases with sequence similarity, while the unfolded state coverage slightly increases. No clear trend is observed in other benchmarks. Even for domain motion, performance plateaus beyond 35% sequence similarity, suggesting that the fine-tuned BioEmu model does not rely on memorization beyond a baseline similarity.

Influence of AlphaFold2 evoformer training on multi-conformation performance Because BioEmu's input module uses AlphaFold2 evoformer embeddings, there is a risk that its performance on multi-conformation benchmarks (Fig. S7) may stem from memorization or retrieval from the embeddings, rather than true generalization – except in the OOD60 benchmark.

Towards that end, we trained an entirely new end-to-end model for encoding protein sequences, thus avoiding the dependence on AlphaFold2 evoformer embeddings. The end-to-end model processes MSA data directly, similar to AlphaFold2, and employs 4 MSA blocks and 48 Pairformer layers, following the AlphaFold3 architecture (*2*). Pairformer layers were used instead of evoformer ones due to their reduced computational cost. The model was first trained on a non-redundant subset of the PDB, and its embeddings were then used in the same pretraining and fine-tuning stages as BioEmu. Specifically, for the training of this model we excluded any training sequence with more than 40% sequence similarity to any test sequence in the static multi-conformation benchmarks. The resulting model achieved performance comparable to BioEmu (Fig. S6). The performance in the OOD60 benchmark at a 3Å success threshold is roughly 10% lower than BioEmu, but exceeds the BioEmu performance at 4Å and beyond. Since OOD60 is out-of-distribution for both BioEmu and AlphaFold2 training sets, this performance variation is unlikely due to the train-test split. In all other benchmarks, the end-to-end model performance is either equal or surpasses that of BioEmu. Although the end-to-end model was trained with a stricter train-test split, it still matches or outperforms all other baseline methods. These baselines either use more lenient splits or reuse AlphaFold2 parameters, such as in MSA subsampling strategies. An exception is AFCluster, which performs better in the cryptic pocket apo benchmark. However, we note that the majority of systems in this benchmark were included in the original AlphaFold2 training set, and hence better performance from this baseline could be expected in this case.

Overall, these findings provide no evidence that the performance of BioEmu in the multi-conformation sampling tasks relies on memorization. However, because the end-to-end model does not consistently outperform BioEmu on the static multi-conformation benchmarks, there is currently no strong justification to replace the AlphaFold2 evoformer embeddings in the main BioEmu model. For completeness, we repeat the analysis of how multi-conformation performance depends on the train-test sequence similarity for the end-to-end model (Fig. S5A). Similarly to the fine-tuned model results, there is no clear trend relating multi-conformation prediction performance to sequence similarity: For the end-to-end model, domain motions and cryptic pocket transitions quickly ramp up to their final performance between 25% and 30% sequence similarity, while for other benchmarks the performance stays constant or even somewhat decreases for more similar, indicating that there are not sufficient examples in the multi-conformation benchmarks to compute a reliable trend as a function of the similarity variable, or that global sequence similarity is not the best metric to predict performance in these tasks.

Multi-conformation baseline methods We selected AFCluster (*15*), AlphaFlow (*13*), DiG (*12*), and uniform MSA subsampling (depth 100) as baseline methods for multi-conformation sampling. AFCluster is an MSA subsampling-based method that clusters the MSA and feeds each cluster representative to AlphaFold2 to generate distinct samples. In contrast, AlphaFlow and DiG – like BioEmu – are deep-learning-based generative models. For all of these baselines, MSAs were generated using ColabFold (*80*) with default parameters. For AlphaFlow and DiG, we generated the same number of samples as with BioEmu. For AFCluster, the number of samples was limited to the number of clustered MSAs produced by the method. For uniform MSA subsampling, we drew the same number of samples as with BioEmu, except when the original MSA depth was below 100 – in those cases, only a single sample was generated. AlphaFlow runs included the recommended `--self_cond --resample` flags during evaluation. Comparisons of our trained models against these baselines on the proposed benchmarks are provided on Fig. S7.

Protein stability benchmarks

System selection We selected proteins from ProThermDB (*38*) with experimental ΔG of unfolding $\geq$ 8 kcal/mol and a single-chain asymmetric unit. We excluded proteins based on several criteria, resulting in a final set of 26 proteins. Exclusions included two membrane proteins, one with an undetermined sequence, and one nucleic acid–protein complex. The initial selection comprised 140 systems. Additionally, we curate a smaller subset consisting of 26 proteins after excluding systems with one of the following conditions:

- proteins annotated as membrane protein
- proteins with a ligand reported under `_refine_hist.pdbx_number_atoms_ligand` (e.g., 1C52)
- proteins with disulfide bonds as reported under `_struct_conn.conn_type_id` (e.g., 1LVE)
- oligomeric proteins (e.g., 1ROP, supposedly only stable as a dimer) or proteins in protein-RNA complexes

- proteins with ligands not reported in `_refine_hist.pdbx_number_atoms_ligand` (e.g., 2LCP)
- proteins with repeated entries due to differing capitalization

We also used the CALVADOS test set from IDRome (*39, 41*), which includes 65 intrinsically disordered proteins (IDPs), to benchmark stability. Sequence similarity search indicated that there was only one protein with a similarity above 40% with respect to the training set of BioEmu.

Evaluating free energy predictions We use the same definition of foldedness as in PPFT to evaluate the folding free energy ΔG of a given protein sequence. Depending on the experimental folding free energy, we generate between 200 and 10,000 samples per protein for evaluation. We compute the change in folding free energy upon mutation, $\Delta\Delta G$, as the difference between the mutant's ΔG and that of the wild type: $\Delta\Delta G_{\mathrm{mut}} = \Delta G_{\mathrm{mut}} - \Delta G_{\mathrm{wt}}$. To estimate confidence intervals in the ΔG predictions, we used the Clopper-Pearson method.

Energy landscape MAE

Folded-state filtering MD force fields are biased towards folded protein states, which typically prevents the sampling unfolding transitions at room temperature. The resulting TICA projections therefore only describe folded-state ensembles and fail to represent unfolded states. Since BioEmu's samples represent the full protein ensemble, applying an MD-derived TICA projection often yields non-informative projections for unfolded samples, usually appearing as high-variance noise in the free energy landscape. To ensure a fair comparison – especially for datasets like the CATH domains that only include folded-state MD data – we filter our samples based on their fraction of native contacts (threshold of 0.5), as defined in Eq. S6.

Macrostate MAE definition Assessing the mean absolute error on protein conformations is a non-trivial task for two reasons. a) Conformational landscapes and corresponding free energy surfaces cannot be directly assessed in $3N$-dimensional space, but require a projection space in which density – and thus free energy – can be computed. b) Free energy landscapes are often very rough and transitional regions have extremely low probabilities compared to metastable states.

Errors in predicting the relative probabilities of metastable states versus those of transition regions can be considered two distinct classes of error. In this paper, our goal was to sample metastable states such as folded or unfolded in the correct ratio, and therefore we focused on the first error. We have chosen the following approach to quantifying the mean absolute error (MAE) of protein free energy landscapes over metastable protein states, which are often referred to as macrostates: First, we parameterized a linear TICA projection (time-lagged independent component analysis (*54*)). Since TICA, like all dimensionality reduction techniques, is fundamentally limited by availability of data, we have limited this analysis to a subset of test systems with sufficient MD data. Only trajectories within the connected sets described in the section "Connectivity filtering for post-hoc analyses" were used. Second, we defined macrostates in the 2D TICA space and clustered them using Hidden Markov Models (HMMs) (*90*), a commonly used approach in MD simulation analysis. HMMs were estimated at comparably short lag-times of 1 ns and with 3 hidden states as a numerically stable choice.

The macrostate MAE (mMAE) was computed by assessing the relative free energies G_i within each macrostate i by sample counting:

$$G_i = -k_B T \ln(p_i) + \text{const}, \tag{S15}$$

with k_B the Boltzmann constant, T the temperature, and p_i the normalized histogram count for macrostate i. As not all macrostates were sampled by our model for the systems considered, a prior count of 1 was assigned to each macrostate. For a model with 10,000 samples, this corresponds to setting a minimum probability of $p_i = \max(p_i, 10^{-4})$, which defines the resolution limit of the model. Relative free energies from ground truth MD distribution and model samples are offset such that $\min_i G_i = 0$. The overall mMAE between model prediction (ML) and ground truth (GT) was then computed as:

$$\text{mMAE} = \frac{1}{N} \sum_i^N \text{abs}(G_i^{\text{ML}} - G_i^{\text{MD}}). \tag{S16}$$

To evaluate BioEmu's performance on MD-generated free energy landscapes, we have applied our mMAE metric to a random test set from the CATH1 dataset. All of these systems have $\geq 100\mu s$ MD data.

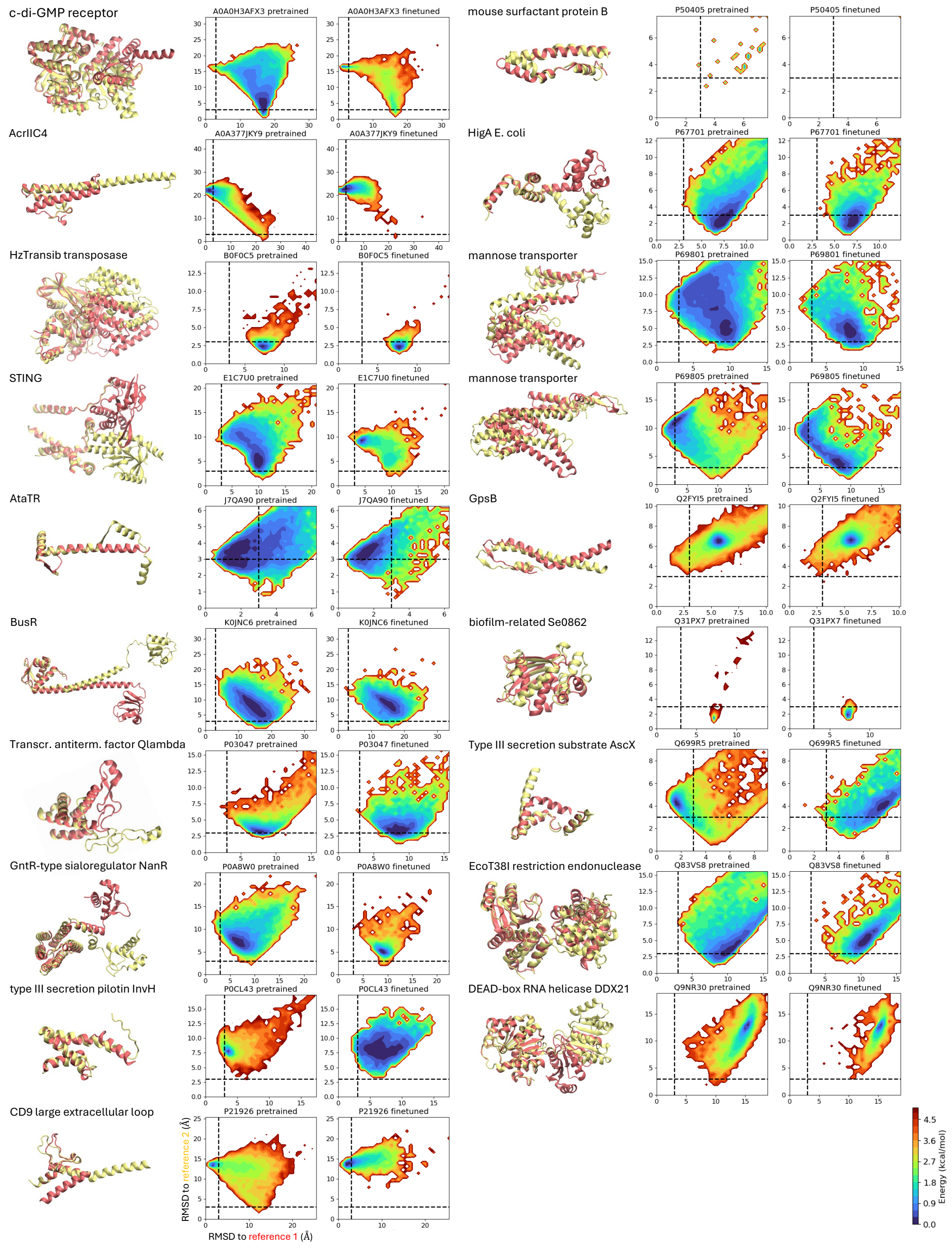

Figure S1: **Multi-conformation benchmark: OOD60.** Benchmark evaluates BioEmu's ability to sample distinct conformations for proteins with ≤ 60% sequence similarity to the AlphaFold2 training set. Each panel compares two distinct experimentally determined PDB structures (red and yellow). The energy landscapes depict the empirical free energy distributions generated by the pre-trained and fine-tuned BioEmu models, plotted against the global C_α RMSD to each reference structure (in Å). A successful match is defined as a sample within 3 Å RMSD of a reference structure (indicated by dashed lines). The detailed benchmark definition, including PDB codes, is provided in (*61*).

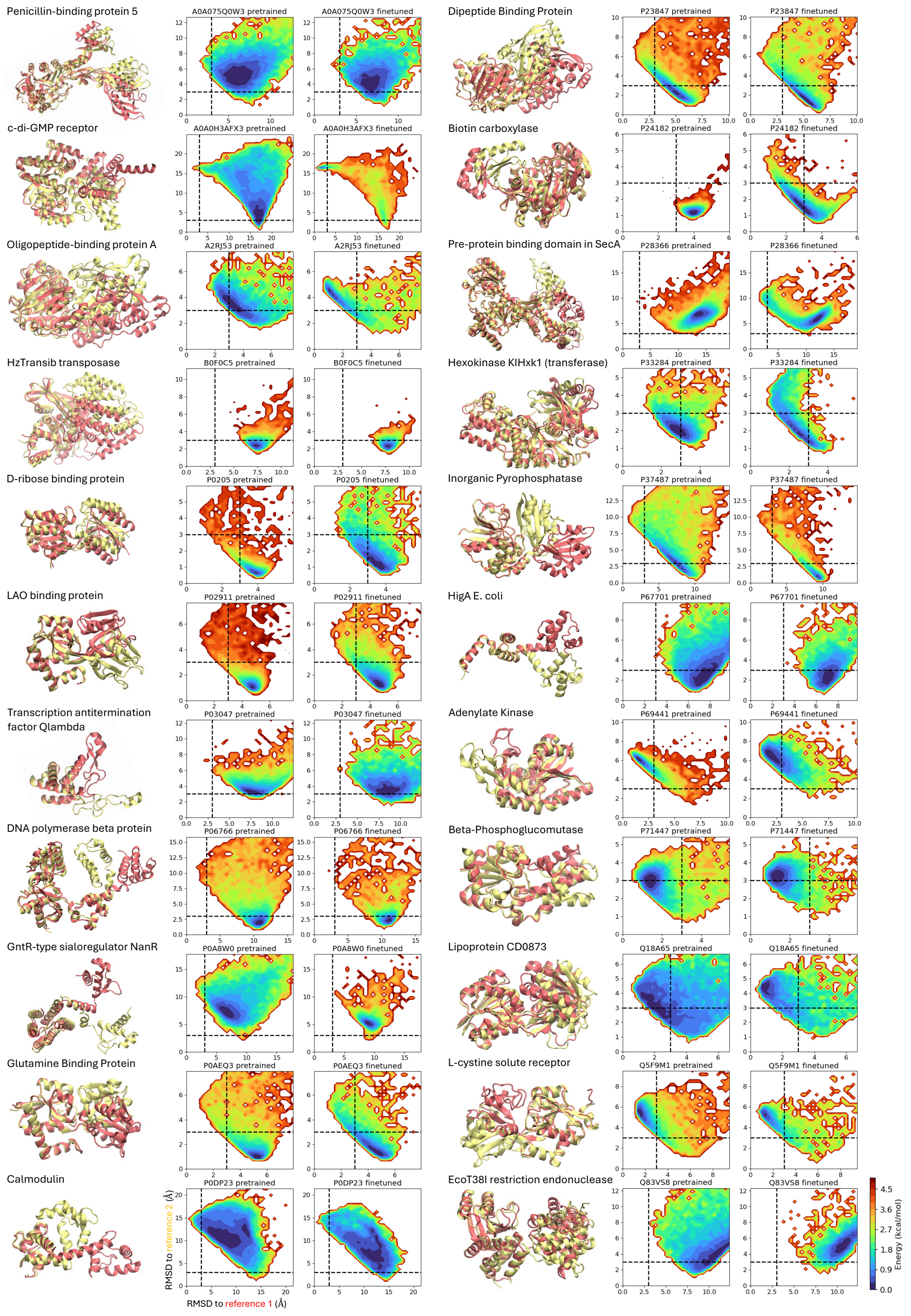

Figure S2: **Multi-conformation benchmark: Domain motions.** This benchmark evaluates BioEmu's ability to sample large-scale domain rearrangements. Each panel compares two reference PDB structures (red and yellow) and shows the empirical free energy landscapes generated by the pre-trained and fine-tuned models, plotted against global C_α root mean square difference (RMSD) to each reference (in Å). A match within 3Å RMSD (dashed lines) is considered successful. The detailed benchmark definition, including PDB codes, is provided in (*61*).

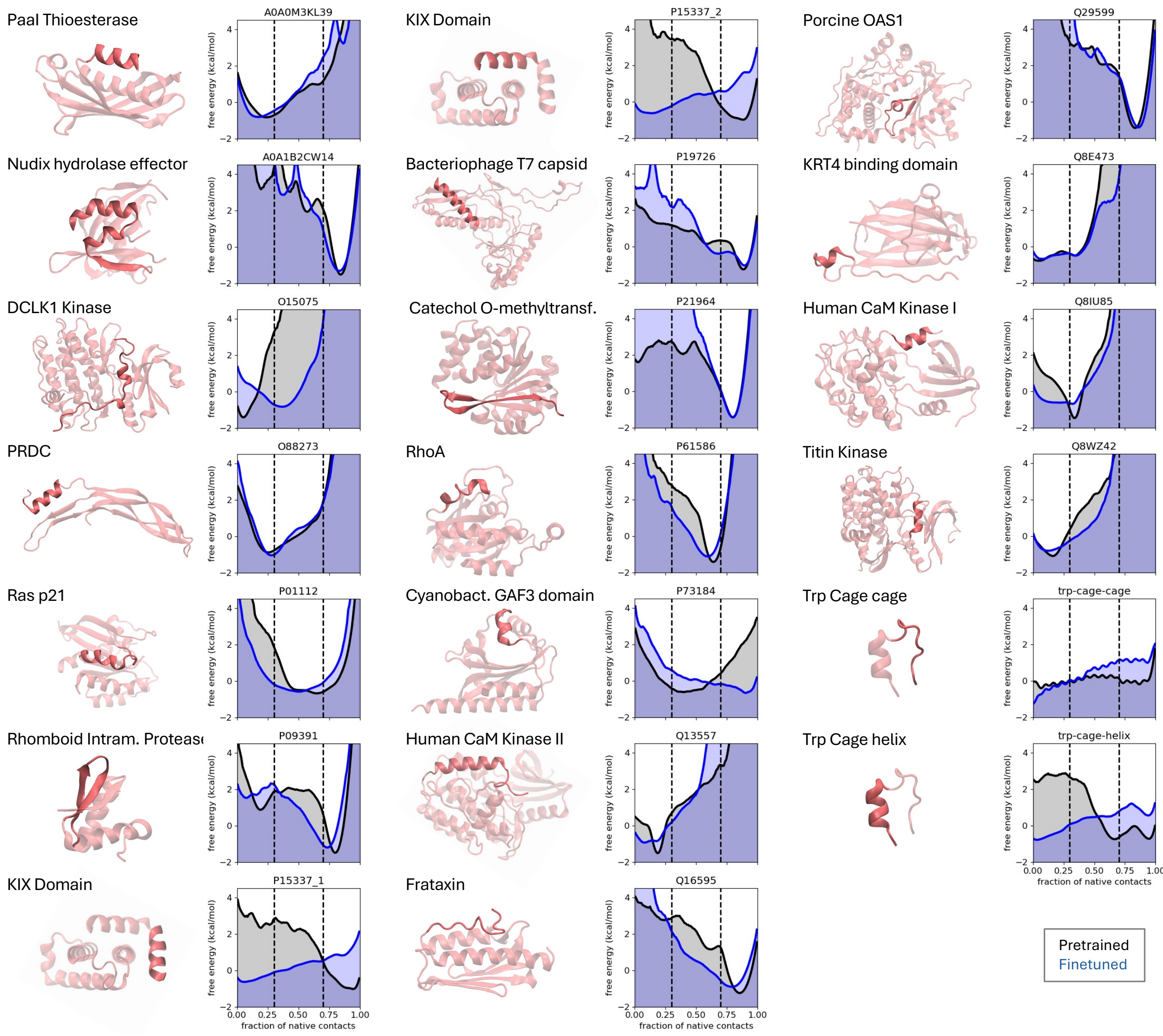

Figure S3: **Multi-conformation benchmark: Local unfolding.** This benchmark tests BioEmu's ability to sample partially unfolded states. Each panel displays the folded reference structure (red) and energy landscapes from the pre-trained (black) and fine-tuned (blue) models, plotted against the fraction of native contacts (FNC) between the unfolding region and the rest of the protein. Samples with FNC > 0.7 are considered folded; those with FNC < 0.3 are considered unfolded. The detailed benchmark definition, including PDB codes, is provided in (*61*). Notable differences between models include improved unfolded-state sampling for KIX and Trp Cage in the fine-tuned model, and suppression of structures that are only stable in crystallographic environments (e.g., domain swapping in crystals of Nudix hydrolase dimers) that appear metastable in the pre-trained model.

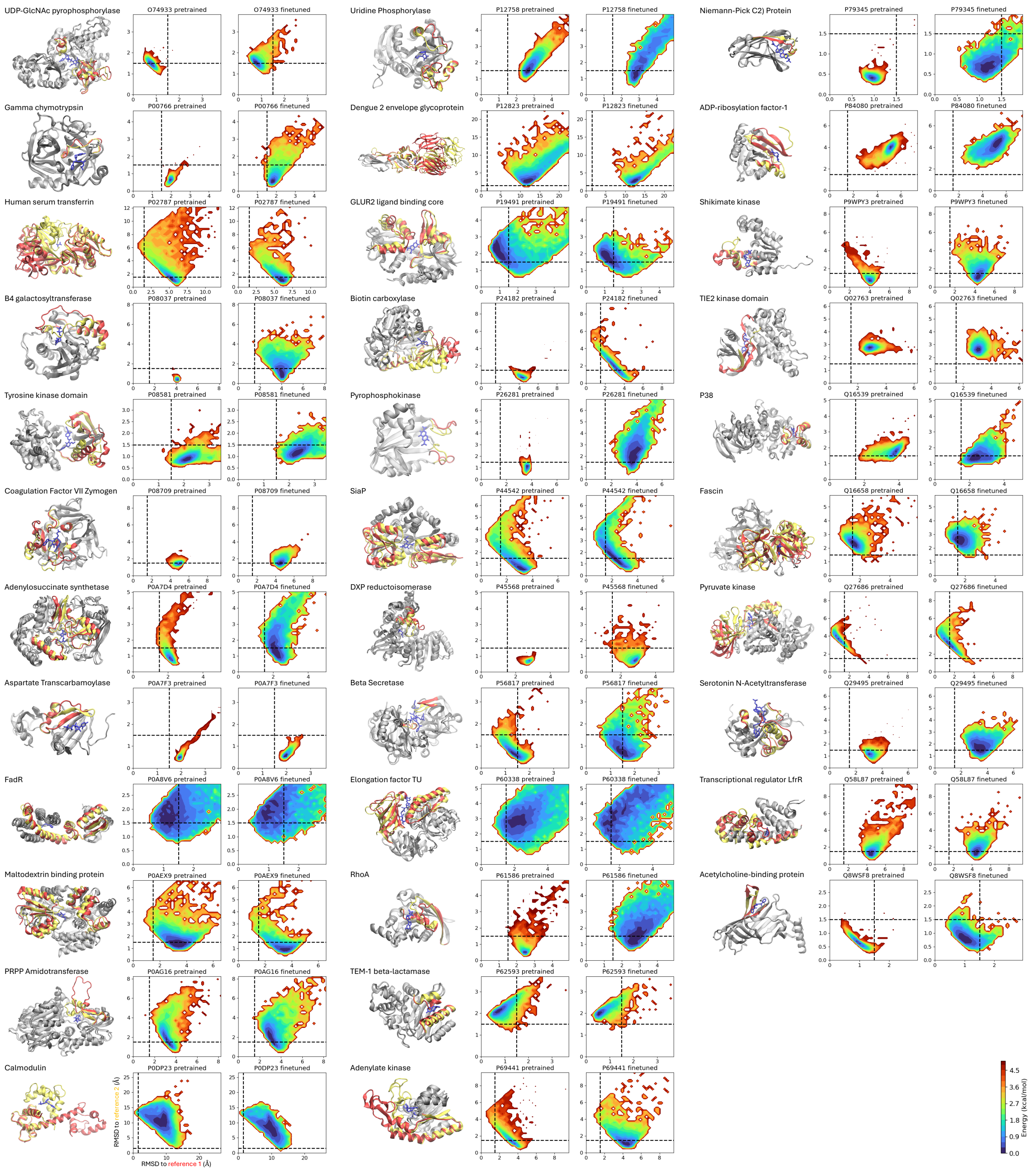

Figure S4: **Multi-conformation benchmark: Cryptic pockets.** Each panel compares two reference structures (red: apo, yellow: holo) and shows energy landscapes from the pre-trained and fine-tuned models. Here, local C_α root mean square difference (RMSD) to each reference (in Å) is used to assess pocket formation. Matches within 1.5 Å (dashed lines) are considered successful. The detailed benchmark definition, including PDB codes, is provided in (*61*).

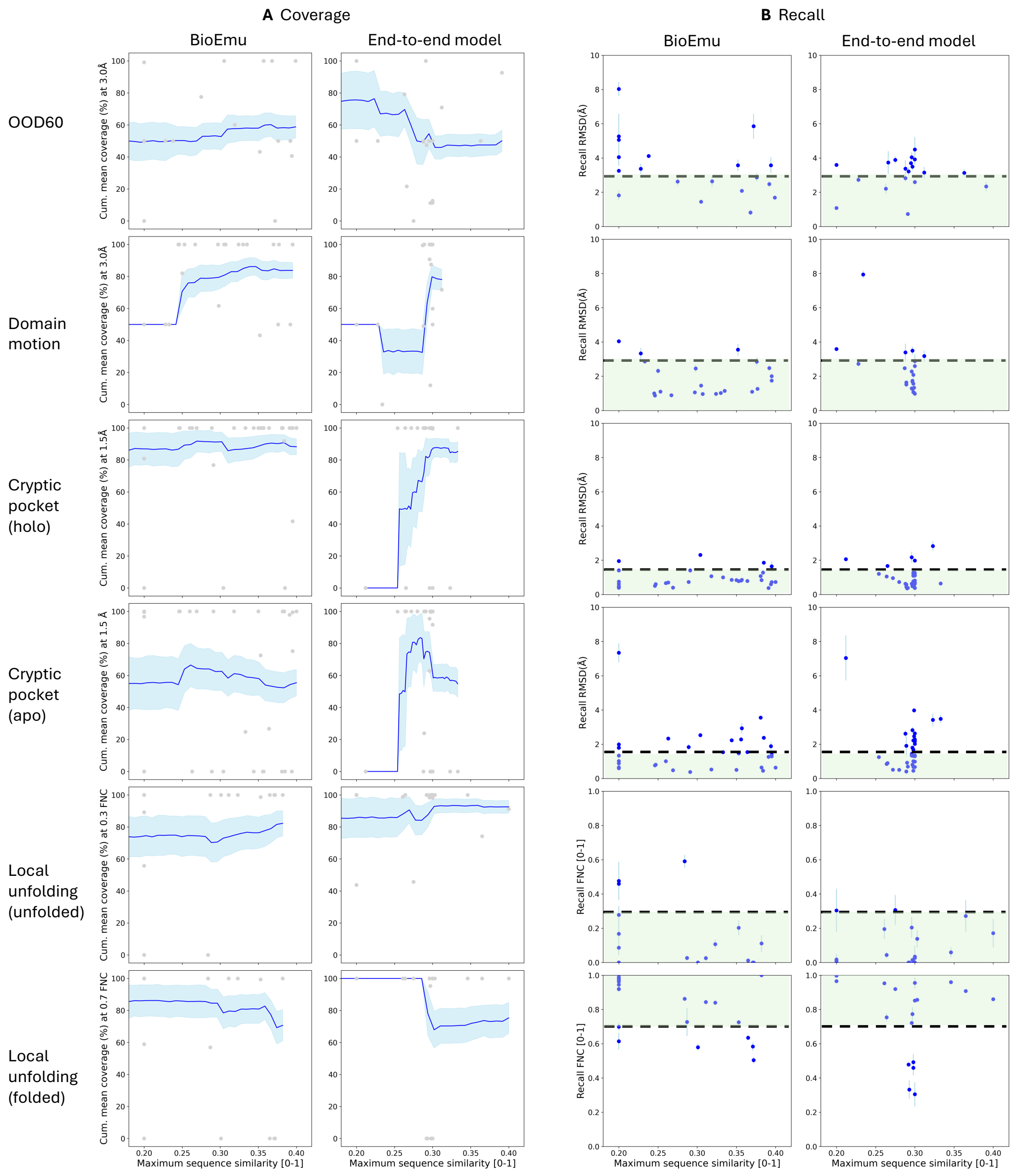

Figure S5: **Multi-conformation benchmark performance as a function of train-test sequence similarity.** (**A**) coverage and (**B**) recall (B) for the benchmarks in Figs. S1–S4, shown as a function of maximum sequence similarity to the training set. Coverage measures the fraction of test systems with at least 0.1% of samples matching the reference (3Å root mean square deviation (RMSD) for OOD60 and domain motion, 1.5 Å RMSD for cryptic pocket, fraction of native contacts (FNC) values of < 0.3 for folded and > 0.7 for unfolded states in the local unfolding benchmark, respectively). Recall measures the average score of the best 0.1% of samples. Dashed lines indicate success thresholds, green highlights indicate regions of successful sampling.

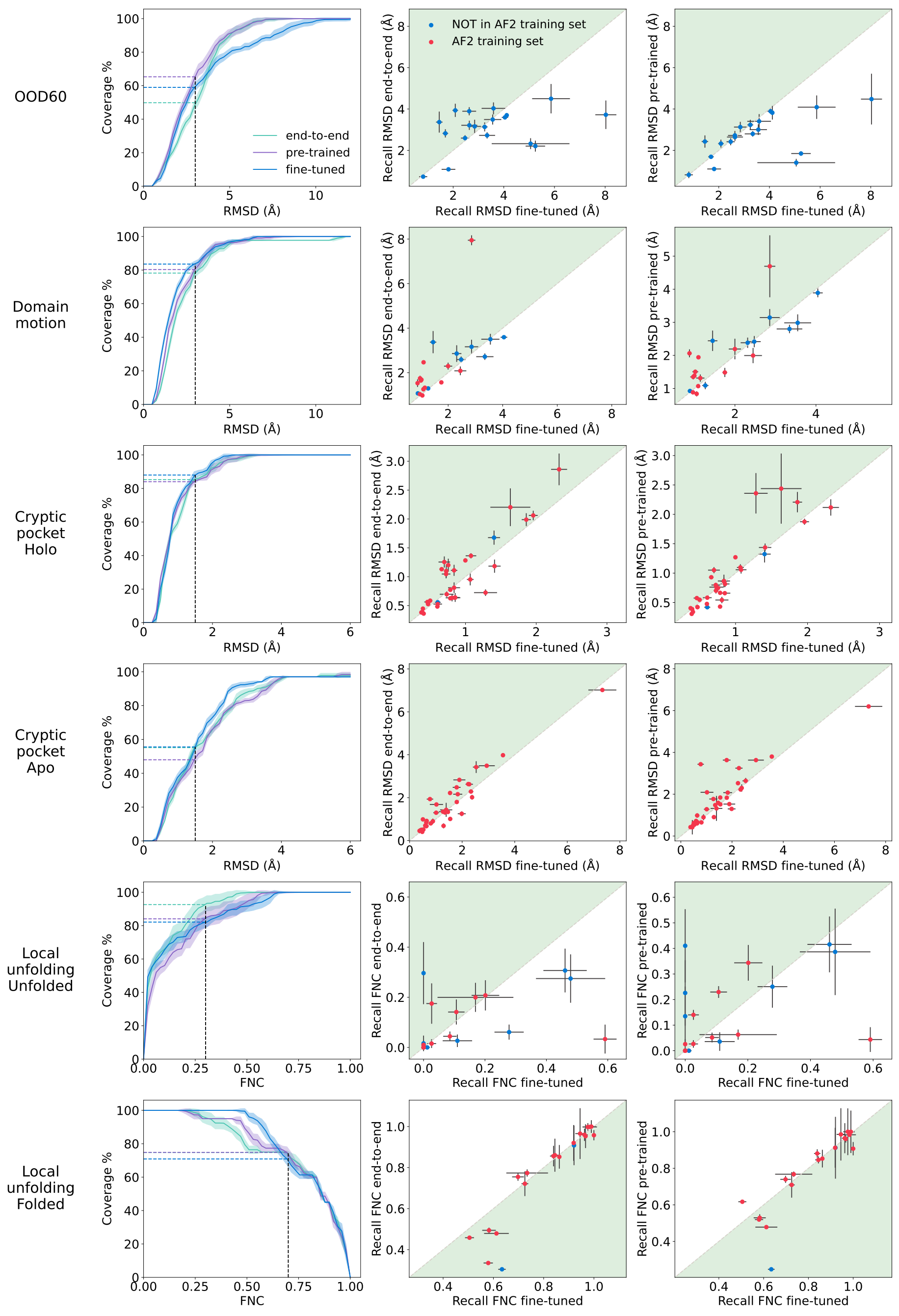

Figure S6: **Multi-conformation benchmark comparison for different BioEmu model variants.** Benchmark performance is shown for the pre-trained, fine-tuned, and end-to-end BioEmu models. The end-to-end model uses multiple sequence alignment (MSA) input instead of starting with an AlphaFold2 evoformer embedding, and maintains a strict train/test split ($\leq$ 40% sequence similarity) throughout all training stages. Coverage and recall are shown for each benchmark (Figs. S1–S4). Green highlights indicate where the fine-tuned model performs best. Abbreviations: root mean square deviation (RMSD), fraction of native contacts (FNC).

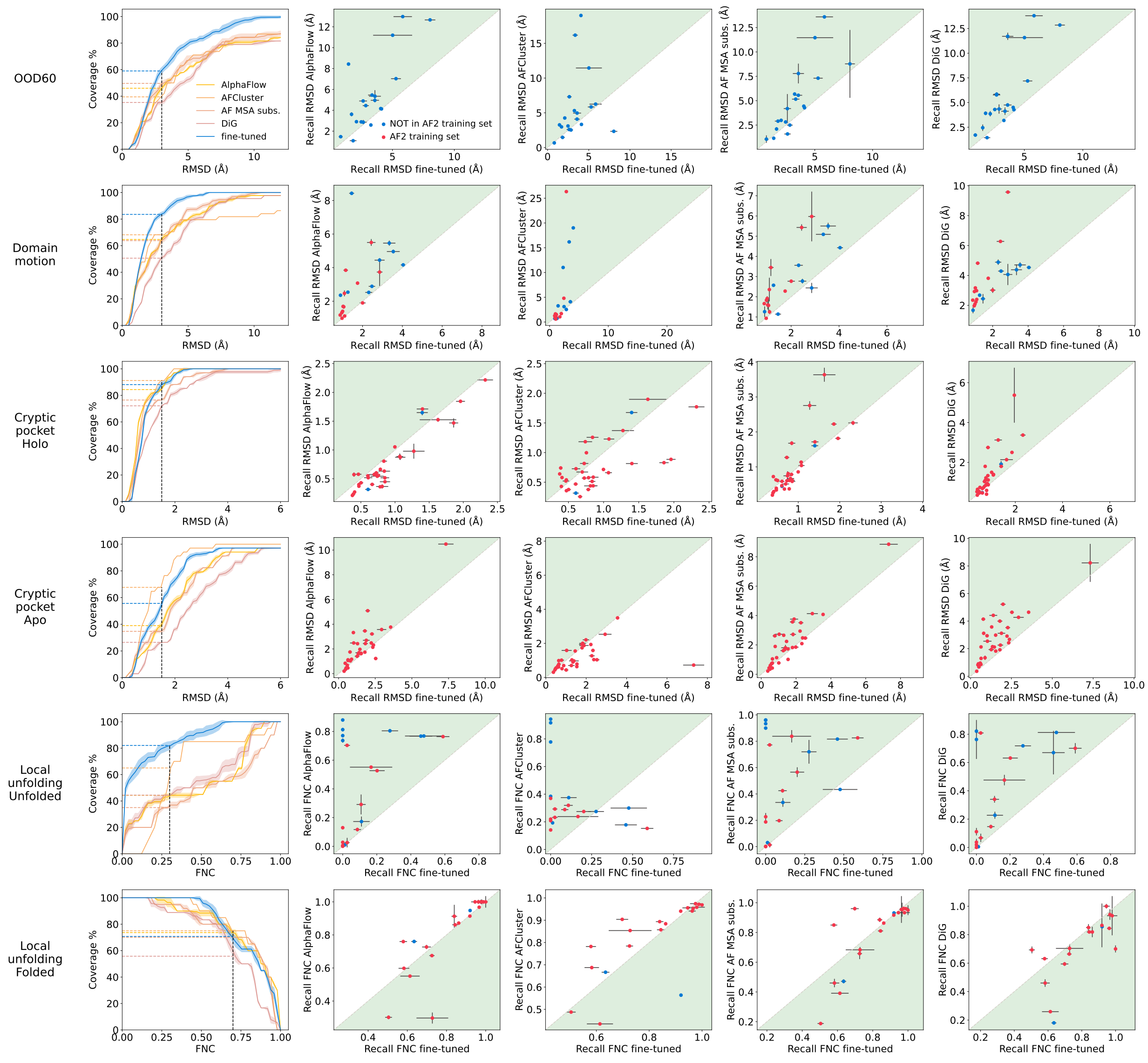

Figure S7: **Multi-conformation benchmark comparison between BioEmu and baseline models.** Comparison of fine-tuned BioEmu with AlphaFlow, AFCluster, uniform MSA subsampling, and DiG across all benchmarks (Figs. S1–S4). Left column: coverage of reference states (higher is better). Other columns: recall per benchmark entry (BioEmu on x-axis, baseline on y-axis). Green regions indicate where BioEmu outperforms the other method. Note: only OOD60 reflects strict generalization; For other benchmarks cases earlier than the AlphaFold 2 cutoff date were either directly used in training or were used to train the AlphaFold 2 evoformer representation, depending on the method. Abbreviations: root mean square deviation (RMSD), fraction of native contacts (FNC).

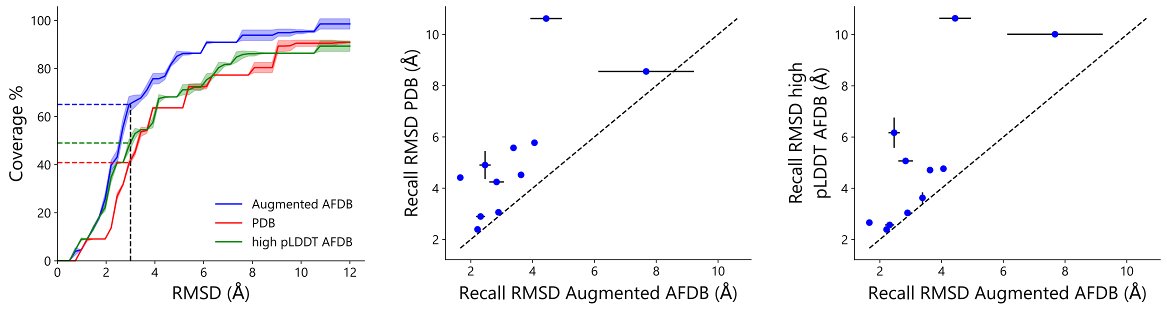

Figure S8: **Effect of pre-training data on multi-conformation sampling.** Benchmark performance on OODVal using different pre-training datasets. Left: coverage of reference states. Middle and right: recall comparisons between models trained on augmented AFDB (x-axis) vs. PDB and high-pLDDT AFDB (y-axis). Abbreviations: root mean square deviation (RMSD), fraction of native contacts (FNC).

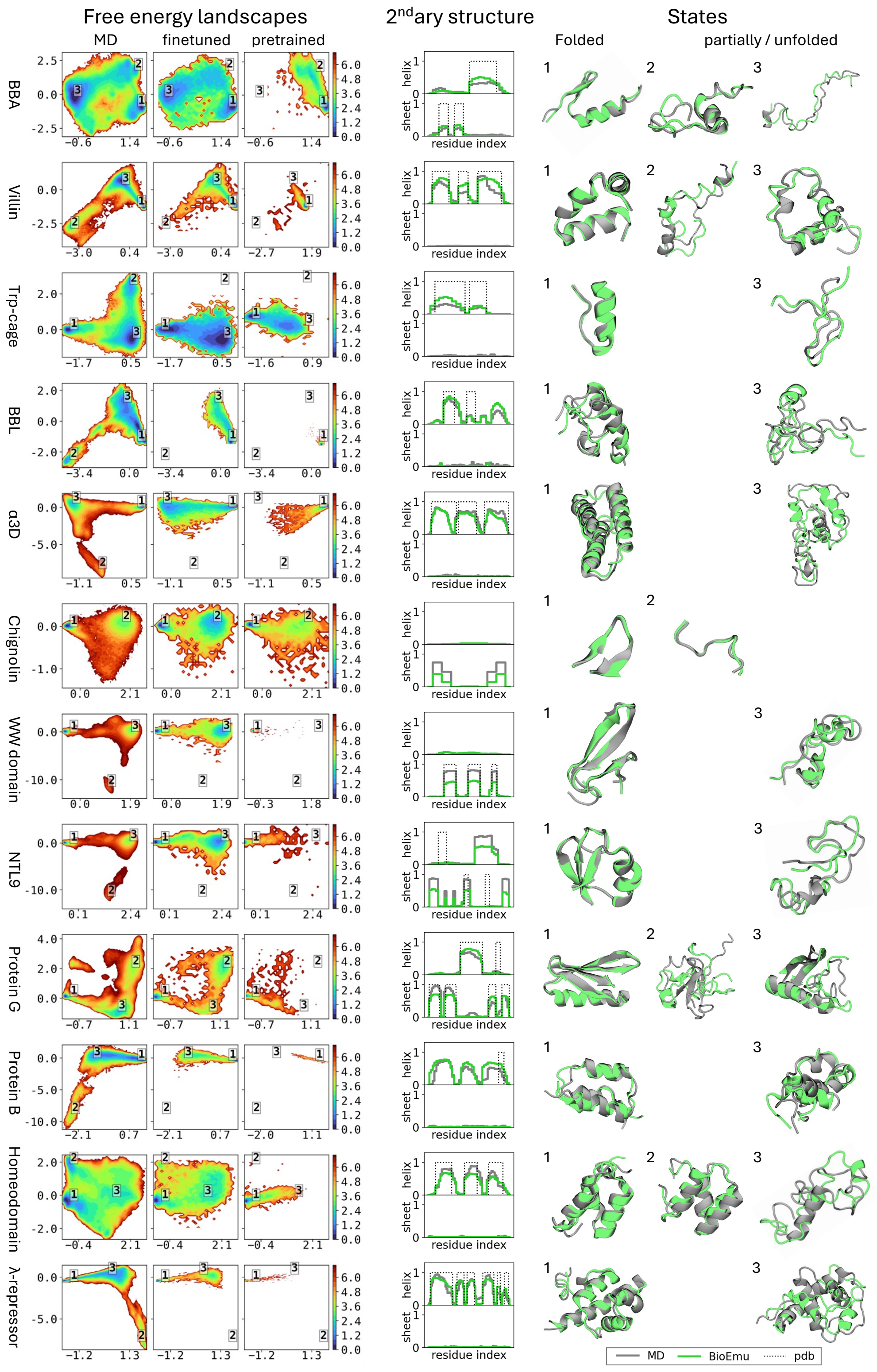

Figure S9: **Detailed results for fast-folding proteins.** Left: Free energy surfaces of MD (1st column), fine-tuned BioEmu (2nd column), and pre-trained BioEmu (3rd column) for DESRES benchmark proteins. Middle: secondary structure propensities of MD (grey) and BioEmu (green). Right: representative structures from MD (grey) and matching BioEmu samples (green) are shown where applicable.

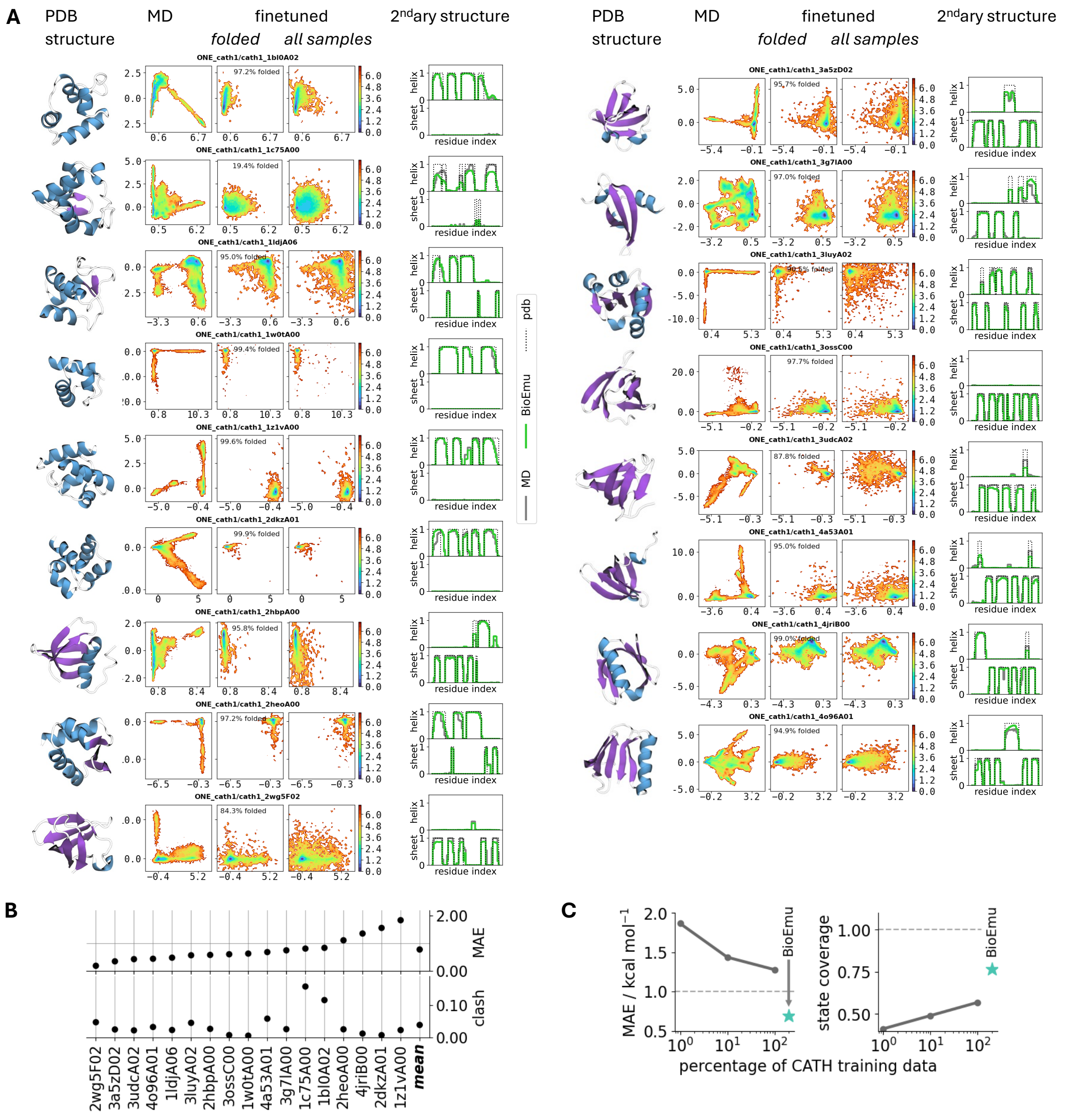

Figure S10: **Free energy surfaces and 2ndary structure propensities for CATH1 test set.** (**A**) From left to right: PDB structure. Molecular dynamcis (MD) free energy surface. BioEmu free energy surfaces shown with filtering for foldedness (excluding unfolded samples with FNC < 0.5) and without filtering. Secondary structure propensity (BioEmu in green, MD in grey, PDB as dotted line). For a definition of reference PDB codes, chains and structures see (*61*). (**B**) Mean absolute error (MAE) of free energy differences of CATH domain macrostates and fraction of unphysical samples, as shown in Fig. 3B, but with all samples instead of restricting to the folded samples. (**C**) Macrostate free energy MAE and state coverage as function of training data of a specialized CATH-only model, as shown in Fig. 3B, but with all samples instead of restricting to the folded samples. BioEmu marked with a cyan star.

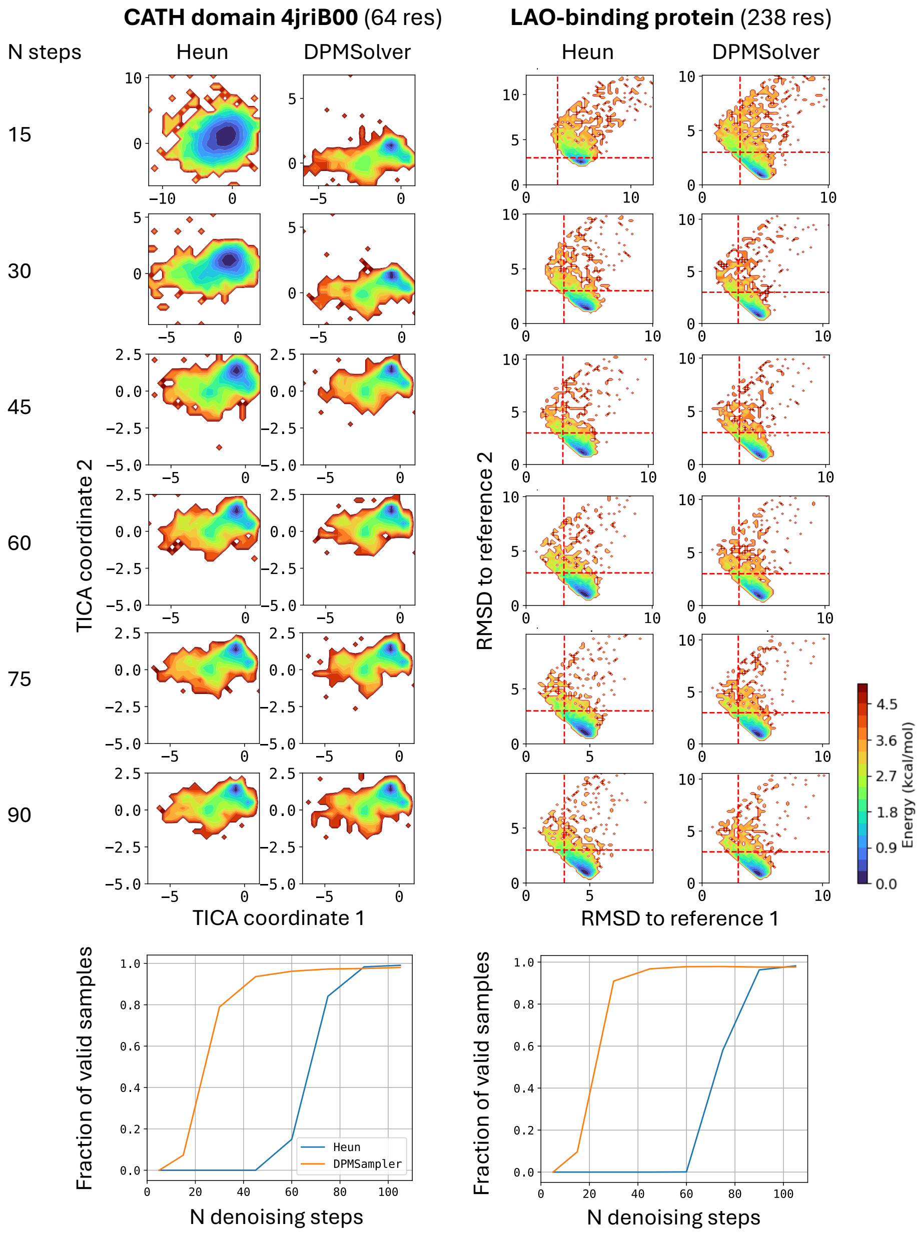

Figure S11: **Impact of sampler and denoising steps on sample quality.** For two benchmark proteins (left: 4jriB00 from the CATH1 molecular dynamics (MD) benchmark, right: LAO-binding protein from the domain motion multi-conformation benchmark), we compare the convergence of free energy surface over time-lagged independent components (TIC) and root mean square deviation (RMSD) with respect to reference structures using Heun and diffusion probabilistic model (DPM) solvers across different denoising step counts. Bottom: fraction of valid samples (no clashes or chain breaks) for each sampler configuration.

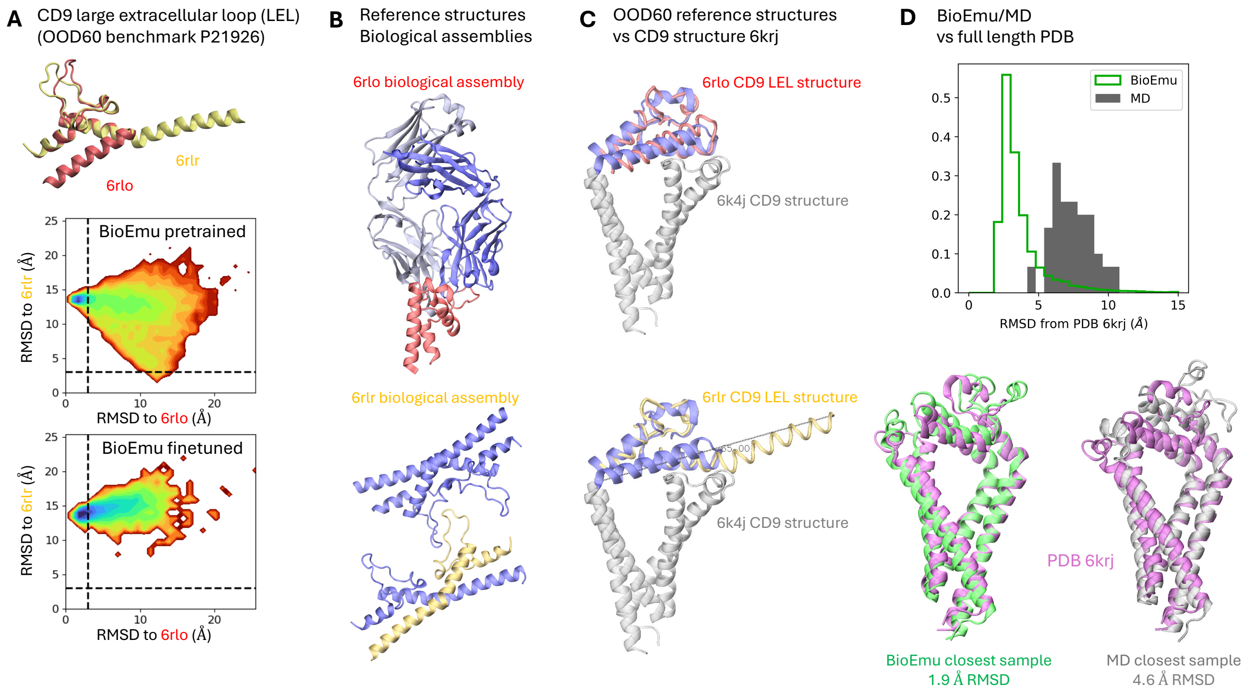

Figure S12: **Tetraspanin CD9 protein: comparison of crystallographic structures, BioEmu samples and molecular dynamics (MD) from (*37*)**. (**A**) Crystal structures 6rlo and 6rlr of the truncated large extracellular loop (LEL) that is part of the OOD60 multi-conformation sampling benchmark (see Fig. S1). Pre-trained model samples both LEL crystal structures (6rlo, 6rlr); fine-tuned BioEmu model favors 6rlo. (**B**) Biological assemblies of the CD9 LEL crystal structures: 6rlo is a complex with an antibody which binds a CD9 LEL monomer. 6rlr contains dimers in the crystal unit cell, where the long helix of each monomer swaps places with the other monomer (domain swapping), leading to a conformational change that is only stable in the dimer of the truncated LEL. (**C**) 6rlo aligns with the full-length CD9 structure (6k4j), while 6rlr is incompatible with the biologically active CD9 protein. (**D**) Comparison of full-length CD9 structures and ensembles between BioEmu, MD (*37*) and PDB 6k4j. BioEmu samples closely match 6k4j (best match 1.9 Å), outperforming MD (best match 4.6 Å). The most similar structures of MD and BioEmu to 6k4j are shown for comparison. See Table S4 for 12-letter PDB codes and original citations.

Table S1: Molecular dynamics training datasets used in this work, their associated number of systems, number of individual chains, simulation time, and forcefield used.

Dataset	**Sim. time (ms)**	**Eff. sim. time (ms)**	**Force field**	**# MD sys.**	**# ind. chains**	**Ref.**
DESRES-fastfolders	8.2	8.2	charmm22*	12	12	(*7*)
FAH-DDR1	6.8	6.8	amber ff99sb-ildn	9	9	(*74*)
FAH-SETD8	5.9	5.9	amber ff99sb-ildn	26	26	(*75*)
FAH-sarscov2	4.5	4.5	amber ff14sb	2	2	(*77*)
FAH-sarscov2-exascale	56.5	81.0	amber ff03	24	46	(*26*)
FUB-MHCII	8.9	26.2	amber ff99sb	68	201	(*78*)
FUB-barnase-barstar	2.0	4.0	amber ff99sb	2	4	(*8*)
MSR-cath2	41.0	41.0	amber ff99sb-ildn	1040	1040	
MSR-megasim	3.8	3.8	amber ff14sb & ff99sb-disp	271	271	
MSR-megasim-mutants	21.5	21.5	amber ff99sb-disp	21458	21458	
ONE-cath1	5.2	5.2	amber ff99sb-ildn	50	50	
ONE-octapeptides	8.0	8.0	amber ff99sb-ildn	1100	1100	
Total	172.2	216.0		24062	24219	

Table S2: MD dataset percentages (%) used for model fine-tuning, which define the proportion of samples drawn from a particular dataset.

MD dataset	**Amber MD finetuning**	**Property prediction finetuning**
FAH-DDR1	7.0	2.5
FAH-sarscov2	0.1	0.03
FAH-sarscov2-exascale	0.9	0.33
FAH-SETD8	6.2	2.14
FUB-MHCII	0.9	0.333
FUB-barnase-barstar	5.2	1.8
ONE-cath1	5.3	1.9
ONE-octapeptides	8.2	2.71
MSR-megasim-merge	0.3	0.5
MSR-cath2	42.5	15.2
MSR-megasim-mutants-mosaic-disp	18.7	33.56
AFDB	4.7	5.0
MegaExp	0.0	34.0
Total	100	100

Table S3: Training hyperparameters in each stage of training

Training stage	**AFDB pre-training**	**Amber MD fine-tuning**	**Property pred. fine-tuning**	**DESRES fine-tuning**
Optimizer	Adam	Adam	Adam	Adam
β_1	0.9	0.9	0.9	0.9
β_2	0.999	0.999	0.999	0.999
ε	1e-8	1e-6	1e-6	1e-8
Initial LR	1e-3	1e-4	1e-4	1e-4
LR decay factor	None	0.8	0.8	None
LR scheduler patience / epochs	None	25	25	None
EMA smoothing factor	0.995	0.999	0.999	0.995
Max residues per batch per GPU	2048	1440	1440	2048
GPU type	A100	A100	A100	A100
Number of GPUs	32	64	64	4
Days to train	~5	~2	~1.5	~3
Training epochs	60	100	100	200
Number of seen residues	~9400M	~5900M	~4400M	~4700M

Table S4: Details for PDB codes shown in the paper figures: 4-letter and 12-letter PDB access codes, citation of the original research article and present paper figure where this PDB code is used.

4-letter	12-letter	citation	Fig.	4-letter	12-letter	citation	Fig.
1ake	pdb_00001ake	(*100*)	2	3n4y	pdb_00003n4y	(*101*)	4
4ake	pdb_00004ake	(*102*)	2	3d2a	pdb_00003d2a	(*103*)	4
6ml0	pdb_00006ml0	(*104*)	2	2lzm	pdb_00002lzm	(*105*)	4
6mlp	pdb_00006mlp	(*104*)	2	1y9o	pdb_00001y9o	(*106*)	4
6pwj	pdb_00006pwj	(*107*)	2	1ril	pdb_00001ril	(*108*)	4
6pwk	pdb_00006pwk	(*107*)	2	1ifb	pdb_00001ifb	(*109*)	4
1q21	pdb_00001q21	(*110*)	2	1h7m	pdb_00001h7m	(*111*)	4
5p21	pdb_00005p21	(*112*)	2	1ekg	pdb_00001ekg	(*113*)	4
4hdd	pdb_00004hdd	(*114*)	2	1a23	pdb_00001a23	(*115*)	4
2lep	pdb_00002lep	(*116*)	2	2jws	pdb_00002jws	(*117*)	4
2vn9	pdb_00002vn9	(*118*)	2	6rlo	pdb_00006rlo	(*119*)	S12
2cey	pdb_00002cey	(*120*)	2	6rlr	pdb_00006rlr	(*121*)	S12
6h76	pdb_00006h76	(*122*)	2	6krj	pdb_00006krj	(*37*)	S12
3p53	pdb_00003p53	(*123*)	2				
6i11	pdb_00006i11	(*124*)	2				
1ecj	pdb_00001ecj	(*125*)	2				
1ecc	pub_00001ecc	(*126*)	2				

Algorithm 1 Score model $s_\theta(\mathbf{x}, \mathbf{h}, \mathbf{z}, t)$

```
Require: single representations $\mathbf{h}_i$, pair representations $\mathbf{z}_{ij}$, positions $\mathbf{r}_i$, rotations $\mathbf{Q}_i$, timestep $t$,
    relative sequence positions $p_i$
1: $\mathbf{h}_i \leftarrow$ Linear(LayerNorm($\mathbf{h}_i$)) + Sinusoidal($t$)
2: $\mathbf{z}_{ij} \leftarrow$ LinearNoBias(LayerNorm($\mathbf{z}_{ij}$)) + Embedding(Bucketize($p_i$))
3: for layer=$1, ..., 8$ do
4:     $\{\mathbf{h}_i\}$ +=Dropout(IPA($\{$LayerNorm($\mathbf{h}_i$)$\}$, $\{\mathbf{z}_{ij}\}$, $\{\mathbf{r}_i\}$, $\{\mathbf{Q}_i\}$)
5:     $\mathbf{h}_i$ +=Dropout(Linear(Dropout(gelu(Linear(LayerNorm($\mathbf{h}_i$))))))
6: end for
7: $s_r$ = Linear(relu(Linear(LayerNorm($\mathbf{h}_i$))))        ▷ translation score
8: $s_Q$ = Linear(relu(Linear(LayerNorm($\mathbf{h}_i$))))        ▷ rotation score
9: return $s_r$, $s_Q$
```

Algorithm 2 Property Prediction Fine-Tuning (PPFT)

```
Require: Score model $s_\theta$, batch of sequences $\{s_i\}$ with experimental folding free energies $\{\Delta G_i\}$,
    intermediate time step for extrapolation $\hat{t}$, number of samples $M$.
 1: for each training iteration do
 2:     Sample a batch of sequences $\{s_i\}$ with corresponding $\Delta G_i$
 3:     for each sequence $s_i$ in batch do
 4:         Compute target foldedness $f_{i,\text{target}}$ from $\Delta G_i$ using Eq. S8
 5:         for $m = 1$ to $M$ do
 6:             Sample noisy frame $\mathbf{x}_{\hat{t}}^m$ by denoising from $t = T$ to $\hat{t}$.
 7:             Detach score model when it is not in partial backprop steps.
 8:             Extrapolate to $t = 0$ to get clean sample $\hat{\mathbf{x}}_0^m$ using Eq. S14
 9:             Compute foldedness $f_m$ using Eq. S12
10:         end for
11:         Compute PPFT loss $L_{\text{ppft}}(s_i)$ using Eq. S9
12:     end for
13:     Average PPFT loss over batch to get $L_{\text{ppft}}$
14:     If simulation data available, compute score matching loss $L_{\text{sm}}$ and total loss $L$ using Eq. S13
15:     Backpropagate and update model parameters $\theta$
16: end for
```