Protein Diffusion Models as Statistical Potentials

James P. Roney
MIT Computational and Systems Biology
jamesron@mit.edu

Chenxi Ou
MIT Biology
chenxiou@mit.edu

Sergey Ovchinnikov
MIT Biology
so3@mit.edu

Abstract

Machine learning has driven rapid progress in protein structure prediction and design, but key challenges remain such as predicting protein structures when evolutionary information is unavailable, modeling full conformational landscapes, and capturing the thermodynamics of mutations and conformational changes. To address these problems we developed ProteinEBM, an Energy-Based Model of protein conformational space. ProteinEBM's energies can be used to rank protein structure correctness, predict protein structures, sample from protein conformational landscapes, and predict the energetic effects of mutations. Across all of these tasks, ProteinEBM shows performance competitive with or exceeding previous machine learning and physics-based methods. We see ProteinEBM as an important step in developing physically-grounded machine learning models for protein science.

1 Introduction

The use of machine learning has triggered a revolution in protein science and engineering, with AlphaFold often being described as a "solution" to the protein-folding problem [1]. In fact, many aspects of the problem are still open questions. For instance, AlphaFold succeeds at predicting structures for which Multiple Sequence Alignments (MSAs) contain sufficient coevolutionary signal to infer contacts, but protein structure prediction remains difficult when MSAs are shallow or nonexistent. This is a crucial limitation when designing novel proteins, and has constrained designed proteins to much simpler folds than those found in nature [2, 3]. AlphaFold also struggles to accurately predict the structural effects of mutations, since the coevolutionary signal only reflects the wildtype structure. In addition, modern ML methods cannot reliably predict the dynamical pathways by which proteins fold into their native structures. And while important progress has been made toward modeling protein conformational ensembles with machine learning, characterizing the conformational landscapes of proteins with high qualitative and quantitative accuracy, as well as designing proteins with fintuned conformational dynamics, are still an open research problems.

A theoretically general solution to these outstanding challenges and others is the development of energy-based models that characterize protein conformational landscapes, as well as efficient samplers for optimizing structures and simulating dynamics from these EBMs. More concretely, energy-based models aim to approximate a function $E_\theta(x, s)$ over protein structures x and sequences s such that

$$p(x|s) = Z_{\theta,s}^{-1} \exp\left(-E_\theta(x, s)/\beta\right)$$

Where $Z_{\theta,s}^{-1}$ is a generally unknown normalizing constant. Thermodynamically, $E_\theta(x, s)$ is a free energy corresponding to a coarse-grained description of a protein conformational state x, integrating out degrees of freedom from the solvent and fine-grained conformational states. There is a rich and highly successful history behind the development of energy functions for proteins, including models based primarily on physical laws, as in the case of molecular dynamics, or a mix of physical terms and knowledge-based terms derived from the PDB, as in the case of potentials like Rosetta [4].

Preprint.

The power and flexibility of EBMs is illustrated by the breadth of their applications. Protein structure prediction corresponds to finding the minimum energy structure $x^* = \text{argmin}_x E_\theta(x, s)$. Compared with models like AlphaFold, which predicts x^* in an end-to-end manner, the EBM formulation decouples prediction into the (easier) task of learning a discriminative energy function E_θ, followed by the application of an optimization procedure. While the optimization problem is highly nontrivial, the decoupled formulation has the advantage that the compute invested in optimization can be scaled arbitrarily at inference time in proportion to the difficulty of the prediction problem, whereas end-to-end models are fundamentally limited by the scale of the network.

EBMs can also be used to directly simulate coarse-grained protein motion via Langevin dynamics $m\ddot{x} = -\nabla_x E_\theta(x, s) - \gamma m\dot{x} + \sqrt{2\gamma m\beta}\eta(t)$. On top of this, EBMs allow for the approximation of free energy differences between conformational states, either by directly calculating energy differences $E_\theta(x_1, s) - E_\theta(x_2, s)$ or by employing enhanced sampling techniques like metadynamics or umbrella sampling. These energy differences encompass important quantities like folding stabilities and binding affinities, as well as the relative occupancies of alternative conformations. The ability to quickly estimate relative occupancies in a differentiable manner also holds great promise in protein design, since it allows sequences to be optimized for the adoption of specific energy landscapes. The breadth of EBMs' applications also allows them to be trained on diverse sources of data, including protein structures, forces from MD simulations, and experimental measurements of energy differences.

In this paper, we aim to combine the flexibility of EBMs with the expressivity of modern ML architectures. We introduce ProteinEBM, an energy-based model of protein structure and sequence. The architecture of ProteinEBM is inspired by recent advances in protein diffusion and is trained using denoising score matching, but unlike previous protein diffusion models, its learned score function is explicitly computed as the gradient of a learned energy. ProteinEBM's energy function shows robust performance across a number of tasks, including protein structure ranking, protein stability ranking, protein structure prediction, and conformational sampling. We believe models like ProteinEBM can serve as a foundation for grounding protein ML models in biophysical principles.

2 Preliminaries and Related Work

2.1 Molecular Dynamics

Molecular dynamics allows for the simulation of biomolecules with atomic resolution through the forward integration of Newtonian dynamics. The energy functions used to compute atomic forces in MD simulations are called force fields. Force fields are generally composed of simple harmonic and trigonometric terms that define the geometric constraints on bond lengths and angles, as well as electrostatic and dispersive forces mediated by Coulomb's law and Lennard-Jones potentials. The simplicity of these so-called "molecular mechanics" force fields allows for them to be computed extremely quickly, which is essential since MD simulations must run for billions of timesteps to reach useful timescales. Despite the relative simplicity of molecular mechanics force fields, MD simulations have yielded numerous important insights about the folding and function of biomolecules [5]. Recently, force fields parameterized by neural networks trained on reference quantum mechanical data have enabled simulations of some systems at far greater accuracy, but with even greater computational expense. [6].

It is also possible to use machine learning to train coarse-grained models that match the average forces of all-atom simulations on a lower-dimensional representation of a system [7, 8]. This allows for faster simulation, since the coarse-grained energy landscape is significantly smoother. However, coarse-grained models trained using force matching require large datasets of MD frames to train, and cannot learn from the wealth of experimental and predicted structures in the Protein Data Bank and AlphaFold Database.

2.2 Statistical Potentials

In the domain of protein modeling, there is a rich family of models that try to improve the accuracy of molecular mechanics potentials by explicitly incorporating terms derived from the empirical distributions of protein conformations found in the PDB. The most famous method to utilize these "statistical potential" terms is Rosetta, which has been successfully applied to a myriad of tasks in protein modeling and design [4]. In addition to conventional molecular mechanics terms and implicit

solvation terms, Rosetta has additional energies based on the Ramachandran distribution for the backbone torsions of each residue, as well as energies derived from the empirical distributions of sidechain rotameric states. Other statistical potentials include Upside, which implements a pairwise distance potential for each combination of two amino acids [9]. Upside was trained using contrastive divergence to minimize the energy of native protein structures relative to alternative samples.

Several papers have also developed creative statistical potentials using modern deep learning methods. Ingraham et al. trained a fully differentiable simulator to learn an energy function that attempts to reproduce experimental structures when optimized under Langevin dynamics [10]. While an impressive engineering feat, this meta-learning approach is computationally intensive and unstable without careful regularization, limiting scalability compared to the far simpler training paradigm of diffusion models. Yang et al. trained an autoregressive model on sequence-structure pairs, and used the model's structure likelihood as an energy function [11]. While their model has promising structure ranking capabilities, they did not demonstrate *ab initio* protein folding or structure prediction abilities. Du et al. trained a transformer-based EBM to calculate the energies of different sidechain rotameric states and achieved competitive results with Rosetta for rotamer packing accuracy, although their model is not applicable to general backbone conformations [12].

2.3 Energy-Parameterized Diffusion Models

Diffusion models have emerged as a leading class of generative models for sampling from complex distributions. These models define a forward noising process that converts samples from the data distribution into an invariant noise distribution:

$$dX = f(X,t)dt + g(t)dB_t$$

Where B_t is the standard Weiner process and $X_0 \sim p_{\text{data}}$ is drawn from the data distribution. By Anderson's theorem the reverse process is given by

$$dX = [f(X,t) - g(t)^2 \nabla_x \log p_t(X)]dt + g(t)B_t$$

Which converts samples from the noise distribution at $t = \infty$ into samples from p_{data} at $t = 0$ when integrated backward in time [13]. In a denoising diffusion model, the time-dependent score function $\nabla_x \log p_t(X)$ is approximated by a neural network. Tweedie's formula states that

$$\nabla_x \log p_t(x) = \frac{E[X_0|X_t = x] - x}{\sigma_t^2}$$

where σ_t^2 is the conditional variance of the distribution $p_t(X|X_0 = x_0)$. This provides a convenient means of training a neural network $s_\theta(x,t)$ to approximate the score function $\nabla_x \log p_t(x)$. While most diffusion models parameterize $s_\theta(x,t)$ as the direct output of a neural network, some authors have noted that is possible to parameterize the score as $s_\theta(x,t) = -\nabla_x E_\theta(x,t)$, where E_θ is a learned energy function, which can yield a number of advantages at the expense of increased computational demands from calculating second-order derivatives during training.

Du. et al. used energy-parameterized diffusion models to enable an MCMC-based correction of their diffusion sampler, which allowed them to achieve composability between multiple independent generative models [14]. Thornton et al. used energy-based diffusion models to facilitate composition and low-temperature sampling via MCMC correction, and also introduced various engineering improvements to the energy-training paradigm.[15]

Arts. et al. trained an energy-parameterized diffusion model on MD simulations of proteins to act as a coarse-grained forcefield for specific systems [16]. They found the energy-based parameterization to be crucial for obtaining a model that recapitulates atomistic simulation results, since dynamical simulations require forces to be conservative in order to remain stable. Plainer et al. extended this approach with regularization to ensure the reverse diffusion process follows the Fokker-Plank equation, promoting consistency between reverse-diffusion sampling and Langevin dynamics [17]. Jin et al. and Chu et al. both developed energy-based diffusion models for docking pairs of interacting proteins, and showed that the learned energy functions are useful approximations of binding affinity and pose correctness [18, 19]. Concurrently to our work, Sarma et al. used the continuous-time change of variables formula to compute log probabilities from a rigid-docking diffusion model, and showed that they produced energy funnels similar to Rosetta [20]. While the approach is viable for the low-dimensional generative process of rigid docking, integrating the Jacobian trace for log probability computation quickly becomes intractable for higher-dimensional processes.

2.4 Conformational Ensemble Generation

Many recent models have aimed to generate samples that reflect the conformational heterogeneity of proteins. Several works have found that subsampling the input MSA to AlphaFold results in diverse structural outputs which can reveal alternative conformations [21, 22]. While highly useful, these alternative structure predictions do not come with any quantitative estimates of relative occupancy, and do not represent unfolded or partially unfolded states. Further, MSA subsampling works by weakening subsets of the evolutionary constraints present in the MSA to produce varied results, which is only effective when the starting coevolutionary signal is sufficiently strong.

Other transferable conformation generators like BioEmu and AlphaFlow have been trained on MD data, allowing them to capture more dynamics than MSA subsampling [23, 24]. These models are very useful tools for MD acceleration, but their lack of an explicit energy function means that there is no direct way of knowing which of their generated samples are more realistic and energetically favorable. In theory, the favorability of states can be computed by simply counting their occupancies in the generated samples. However, the density of samples generated by reverse diffusion and the density implied by the energy function at low values of t can be different due to a divergence of the learned diffusion process from the Fokker-Planck equation [17]. We argue that, when training a transferable conformation generator, the density implied by the energy function at low t is more likely to be correct than the density from the naïve sampler (at least at moderately low levels of t where numerical instabilities do not hamper score learning). This is due to the simple intuition that identifying good samples is easier than generating good samples, which is supported by decades of research on structure prediction and our own experiments (see sections 4.3 and 4.4)[25]. Thus, having an energy function built into the diffusion sampler allows us to rerank and resample the structures produced by reverse diffusion, guiding us toward favorable structures that may be sparsely populated by the naïve sampler. Compared to standard diffusion models, EBMs can also be used in arbitrary enhanced sampling protocols, like MCMC-corrected sampling and replica exchange between different models, while energy-free methods are limited to enhanced sampling schemes that use additive biasing potentials [26].

Boltzmann generators use normalizing flows to model conformational ensembles [27]. In theory, Boltzmann generators are the ideal method for ensemble modeling: They provide a normalized probability density (which is strictly more powerful than an energy), and they can naturally train on both data samples and energy labels. In practice, however, the architectural constraints of normalizing flows have made it difficult to scale Boltzmann generators to large proteins, and most published models only generate small peptides [28]. We see EBMs as a middle ground that combine some of the capabilities of Boltzmann generators with more architectural flexibility.

2.5 Structure Scoring Methods

Several machine learning methods have been developed to assess the accuracy of candidate protein structures for a given sequence [29]. Methods like AF2Rank have successfully applied AlphaFold's confidence scores to identify correct proposal structures for both monomers and protein complexes [25, 30]. While the function learned by AlphaFold to identify correct protein structures may contain some physical principles, it can yield physically unrealistic results when applied to sequence design and mutation effect prediction tasks. Confidence metrics like pLDDT and pTM are also not energies in any mathematically rigorous sense (i.e., they do not approximate the log-density of any relevant distribution), which limits their ability to be used or finetuned for quantitatively accurate dynamics simulation or energy prediction tasks. While log probabilities based on AlphaFold's distogram predictions can be interpreted as energies, they require AlphaFold to have a largely correct prediction of the native structure in order to be meaningful [31]. AlphaFold-based scoring is also quite slow, especially if many scoring iterations are needed to optimize a structure prediction.

2.6 Our Contributions

In this paper, we introduce a powerful diffusion-based EBM for protein modeling. While prior energy-parameterized diffusion models have been applied to specific proteins and rigid docking tasks, we tackle the broader task of learning a universal model of protein conformational energies, with the goal of generalizing to folds and conformations that are not well represented in the PDB. Our model shows robust capabilities across a wide range of tasks, including structure ranking, stability

prediction, structure prediction, and conformational sampling. An illustration of our approach can be found in Figure 1. To our knowledge, this is the first demonstration of an ML-based EBM with the task generality expected from a model that represents protein biophysics.

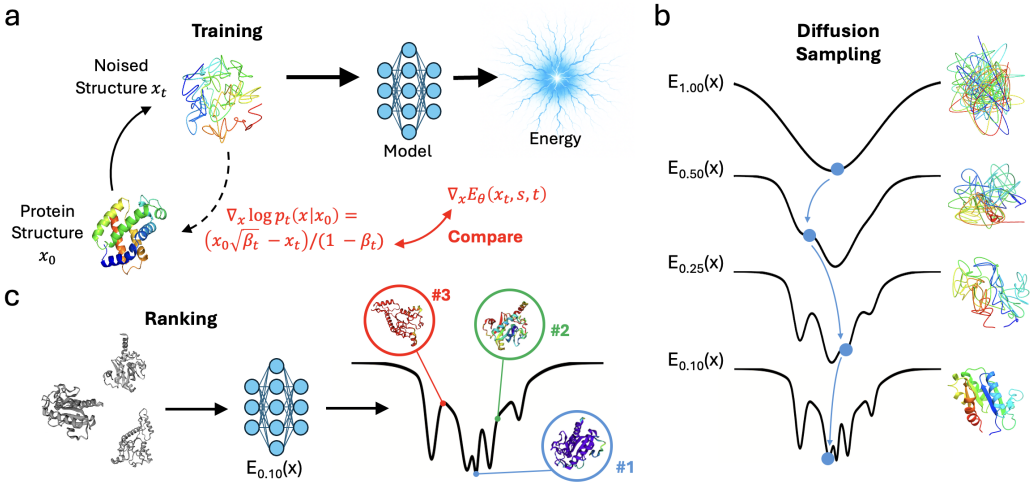

Figure 1: a.) Schematic of the score matching training process. b.) ProteinEBM can generate samples via reverse diffusion over the learned energy landscapes. c.) ProteinEBM can rank arbitrary input structures via its learned energy.

3 Methods

ProteinEBM is an energy-parameterized, sequence-conditioned protein diffusion model based on the diffusion modules from AlphaFold3 and Boltz-1 [32, 33]. This means that, unlike previous diffusion models, ProteinEBM does not utilize an inherently equivariant model architecture, and relies on data augmentations to learn 3D symmetries. We found a non-equivariant architecture to be superior to Invariant Point Attention for achieving stable denoising trajectories with an energy-parameterized model, likely because optimizing second-order derivatives through IPA can result in instability. We downscaled the model from the version used in Boltz-1, resulting in a model with 85M parameters.

We trained ProteinEBM on a set of protein domains from the PDB and AlphaFold Database, using domain definitions from CATH and The Encyclopedia of Domains (TED) [34, 35]. We pretrained on a total of 32k CATH domains and 590k AFDB domains, as well as 18k two-chain protein complexes from the set published by Jin et al. [18]. Following pretraining, we finetuned on molecular dynamics simulations of 1k CATH domains that were previously used to train BioEmu [23]. These simulations were run at 300K using various AMBER forcefields, and contain both folded and unfolded structures.

We pretrained our models for 60 epochs, with each pretraining epoch containing all 32k CATH structures, all 18k complex structures, and 32k AFDB structures. We then finetuned on 1M-3M frames of MD data, with the exact amount of finetuning varying depending on application.

ProteinEBM was trained using the denoising score matching framework. For a sample $x_0 \sim p_{data}$ and its noised version $x_t \sim \mathcal{N}(\sqrt{\beta_t}x_0, 1-\beta_t)$, the score matching loss is

$$\frac{1}{N_r}\left|\left|-\nabla_x E_\theta(x_t, s, t) - \frac{1}{1-\beta_t}\left(x_t - \sqrt{\beta_t}x_0\right)\right|\right|^2$$

Where N_r is the number of residues. Optimizing this loss drives $E_\theta(x_t, s, t)$ to equal $-\log p_t(x|s)$ up to a constant, where $p_t(x|s)$ is the sequence-conditional data density noised to level t. At low enough t, the learned energy should theoretically approach $-\log p_{data}(x|s)$.

4 Results

4.1 Decoy Ranking

Discriminating correct native structures from plausible but incorrect "decoy" structures is a fundamental capability for energy-based protein models. This capability is important for selecting models generated by structure predictors and weighting conformational ensembles produced by MSA subsampling methods, as well as for predicting protein structures in regimes where coevolutionary information is scarce. In this setting, where tools like AlphaFold can fail, investing a larger amount of compute searching over structure space for low-energy structures is a promising approach to improving prediction results [25]. This is similar to the idea of test-time scaling in Large Language Models, for which optimization over EBMs has recently been proposed as a general solution [36].

To validate ProteinEBM's ranking ability we used the Rosetta decoy set, which we further subdivided into 50% validation and 50% test structures [37]. Our training domains were filtered to have less than 40% sequence identity to proteins in either set. Additionally, we performed a structure-based split against 20 test targets by filtering out all training structures sharing a topology-level CATH classification with the targets. We refer to the proteins subjected to the CATH split as hard targets, and those subjected to the sequence split as easy targets. For decoy ranking, we used a ProteinEBM model finetuned on 1M MD frames with only the first and last transformer layers unfrozen. To rank a decoy structure x we computed $E_\theta(x, s, t)$, with the time level t becoming a hyperparameter.

We found that decoy ranking performance peaked at time levels slightly above 0, and then decayed for larger values of t. Setting $t = 0$ should theoretically make the model energy converge to the data density, but in practice at extremely low noise levels the model only needs to correct very local structural features and thus its energy does not correlate well with global structural correctness. At very high levels of t the model energy converges to a Gaussian density, so it effectively ranks decoys based on their radius of gyration. Low but nonzero values of t interpolate between these extremes and give the best performance. Based on the validation data, we selected $t = 0.1$ as the optimal value.

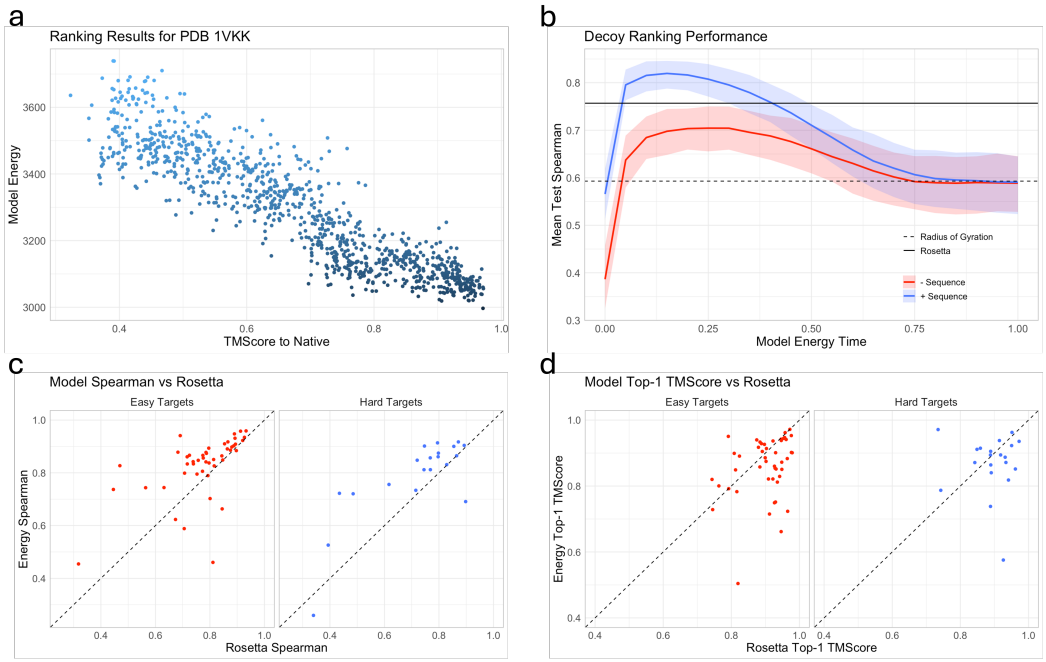

Figure 2: a.) Example correlation between model energy and decoy TMScore for a test set protein. b.) Mean Spearman correlations between energy and TMScore at different values of t. Ribbons are 95% bootstrap confidence intervals. c.) Comparison of Spearman correlations for ProteinEBM and Rosetta on each target in the test set. Dots are separated into easy and hard targets. Hard targets were subjected to a structure-based split and are farther from the training set d.) Comparison of the TMScore of the top-ranked decoy for each test target for ProteinEBM and Rosetta.

ProteinEBM performed very well on ranking decoys in the test set, with an average Spearman correlation of the 0.815 between the model energy and decoy TMScore [38]. This is substantially higher than the Rosetta energy function, which had an average Spearman of 0.757. ProteinEBM considers the sequence-structure relationship and does not merely use backbone realism for ranking, as performance degrades significantly when the sequence is masked out (Figure 2b). ProteinEBM's energy also far outperforms a simple strategy of ranking structures by the norm of the score, highlighting the necessity of the energy parameterization (Figure S1). We found no significant performance degradation on targets subjected to the stricter CATH-based split (Figure 2c-d), indicating good generalization. The average TMScore of ProteinEBM's minimum-energy decoy for each target was 0.862, which was somewhat lower than Rosetta's top-1 TMScore of 0.899. However, Rosetta must relax its input structures and pack sidechains before scoring, making it more computationally expensive than the coarse-grained ProteinEBM (Figure S4). ProteinEBM's efficiency and strong correlation between energy and structure accuracy mean that, in addition to being used as a standalone scoring function, it can be used as a fast prefilter for expensive methods like Rosetta and AF2Rank.

4.2 Stability Prediction

We next examined the ability of ProteinEBM to predict stability changes in proteins upon mutation. We did this by evaluating the energy difference

$$\Delta\Delta G = [E_\theta(x, s, t) - E_\theta(x, s', t)] - \log \frac{\sum_{x_i \sim p_{UF}} \exp\left(-E_\theta\left(x_i, s', t\right)\right) p_{UF}(x_i)^{-1}}{\sum_{x_i \sim p_{UF}} \exp\left(-E_\theta\left(x_i, s, t\right)\right) p_{UF}(x_i)^{-1}}$$

Where the right represents a Monte Carlo estimate of the free energy difference integrated uniformly over the unfolded states. We implemented p_{UF} as a Ramachandran-random sample of backbone torsion angles. This is only an approximation, and requires that the density of states near x remains nearly constant under mutation. While not fully rigorous, such an approximation is commonly used when estimating point mutation energies ([18], [7]).

We benchmarked our predicted $\Delta\Delta G$s on the stability datasets in ProteinGym, most of which were computed by Tsuboyama et al. via cDNA proteolysis [39, 40]. Our estimated energy differences showed a robust correlation with experimental values, with an average spearman correlation of 0.554 across 66 evaluated proteins. This is better than many protein language models and competitive with sequence-to-structure models like ProteinMPNN, which is among the top performers on ProteinGym with mean spearman of .565 (Figure 3). This is notable because ProteinEBM was not trained to predict the distribution of natural protein sequences, unlike ProteinMPNN.

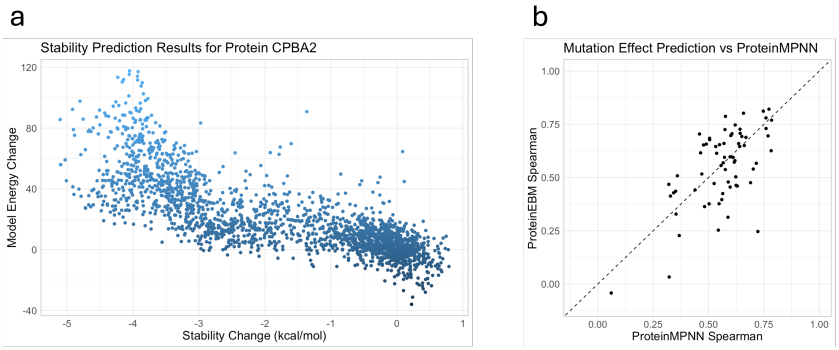

Figure 3: a.) Example correlation between predicted and experimental $\Delta\Delta G$ for protein CPBA2. b.) Spearman correlations between ProteinEBM energy differences and experimental $\Delta\Delta G$s for all stability targets in ProteinGym vs ProteinMPNN Spearmans.

4.3 Conformational Sampling

Our model can be used to generate structural samples in two ways: either by performing i.i.d. sampling via the reverse diffusion process, or by exploring the local energy landscape via dynamical simulation. The former method allows for perfectly decorrelated samples to be drawn without

concern for the convergence of the sampling process, while the latter allows for local exploration of specific regions of conformation space and the elucidation of transition paths and kinetic parameters. Simulating directly from a low-noise energy level may also reveal new energetic basins that have been learned by the low-energy model, but are never revealed in reverse diffusion due to the divergence of the higher-noise energies from the Fokker-Planck equation. We performed local sampling using Langevin dynamics, and the details of our simulation algorithm can be found in Appendix A.1.

To examine ProteinEBM's ability to predict structures and sample conformational landscapes, we examined its performance on 12 fast-folding proteins first examined by Lindorff-Larson et al. [5]. These proteins were also split from the training set at a sequence identity threshold of 40%.

To generate structural landscapes, we performed Langevin annealing trajectories, in which we started from random noise and took 100 Langevin steps at each noise level before taking a reverse diffusion step to the next noise level in the reverse diffusion process. This approach enables better exploration by allowing the samples to equilibrate to each energy level (see Appendix B.3 for analysis). To analyze our model's exploration of conformational space, we used Time-Lagged Independent Components Analysis to compare our samples to the simulations published by Majewski et al [8].

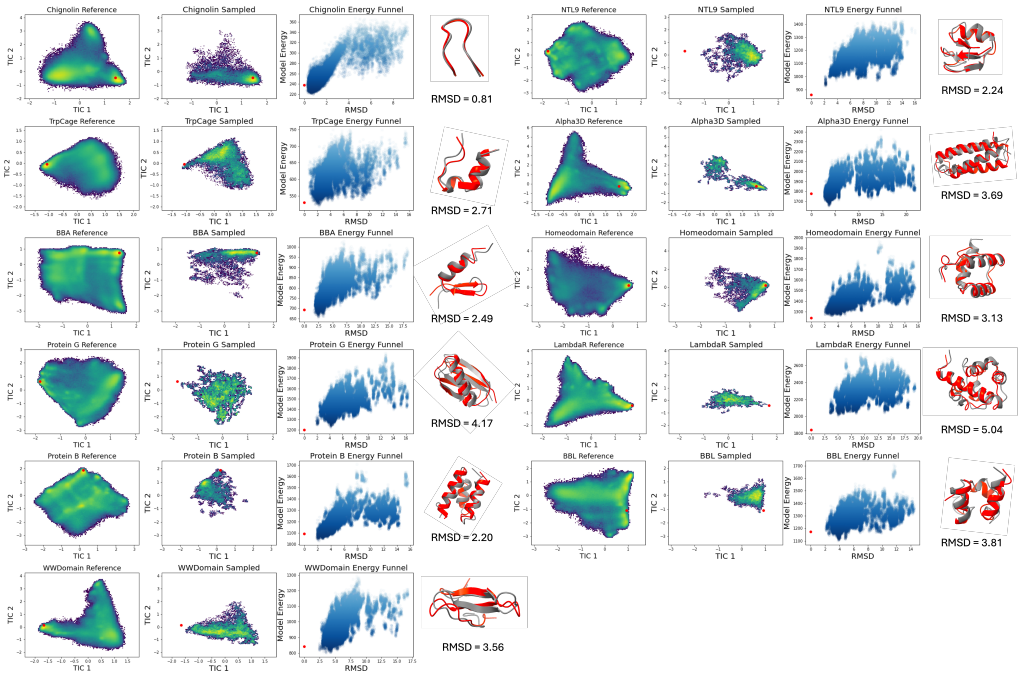

Figure 4: TICA plots showing conformational landscape for MD simulations (left) and ProteinEBM samples (right) for each fast folder, with red dot indicating native structure. Energy funnels show the relationship between model energy (with $t = 0.1$) and RMSD to native for sampled structures. For protein structure graphics, the lowest-energy ProteinEBM sample is in red, and the native is in grey.

We found that, with Langevin annealing, our model generally captured the correct native structure and explored a significant portion of the protein phase space. For all but 2 targets, the lowest-energy samples were within 4 Angstroms RMSD of the native structure. The exceptions were Protein G (4.17Å RMSD), where the minimum energy structure was a strand-swapped version of the native, and Lambda Repressor (5.04Å), which had a somewhat distorted helix packing.

Further, for some targets the lowest-energy samples were highly accurate despite the fact that most samples were unfolded or otherwise distant from the native. For example, the vast majority of sampled TrpCage structures were unfolded, but the lowest-energy sample was 2.71Å from the folded NMR structure. This highlights the power of EBMs to perform massive sampling and subsequent reranking to facilitate accurate structure prediction, which is not possible with models like BioEmu that lack an explicit energy. The energy funnels in Figure 4 are also qualitatively reminiscent of energy funnels produced by methods like Rosetta.

4.4 Structure Prediction

Finally, we tested ProteinEBM's ability to predict structures for proteins in our Rosetta test set without using MSAs. In our initial experiments on the validation set, we found that using models finetuned on on more MD data facilitated better exploration. Ultimately, we settled on a protocol of running Langevin annealing with a model finetuned on 3M frames of MD data, followed by resampling the lowest-energy structures and re-annealing with a model finetuned on 1M frames of MD data. In order to boost backbone quality, we clustered the final structures and passed the lowest-energy cluster centers to AF2Rank. Despite the high computational cost of AF2Rank, this step was computationally cheap because we scored at most 40 samples per target using AF2Rank out of thousands of total ProteinEBM samples. We selected the final prediction to be the AF2Rank output structure with the highest pTM. Details of the sampling protocol can be found in Appendix B.2.

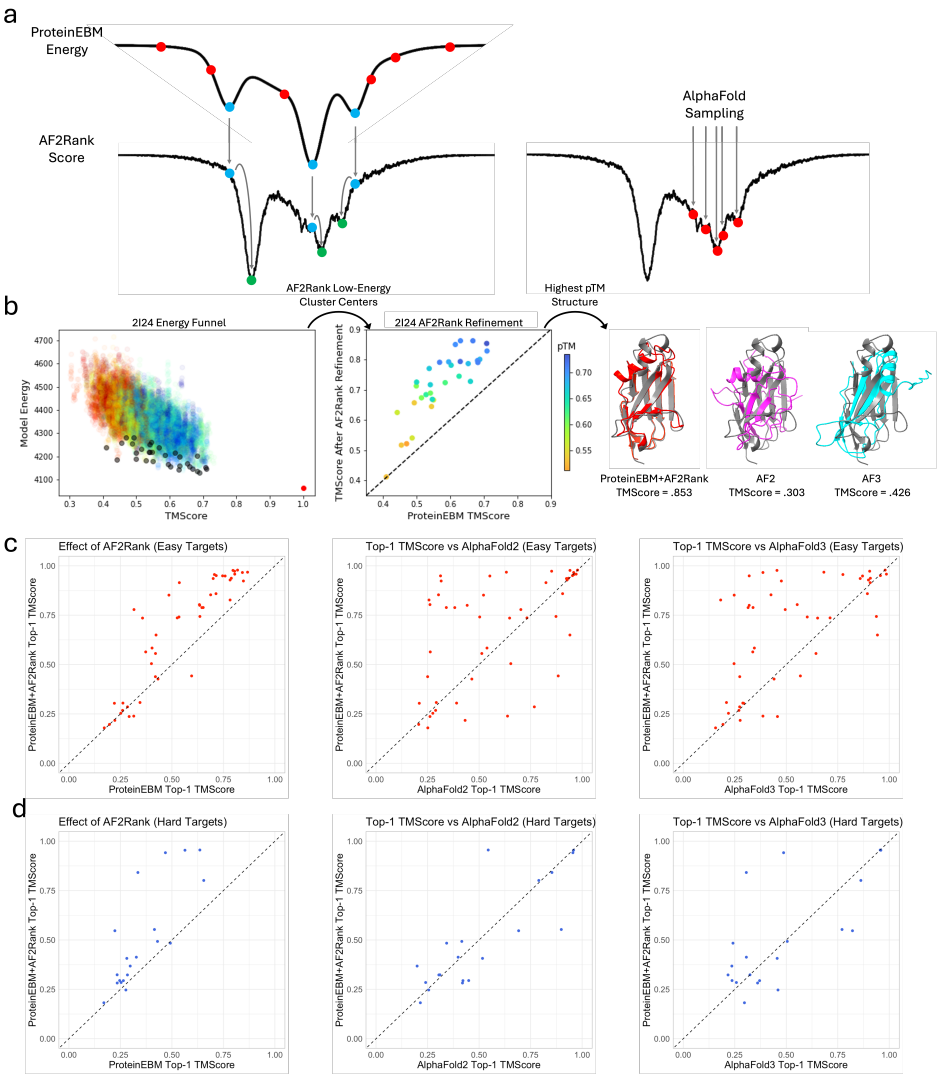

Figure 5: a.) ProteinEBM Langevin annealing (left) explores broadly, and the best samples are passed to AF2Rank. Standard AlphaFold sampling (right) can fail to sample global minima. b.) Left: TMScore vs Energy for ProteinEBM samples on PDB 2i24. Red dot is native, colors are clusters, grey dots are cluster centers. Center: AF2Rank refines and reranks lowest-energy cluster centers. Right: Predicted vs native structure for ProteinEBM+AF2Rank, AF2, and AF3. c.) Left: TMScores of best-ranked ProteinEBM outputs on easy targets vs ProteinEBM+AF2Rank. Center: TMScore of ProteinEBM+AF2Rank output vs TMScore of most confident structure from large-scale AF2 sampling. Right: Comparison against large-scale AF3 sampling. d.) Same as part c for hard targets.

To create strong baseline methods, we ran AlphaFold2 in single-sequence mode for 10,000 seeds per target, and AlphaFold3 in single-sequence mode for 15,000 seeds per target, taking the prediction with the highest pTM as the final output. We calculated these sample numbers based on empirical measurements of the runtime for each method. Details of the calculation are provided in Appendix F.

ProteinEBM sampling with AF2Rank refinement significantly outperformed both the AF2 and AF3 baselines. ProteinEBM gave an average TMScore of 0.620, compared with 0.561 for AF2 and 0.512 for AF3 (Figure 5). 60% of top-ranked ProteinEBM+AF2Rank samples had TMScore >0.5, indicating that most predicted structures were broadly correct (see Figure 5a for a successful example). It is interesting that ProteinEBM with AF2Rank refinement outperformed massive AlphaFold sampling, since the efficacy of AF2Rank demonstrates that AlphaFold's internal scoring function is at least as good as ProteinEBM at identifying correct candidate structures for the test targets. One explanation is that extensive simulation with Langevin dynamics should eventually converge and sample all of the minima in ProteinEBM's energy landscape, while running multiple AlphaFold seeds is not guaranteed to sample all of the minima in AlphaFold's scoring function, and may only generate limited diversity. This hypothesis is illustrated in Figure 5a.

Despite generally strong performance, the structure prediction results were not perfect, and exhibited multiple failure modes. Sometimes the model failed to explore structures close to the native, despite the fact that the native structure was scored with a lower energy than the structures produced by the model. In other cases, the model explored native-like structures but erroneously ranked other worse structures with lower energy. Examples of these failure cases can be found in Figure S6. These failure modes point to improvements in both the model's ranking ability and exploration ability as important areas of future work. Broader exploration could also benefit from the use of generative models that explicitly enumerate diverse protein folds.

5 Discussion and Future Directions

We have developed an energy function for protein structural landscapes that shows strong performance in structure ranking, stability ranking, protein structure prediction, and conformational sampling. Our EBM is trained using a simple denoising score-matching paradigm, and is highly efficient compared to frontier structure prediction models. Our strong results demonstrate that EBMs are a promising framework for grounding ML protein structure models in thermodynamic principles. Protein conformational preferences, stabilities, binding affinities, and coarse-grained dynamical motions can all be formulated in terms of free energies, which allows EBMs like ours to generalize to a wide array of tasks in protein modeling. And by decoupling structure scoring from sampling, we can scale computing resources arbitrarily at inference time to facilitate a broader search over structures for difficult prediction targets. This paradigm can potentially relax the dependence of structure prediction on multiple sequence alignments, thereby unlocking greater regions of protein fold space for *de novo* deisgn.

There are many possible avenues for improving the capabilities of ProteinEBM. While our model exhibits impressive unsupervised performance on $\Delta\Delta G$ prediction, large experimental stability datasets could be used to actively supervise the model's energy differences. The model could also be finetuned using contrastive divergence, with large sets of decoys generated from sampling serving as a replay buffer of contrastive samples. ProteinEBM's capabilities could be evaluated and optimized for protein complexes. Future work should also explore the use of more enhanced sampling techniques to improve exploration during structure prediction and simulation. It will also be useful to train EBMs on both finer and coarser levels of protein structure representation, such as all-atom models and models that operate on smoother latent representation spaces.

6 Code and Data Availability

All code and model weights are available at `https://github.com/jproney/ProteinEBM`.

7 Acknowledgements

James P. Roney acknowledges support from the Fannie and John Hertz Foundation. Chenxi Ou acknowledges support from the Herman N. Eisen Fellowship Fund. We thank all members of the

Ovchinnikov lab for useful discussions and feedback. Sergey Ovchinnikov acknowledges funding from NSF grant MCB2032259, CocaCola and Amgen.

References

[1] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Johannes Kohl, Alexander J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stoian Nikolov, Roberto Jain, Jonas Adler, Trevor Back, Sara Petersen, David Reiman, Emily Clancy, Michal Zielinski, Martin Steinegger, Magda Pacholska, Thomas Berghammer, Paul Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596:583–589, 2021.

[2] Tianyu Lu, Melissa Liu, Yilin Chen, Jinho Kim, and Po-Ssu Huang. Assessing generative model coverage of protein structures with shapes. *Cell Systems*, 16(8):101347, 2025.

[3] Zhidian Zhang, Chenxi Ou, Yehlin Cho, Yo Akiyama, and Sergey Ovchinnikov. Artificial intelligence methods for protein folding and design. *Current Opinion in Structural Biology*, 93:103066, 2025.

[4] Richard F. Alford, Adrien Leaver-Fay, Jason R. Jeliazkov, Michael J. O'Meara, Frank P. DiMaio, Hahnbeom Park, Maxim V. Shapovalov, Peter D. Renfrew, Vincent K. Mulligan, Kyle Kappel, Joseph W. Labonte, Marissa S. Pacella, Richard Bonneau, Philip Bradley, Roland L. Jr. Dunbrack, Rhiju Das, David Baker, Brian Kuhlman, Tanja Kortemme, and Jeffrey J. Gray. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of Chemical Theory and Computation*, 13(6):3031–3048, 2017.

[5] Kresten Lindorff-Larsen, Stefano Piana, Ron O. Dror, and David E. Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011.

[6] Justin S. Smith, Olexandr Isayev, and Adrian E. Roitberg. Ani-1: An extensible neural network potential with dft accuracy at force field computational cost. *Chemical Science*, 8:3192–3203, 2017.

[7] Nicholas E. Charron, Felix Musil, Andrea Guljas, Yaoyi Chen, Klara Bonneau, Aldo S. Pasos-Trejo, Jacopo Venturin, Daria Gusew, Iryna Zaporozhets, Andreas Krämer, Clark Templeton, Atharva Kelkar, Aleksander E. P. Durumeric, Simon Olsson, Adrià Pérez, Maciej Majewski, Brooke E. Husic, Ankit Patel, Gianni De Fabritiis, Frank Noé, and Cecilia Clementi. Navigating protein landscapes with a machine-learned transferable coarse-grained model. *Nature Chemistry*, 17(8):1284–1292, 2025.

[8] Maciej Majewski, Adrià Pérez, Philipp Thölke, Stefan Doerr, Nicholas E. Charron, Toni Giorgino, Brooke E. Husic, Cecilia Clementi, Frank Noé, and Gianni De Fabritiis. Machine learning coarse-grained potentials of protein thermodynamics. *Nature Communications*, 14(1):5739, 2023.

[9] John M. Jumper, Nabil F. Faruk, Karl F. Freed, and Tobin R. Sosnick. Trajectory-based training enables protein simulations with accurate folding and boltzmann ensembles in cpu-hours. *PLOS Computational Biology*, 14(12):e1006578, 2018.

[10] John Ingraham, Adam Riesselman, Chris Sander, and Debora Marks. Learning protein structure with a differentiable simulator. In *International Conference on Learning Representations (ICLR)*, 2019.

[11] Huan Yang, Zhaoping Xiong, and Francesco Zonta. Construction of a deep neural network energy function for protein physics. *Journal of Chemical Theory and Computation*, 18(5):3119–3131, 2022.

[12] Yilun Du, Joshua Meier, Jerry Ma, Rob Fergus, and Alexander Rives. Energy-based models for atomic-resolution protein conformations. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

[13] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[14] Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *ICML*, 2023.

[15] James Thornton, Louis Béthune, Ruixiang Zhang, Arwen Bradley, Preetum Nakkiran, and Shuangfei Zhai. Composition and control with distilled energy diffusion models and sequential monte carlo. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 258, pages 3259–3267. Proceedings of Machine Learning Research (PMLR), May 2025.

[16] Marloes Arts, Victor Garcia Satorras, Chin-Wei Huang, Daniel Zügner, Marco Federici, Cecilia Clementi, Frank Noé, Robert Pinsler, and Rianne van den Berg. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *Journal of Chemical Theory and Computation*, 19(18):6151–6159, 2023.

[17] Michael Plainer, Hao Wu, Leon Klein, Stephan Günnemann, and Frank Noé. Consistent sampling and simulation: Molecular dynamics with energy-based diffusion models, 2025.

[18] Wengong Jin, Siranush Sarkizova, Xun Chen, Nir Hacohen, and Caroline Uhler. Unsupervised protein-ligand binding energy prediction via neural euler's rotation equation. In *NeurIPS*, 2023.

[19] Lee-Shin Chu, Sudeep Sarma, and Jeffrey J. Gray. Unified sampling and ranking for protein docking with dfmdock. *bioRxiv*, 2024.

[20] Sudeep Sarma, Harrison Truscott, Da Xu, Kendall Reid, Lee-Shin Chu, Jacky Chen, and Jeffrey J. Gray. Can we extract physics-like energies from generative protein diffusion models? *bioRxiv*, 2025.

[21] Diego del Alamo, Davide Sala, Hassane S. McHaourab, and Jens Meiler. Sampling alternative conformational states of transporters and receptors with alphafold2. *eLife*, 11:e75751, 2022.

[22] Hanah K. Wayment-Steele, Ayodele Ojoawo, Sergey Ovchinnikov, Dorothee Kern, and Lucy Colwell. Predicting multiple conformations via sequence clustering with alphafold2. *Nature*, 625:832–839, 2023.

[23] Sarah Lewis, Tim Hempel, José Jiménez Luna, Michael Gastegger, Yu Xie, Andrew Y. K. Foong, Victor García Satorras, Osama Abdin, Bastiaan S. Veeling, Iryna Zaporozhets, Yaoyi Chen, Soojung Yang, Arne Schneuing, Jigyasa Nigam, Federico Barbero, Vincent Stimper, Andrew Campbell, Jason Yim, Marten Lienen, Yu Shi, Shuxin Zheng, Hannes Schulz, Usman Munir, Cecilia Clementi, and Frank Noé. Scalable emulation of protein equilibrium ensembles with generative deep learning. *Science*, 370:abn9817, 2025.

[24] Bowen Jing, Bonnie Berger, and Tommi Jaakkola. Alphafold meets flow matching for generating protein ensembles. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

[25] James P. Roney and Sergey Ovchinnikov. State-of-the-art estimation of protein model accuracy using AlphaFold. *Physical Review Letters*, 129(23):238101, November 2022.

[26] Daniel D. Richman, Jessica Karaguesian, Carl-Mikael Suomivuori, and Ron O. Dror. Unlocking hidden biomolecular conformational landscapes in diffusion models at inference time. In *NeurIPS 2025*, 2025.

[27] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

[28] Leon Klein and Frank Noé. Transferable boltzmann generators. In *NeurIPS 2024*, 2024.

[29] Naozumi Hiranuma, Hahnbeom Park, Minkyung Baek, Ivan Anishchenko, Justas Dauparas, and David Baker. Improved protein structure refinement guided by deep learning based accuracy estimation. *Nature Communications*, 12(1):1340, 2021.

[30] Nathaniel R. Bennett, Brian Coventry, Inna Goreshnik, Buwei Huang, Andrew Allen, Damian Vafeados, and David Baker. Improving de novo protein binder design with deep learning. *Nature Communications*, 14(1), 2023.

[31] Büşra Savaş, Ayşe Berçin Barlas, and Ezgi Karaca. Exploring the potential of alphafold distograms for predicting binding-induced hinge motions. *bioRxiv*, 2025.

[32] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

[33] Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Noah Getz, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Liam Atkinson, Tally Portnoi, Itamar Chinn, Jacob Silterra, Tommi Jaakkola, and Regina Barzilay. Boltz-1: Democratizing biomolecular interaction modeling. *bioRxiv*, 2024.

[34] Ian Sillitoe, Nicola Bordin, Nadia Dawson, Vaishali P. Waman, others, and Christine A. Orengo. Cath: increased structural coverage of functional space. *Nucleic Acids Research*, 49:D348–D359, 2021.

[35] Andy M. Lau, Nicola Bordin, Shaun M. Kandathil, Ian Sillitoe, Vaishali P. Waman, David T. Jones, and Christine A. Orengo. Exploring structural diversity across the protein universe with the encyclopedia of domains. *Science*, 2025.

[36] Alexi Gladstone, Ganesh Nanduru, Md Mofijul Islam, Peixuan Han, Hyeonjeong Ha, Aman Chadha, Yilun Du, Heng Ji, Jundong Li, and Tariq Iqbal. Energy-based transformers are scalable learners and thinkers. *aRxiv*, 2025.

[37] Hahnbeom Park, Philip Bradley, Per Greisen, Jr., Yuan Liu, Vikram Khipple Mulligan, David E. Kim, David Baker, and Frank DiMaio. Simultaneous optimization of biomolecular energy functions on features from small molecules and macromolecules. *Journal of Chemical Theory and Computation*, 12(12):6201–6212, 2016.

[38] Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.

[39] Kotaro Tsuboyama, Justas Dauparas, Jonathan H. Chen, Élodie Laine, Yasser Mohseni Behbahani, Jonathan J. Weinstein, Niall M. Mangan, Sergey Ovchinnikov, and Gabriel J. Rocklin. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 620(7973):434–444, 2023.

[40] Pascal Notin, Aaron W. Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Y. Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Yarin Gal, and Debora S. Marks. Proteingym: Large-scale benchmarks for protein fitness prediction and design. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 64331–64379, 2023.

[41] Joseph L. Watson, Jue Wang, Sidney Lisanza, Tamara Broderick, Brian L. Trippe, Tommi Jaakkola, Regina Barzilay, and David Baker. De novo design of protein structure and function with rfdiffusion. *Nature*, 620:225–231, 2023.

[42] Jason Yim, Brian L. Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. SE(3) diffusion model with application to protein backbone generation. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 40001–40039. PMLR, 2023.

[43] Matteo Cagiada, Sergey Ovchinnikov, and Kresten Lindorff-Larsen. Predicting absolute protein folding stability using generative models. *Protein Science*, 34(1):e5233, 2025.

[44] Peter E. Cavanagh, Andrew G. Xue, Shizhong Dai, Albert Qiang, Tsutomu Matsui, and Alice Y. Ting. Computational design of conformation-biasing mutations to alter protein functions. 2025.

A Training Details

The primary score-matching loss used for training ProteimEBM is as described in Section 3:

$$L_{DSM} = \frac{1}{N_r} \left|\left| -\nabla_x E_\theta(x_t, s, t) - \frac{1}{1-\beta_t}\left(x_t - \sqrt{\beta_t} x_0\right) \right|\right|^2$$

We parameterized the energy output as $E_\theta(x_t, s, t) = \sum_i ||r^{(i)}(x_t, s, t)||^2$, where $r^{(i)}(x_t, s, t)$ is a 3-dimensional output vector corresponding to each residue.

As an auxiliary prediction, we also added a non-conservative direct score output to the model. The auxiliary score loss was computed as:

$$L_{aux} = \frac{1}{N_r} \sum_{i=1}^{N_r} \left|\left| r_{aux}^{(i)}(x_t, s, t) - \frac{1}{1-\beta_t}\left(x_t^{(i)} - \sqrt{\beta_t} x_0^{(i)}\right) \right|\right|^2$$

Both the r and r_{aux} projections were implemented as linear projections from the final transformer layer. Finally, we introduced an all-atom modeling loss as follows:

$$x_0^{(i)} \in \mathbb{R}^{37\times 3},\ \tilde{x}_0^{(i)} = x_0^{(i)} - x_0^{(i)}[1, :]$$

$$L_{atom} = \frac{1}{N_r} \sum_{i=1}^{N_r} \frac{1}{N_a^{(i)}} \left|\left| \tilde{x}_0^{(i)} - r_{atom}^{(i)}(x_t, s, t) \right|\right|^2$$

Where N_a is the number of non-$C\alpha$ heavy atoms present in the residue. Effectively, $r_{atom}^{(i)}$ aims to predict the $C\alpha$-centered coordinates of all heavy atoms in the residue from the noised $C\alpha$ coordinates x_t. Note that, because the non-$C\alpha$ atoms are not part of the energy function or the diffusion process, predictions from the all-atom output head suffer from mean collapse and do not necessarily correspond closely to the data distribution. This can potentially be corrected by including sidechain atoms in the diffusion process or adding losses that penalize physical violations. The final training loss was computed as

$$L = 3 \cdot L_{DSM} + 0.75 \cdot L_{aux} + 0.1 \cdot L_{atom}$$

We trained using the Adam optimizer with base learning rate .0001 and a OneCycle learning rate schedule. During training the sequence information s is masked out 5% of the time. We also trained with self-conditioning 50% of the time as introduced by [41]. When self-conditioning was used, we obtained a denoised estimate $\hat{x}_0$ using the auxiliary score output from a preliminary model run, and then recycled it back into the model input for the main training step. During decoy scoring we passed the candidate structure coordinates into both the main input channel and the self-conditioning channel.

The hyperparameters for our cartesian diffusion proces are largely derived from Yim et al. [42]. The marginal distribution of our diffusion process at time t is given by $x_t \sim \mathcal{N}(\sqrt{\beta_t} x_0, 1 - \beta_t)$ where

$$\beta_t = \exp\left[-1/2\left(0.1 \cdot t + (1/2) * t^2 * (20.0 - .01)\right)\right]$$

The diffusion process and score are all defined in units of nanometers in order to make the data samples closer to unit variance.

B Sampling and Simulation

B.1 Overdamped Langevin Dynamics

For the simulation of local dynamics, we utilized overdamped Langevin dynamics on the energy surface for time-level $t = \tau$ defined as follows:

$$\frac{dx_\tau}{dt} = -\nabla E_\theta(x_\tau, s, \tau)/\gamma + \sqrt{\frac{2}{\gamma}}\eta(t)$$

Where γ is a friction coefficient and η is decorrelated Gaussian noise. When discretized as an Euler-Maruyama step, this becomes:

$$x_\tau^{(t+1)} = x_\tau^{(t)} - \nabla E_\theta(x_\tau, s, \tau)\Delta t/\gamma + \sqrt{\frac{2\Delta t}{\gamma}}\eta(t)$$

Now compare this with one step of denoising diffusion at time τ, followed by re-noising back to time τ:

$$x_{\tau-\Delta t} = \left[x_\tau + \left[\frac{1}{2}b_\tau x_\tau - b_\tau \nabla_x E_\theta(x_\tau, s, \tau)\right]\Delta t + \sqrt{b_\tau \Delta t}\eta_1(\tau)\right]$$

$$x'_\tau = x_{\tau-\Delta t} - \frac{1}{2}b_{\tau-\Delta t}x_{\tau-\Delta t}\Delta t + \sqrt{b_{\tau-\Delta t}\Delta t}\eta_2(\tau)$$

For a standard variance-preserving forward diffusion process with drift coefficient $f(X, t) = -\frac{1}{2}b_t X$ and diffusion coefficient $g(t) = \sqrt{b_t}$. In the limit of small Δt this converges to

$$x'_\tau = x_\tau - b_\tau \nabla_x E_\theta(x_\tau, s, \tau)\Delta t + \sqrt{2b_\tau \Delta t}\eta(\tau)$$

Which is equivalent to overdamped Langevin dynamics with friction coefficient $\gamma = 1/b_\tau$. Thus, we can sample from the energy surface at time τ by iteratively running reverse and forward diffusion steps with a small timestep Δt. The energy in the reverse diffusion step can also be adjusted with a correction factor T_{train}/T_{sim} that shifts the simulation temperature from the training temperature. This correspondence is analogous to the result from Arts et al. translated into the language of continuous-time diffusion [16].

B.2 Annealing Protocols

To produce the energy funnels for the fast folding proteins, we ran reverse diffusion from a Gaussian noise distribution at $t = 1$ with 200 reverse steps. For each noise level with $t < 0.5$, we ran 100 Langevin steps before taking the next reverse step to descend to the next energy level. We ran 400 such trajectories for each protein. To reduce computational cost, we used the auxiliary score output for simulation instead of the true conservative score. The Langevin samples from the lowest noise level of each trajectory were then rescored with $t = 0.1$ to produce the final funnels and TICA plots. All fast folding simulations were run with a model finetuned on $1M$ frames of MD data with only the first and last transformer layers unfrozen. We also ran the Langevin steps with a correction factor $T_{train}/T_{sim} = 300/350$ to match the fact that the fast folder reference simulations were run at $350K$, while our model was finetuned on simulations run at $300K$.

For the Rosetta data set, we used a similar annealing protocol but with additional "refinement" stages to increase sample quality. For the initial sampling stage we followed the same protocol as for the fast folders, except using a model that was finetuned on 3M frames of MD with all parameters unfrozen. For refinement, we first filtered the best 5% of samples based on model energy, and resampled those structures with replacement with Boltzmann-weighted probabilities to generate 800 starting samples. Those samples were then renoised to $t = 0.1$, and denoised by Langevin annealing with a different unfrozen model finetuned on 1M frames of MD. During this annealing phase there were 20 reverse steps, with 10 Langevin steps at each noise level. During refinement we didn't use a temperature scaling factor, and we used the conservative score from the model energy, which we found to produce more accurate structures. The refinement protocol was repeated three times, and the Langevin samples from the lowest noise level from all three refinement stages were pooled to create the final set of samples.

C Ablation of Energy for Ranking

As shown in our results, training an energy-based diffusion model results in an effective method to rank protein structures. A simpler alternative is to train an direct-score model and use the norm of the score as a pseudo-energy, which is related to the true energy:

$$\tilde{E}_\theta(x) = \frac{1}{2}||s_\theta(x)||^2, \ \nabla_x \tilde{E}_\theta(x) = s_\theta(x) \nabla_x s_\theta(x)$$

We implemented this method using the auxiliary direct score from our ProteinEBM models, and compared with the true energy. As seen in Figure S1, this method performed significantly worse.

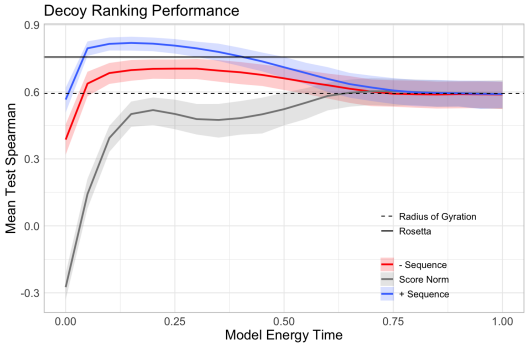

Figure S1: Rosetta test set spearman vs model energy time, including taking the norm of the score as a pseudo-energy.

D Absolute Stability Prediction

In addition to predicting $\Delta\Delta G$ of folding for mutated proteins, ProteinEBM can also be used in principle to predict absolute ΔG of folding for different proteins. To do this we computed

$$\Delta G = \beta_0 \left(-E_\theta(x, s, t) - \log \left(\frac{\sum_{x_i \sim p_{UF}} \exp\left(-E_\theta\left(x_i, s, t\right)\right) p_{UF}(x_i)^{-1}}{\sum_i p_{UF}(x_i)^{-1}} \right) \right) + \beta_1$$

using the same importance sampling estimator as in relative stability estimation. β_0 and β_1 are linear coefficients fit on top of the estimator to account for differences in units. We evaluated ProteinEBM on this task using data from Tsuboyama et al., following the analysis of Cagiada et al. [39, 43].

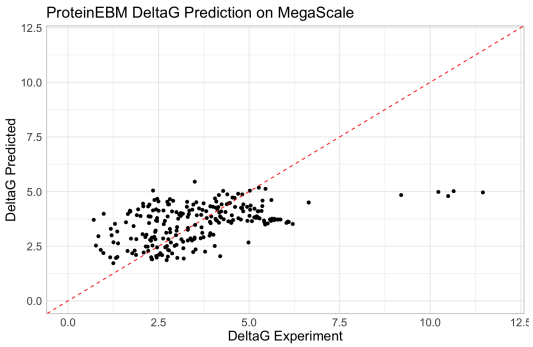

Figure S2: Evaluation of ProteinEBM ΔG prediction on MegaScale dataset.

ProteinEBM showed mediocre performance at this task, with a Spearman correlation of $.481$ and a MAE of 1.08 kcal/mol. This is significantly worse than the results from ESM-IF in Cagiada et al. ($\rho = .672, \ \text{MAE} = .96$), although it is somewhat better than the naïve baseline of predicting based on the number of amino acids ($\rho = .415, \ \text{MAE} = 1.12$). ProteinEBM could likely be improved on this task by training directly on experimental stability measurements, as was done for BioEmu [23].

E Conformational Biasing

One of the most interesting applications of EBMs is predicting the relative favorabilities of different conformational states. The free energy difference between two conformations (in units of $k_B T$) can be approximated as $-(E_\theta(x_1, s, t) - E_\theta(x_2, s, t))$. As an initial exploration of this capability, we applied ProteinEBM to predicting the free energy difference between the open and closed states for various mutants of lipoic acid ligase (LplA). Cavanaugh et al. showed that ProteinMPNN can be used to predict which mutations bias LplA toward the open or closed state [44]. They computed the score $\log\left(p(s'|x_{open})/p(s'|x_{closed})\right)$ for various mutant sequences s', with p computed using ProteinMPNN or another structure-to-sequence model. By Bayes' rule, the following identity holds:

$$\log\left(\frac{p(s'|x_{open})}{p(s'|x_{closed})}\right) = \log\left(\frac{p(x_{open}|s')}{p(x_{closed}|s')}\right) + \log\left(\frac{p(x_{closed})}{p(x_{open})}\right)$$

So the ProteinMPNN conformational biasing score is related by a constant offset to the true free energy difference $\log\left(\frac{p(x_{open}|s')}{p(x_{closed}|s')}\right)$. Therefore this score is mathematically correct for ranking purposes, but may not be accurate for predicting which conformation will dominate in absolute terms.

We conducted a similar analysis, ranking the mutations by $-(E_\theta(x_{open}, s', t) - E_\theta(x_{closed}, s', t))$, using $t = 0.1$ with the same model used for decoy ranking. Like the ProteinMPNN score, we found the ProteinEBM energy score to be positively correlated with experimental measurements of LplA promiscuity, which is correlated with time spent in the open state. Although ProteimEBM energies had significant predictive power, they were less correlated with promiscuous activity than the ProteinMPNN score.

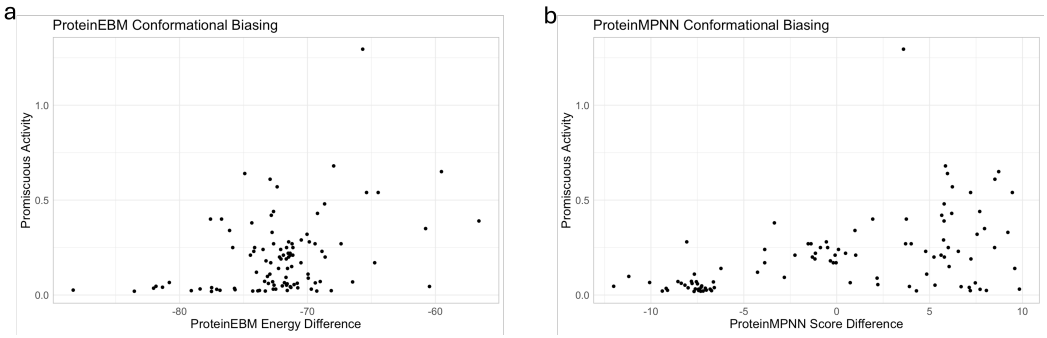

Figure S3: Correlation with LplA promiscuity for a.) ProteinEBM energy difference, b.) ProteinMPNN score

In theory the ProteinEBM score should predict the free energy differences in absolute terms, unlike ProteinMPNN. However, the data from Cavanaugh et al. only reflect conformation occupancy via the measured promiscuity of LplA activity, which is correlated with time spent in the open state but does not reveal the absolute difference in state occupancies. Future analyses should explore whether ProteinEBM's energy differences are accurate for proteins with known free energy differences between states.

F Performance Analysis

To benchmark relative computational cost, we measured the runtimes of AlphaFold3, AlphaFold2, ProteinEBM, and Rosetta on proteins of various lengths (Figure S4). To avoid comparing Jax and Pytorch models, we used the OpenFold implementation for AlphaFold2, and used Boltz-1 as a surrogate for AlphaFold3. Based on these results, we calculated the number of samples from AlphaFold3 and AlphaFold2 needed to match the computational resources used for Langevin annealing with ProteinEBM. For ProteinEBM sampling we ran 400 initial annealing trajectories, each of which consisted of 200 reverse steps and 100 Langevin steps for each noise level under 0.5. This gives $400 \times (200 + 100 \times 100) = 4,080,000$ steps (See Appendix B.2 for details). We then conducted three refinement stages in which 800 trajectories were resampled and renoised to $t = 0.1$, and then

denoised with 20 reverse steps and 10 Langevin steps per noise level for $3 \times 800 \times (20 + 20 \times 10) = 528,000$ steps. Since we used conservative scores for the refinement stages, we add an additional cost factor of 3 based on Figure S4, giving a total of $4,080,000 + 3 \times 528,000 = 5,664,000$ total effective steps.

To compute conversion factors, we used the ratios of measured runtimes for a protein of size 128, which is representative of our test set. This relative runtime factor is 2 for the AF3 diffusion model, so with 200 diffusion steps per AF3 sample the number of computationally equivalent samples is $5,664,000/2/200 = 14,160$, which we rounded up to 15,000. To generate these samples, we ran AF3 with 150 seeds per target, generating 100 independent diffusion samples from each seed.

For AF2 the relative conversion factor is approximately 200 per model iteration. We used 3 recycles, so the equivalent number of samples is $5,664,000/200/4 = 7,080$, which we rounded up to 10,000. For AF2 we ran 2000 seeds per model across the 5 available AF2 monomer models with a pTM head.

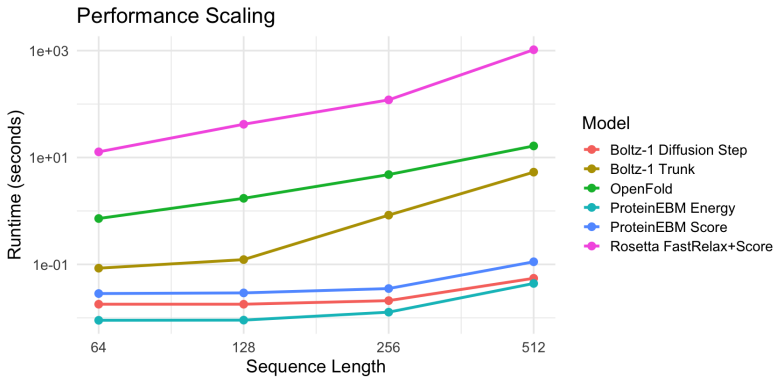

Figure S4: Runtime vs protein size for various protein modeling tools.

G Effect of Sample Quantity on Predction Accuracy

We chose to use 10,000 and 15,000 samples for our baseline comparisons against AF2 and AF3, respectively. Using bootstrapping, we estimated how top-1 accuracy increased as sample numbers were increased. The results of this analysis are shown in Figure S5. Interesting, while the top-ranked AF2 TMScore increased significantly with increased sampling, this was not the case for AF3. However, the ground-truth highest TMScore ("AF3 Oracle" in Figure S5b) did increase steadily with increased sampling. This suggests that AF3's confidence module may be less well calibrated than AF2, at least in the case of MSA-free monomer sampling.

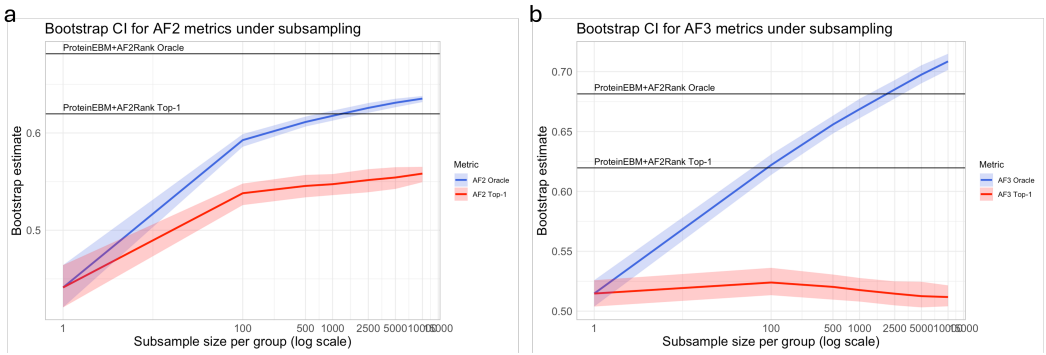

Figure S5: a.) Oracle TMScore and TMScore of top-ranked decoy for AF2 as number of samples is increased. b.) Oracle TMScore and TMScore of top-ranked decoy for AF3 as number of samples is increased. Ribbons are 95% bootstrap confidence intervals of the mean.

H Failure Modes

During structure prediction, ProteinEBM exhibited a variety of failure modes on some targets. Two examples are shown in Figure S6.

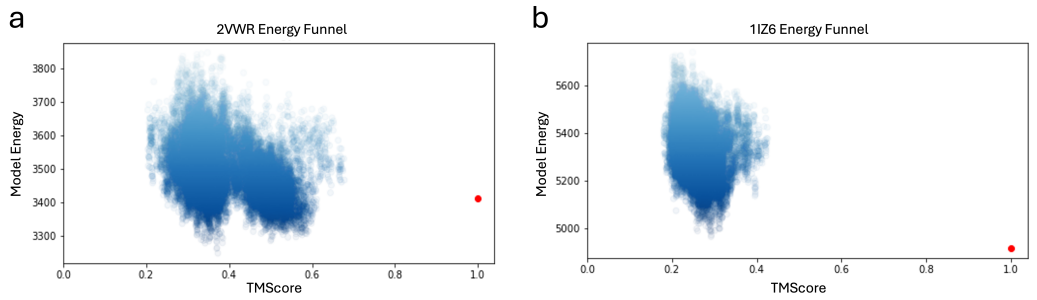

Figure S6: Energy funnels for failed prediction targets, with samples in blue and the native structure in red. a.) Example where ProteinEBM explored a structure close to the native (TMScore > 0.7), but erroneously scored a much less accurate structure with lower energy. b.) Example where the model failed to explore close to the native structure, although the native structure had a lower energy than the structures that were actually sampled.