ENERGY-BASED MODELS FOR ATOMIC-RESOLUTION PROTEIN CONFORMATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose an energy-based model (EBM) of protein conformations that operates at atomic scale. The model is trained solely on crystallized protein data. By contrast, existing approaches for scoring conformations use energy functions that incorporate knowledge of physical principles and features that are the complex product of several decades of research and tuning. To evaluate our model, we benchmark on the rotamer recovery task, a restricted problem setting used to evaluate energy functions for protein design. Our model achieves comparable performance to the Rosetta energy function, a state-of-the-art method widely used in protein structure prediction and design. An investigation of the model’s outputs and hidden representations finds that it captures physicochemical properties relevant to protein energy.

1 INTRODUCTION

Methods for the rational design of proteins make use of complex energy functions that approximate the physical forces that determine protein conformations (Cornell et al., 1995; Jorgensen et al., 1996; MacKerell Jr et al., 1998), incorporating knowledge about statistical patterns in databases of protein crystal structures (Boas & Harbury, 2007). The physical approximations and knowledge-derived features that are included in protein design energy functions have been developed over decades, building on results from a large community of researchers (Alford et al., 2017).

In this work, we investigate learning an energy function for protein conformations directly from protein crystal structure data. To this end, we propose an energy-based model based on the Transformer architecture, that accepts as inputs sets of atoms and computes an energy for their configuration. Our work is a logical extension of statistical potential methods (Tanaka & Scheraga, 1976; Sippl, 1990; Lazaridis & Karplus, 2000) that fit energetic terms from data, which, in combination with physically motivated force fields, have contributed to the feasibility of de novo design of protein structures and functions (Kuhlman et al., 2003; Ambroggio & Kuhlman, 2006; Jiang et al., 2008; King et al., 2014).

To date, energy functions for protein design have incorporated extensive feature engineering, encoding knowledge of physical and biochemical principles (Boas & Harbury, 2007; Alford et al., 2017). Learning from data circumvents the process of developing knowledge-based potential functions by automatically discovering features that contribute to the protein’s energy, including terms that are unknown or are difficult to express with rules or simple functions. Furthermore, since energy functions are additive, terms learned by neural energy-based models can be naturally composed with those proposed by expert knowledge.

In principle, neural networks have the ability to identify and represent non-additive higher order dependencies that might uncover features such as large hydrogen bonding networks. Such features have been shown to have important roles in protein structure and function (Guo & Salahub, 1998; Redzic & Bowler, 2005; Livesay et al., 2008), and are important in protein design (Boyken et al., 2016). Incorporation of higher order terms has been an active research area for energy function design (Maguire et al., 2018).

Evaluations of molecular energy functions have used, as a measure of fidelity, the ability to identify native side-chain configurations (rotamers) from crystal structures where the ground-truth configuration has been masked out (Jacobson et al., 2002; Bower et al., 1997; Leaver-Fay et al., 2013).
introduced a set of benchmarks for the Rosetta energy function that includes the task of rotamer recovery. In the benchmark, the ground-truth configuration of the side chain is masked and rotamers (possible configurations of the side-chain) are sampled and evaluated within the surrounding molecular context (the rest of the atoms in the protein structure not belonging to the side chain). The energy function is scored by comparing the lowest-energy rotamer (as determined by the energy function) against the rotamer that was observed in the empirically-determined crystal structure.

This work takes an initial step toward fully learning an atomic-resolution energy function from data. Prediction of native rotamers from their context is a restricted problem setting for exploring how neural networks might be used to learn an atomic-resolution energy function for protein design. We compare the performance of the model to that of the Rosetta energy function, as detailed in Leaver-Fay et al. (2013), and find that we obtain comparable results using our trained with deep learning. We investigate the outputs and representations of the model toward understanding its representation of molecular energies, and exploring relationships to physical properties of proteins.

Our results serve as a gateway for progress on the more general problem settings of combinatorial side chain optimization for a fixed backbone (Tuffery et al., 1995; Hinton & Salakhutdinov, 2006; LeCun et al., 2006). This is motivated by their simplicity and scalability, as well as their compelling results in other domains, such as image generation (Du & Mordatch, 2019). In EBM, a scalar parametric energy function $E_\theta(x)$ is fit to the data, with $\theta$ set through a learning procedure such that the energy is low in regions around $x$ and high elsewhere. The energy function maps to a probability density using the Boltzmann distribution: $p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z(\theta)}$, where $Z = \int \exp(-E_\theta(x))dx$ denotes the partition function.

EBMs are typically trained using the maximum-likelihood method (ML), which consists of adjusting $\theta$ to minimize $KL(p_\theta(x)||p_D(x))$, the KL-divergence between the distribution of the model and the data itself. This corresponds to maximizing the log-likelihood of the data under the model:

$$L_{ML}(\theta) = E_{x \sim p_D}[\log p_\theta(x)] = E_{x \sim p_D}[E_\theta(x) - \log Z(\theta)]$$
Following Carreira-Perpinan & Hinton (2005), the gradient of this objective can be written as:

$$\nabla_\theta L_{ML} \approx E_{x^+ \sim p_D} [\nabla_\theta E_\theta (x^+)] - E_{x^- \sim p_\theta} [\nabla_\theta E_\theta (x^-)]$$

Intuitively, this gradient decreases the energy of samples from the data distribution $x^+$ and increases the energy of samples drawn from the model $x^-$. Sampling from $p_\theta$ can be done in a variety of ways, such as MCMC or Gibbs sampling (Hinton & Salakhutdinov, 2006), possibly accelerated using Langevin dynamics (Du & Mordatch, 2019). Our method uses a simpler scheme that approximates $\nabla_\theta L_{ML}$, detailed in Section 3.4.

3 Method

Our goal is to score molecular configurations of the protein side chains given a fixed target backbone structure. To that end, we define an architecture for an energy-based model and describe its training.

3.1 Model

The model calculates scalar functions $f_\theta(A)$ of size-$k$ subsets $A$ of atoms within a protein.

Selection of atom subsets In our experiments, we choose $A$ to be nearest-neighbor sets around the residues of the protein and set $k = 64$. For a given residue, we construct $A$ to be the $k$ atoms that are nearest the position of the residue’s beta carbon.

Atom input representations Each atom in $A$ is described by its 3D Cartesian coordinates and categorical features: (i) the identity of the atom (N, C, O, S); (ii) an ordinal label of the atom in the side chain (i.e. which specific carbon, nitrogen, etc. atom it is in the side chain) and (iii) the amino acid type (which of the 20 types of amino acids the atom belongs to). The coordinates are normalized to have zero mean across the $k$ atoms. Each categorical feature is embedded into 28 dimensions, and the spatial coordinates are projected into 172 dimensions\(^1\) which are then concatenated into a 256-dimensional atom representation. The parameters for the input embeddings and projections of spatial information are learned via training. During training, a random rotation is applied to the coordinates in order to encourage rotational invariance of the model. For visualizations, a fixed number of random rotations (100) is applied and the results are averaged.

Architecture In our proposed approach, $f_\theta(A)$ takes the form of a Transformer model (Vaswani et al., 2017) that processes the set of atom representations. The self-attention layers allow each atom to attend to the representations of other atoms in the set, modeling the energy of the molecular configuration as a non-linear interaction of single, pairwise, and higher-order interactions between the atoms. The final hidden representations of the Transformer are pooled across the atoms to produce a single vector, which is finally passed to a two-layer multilayer perceptron (MLP) that produces the scalar output of the model. Figure 1 illustrates the model.

\(^1\)The high dimensionality of the spatial projection was important to ensure a high weighting on the spatial coordinates, which proved necessary for the model to train reliably.
For all experiments, we use a 6-layer Transformer with embedding dimension of 256 (split over 8 attention heads) and feed-forward dimension of 1024. The final MLP contains 256 hidden units. The models are trained without dropout and layer normalization [Ba et al., 2016] is applied before the attention blocks.

3.2 PARAMETERIZATION OF PROTEIN CONFORMATIONS

The structure of a protein can be represented by two parameterizations: (1) absolute Cartesian coordinates of the set of atoms, and (2) internal coordinates of the atoms are encoded as a set of in-plane/out-of-plane rotations and displacements relative to each atom’s reference frame. Out-of-plane rotations are parameterized by \( \chi \) angles and are the primary factor of variation between different rotamers of proteins. These coordinate systems are interchangeable.

3.3 USAGE AS AN ENERGY FUNCTION

We specify our energy function \( E_\theta(x, c) \) to take an input set composed of two parts: (1) the atoms belonging to a rotamer to be predicted, \( x \), and (2) the atoms of the surrounding molecular context, \( c \). The energy function is defined as follows:

\[
E_\theta(x, c) = f_\theta(A(x, c))
\]

where \( A(x, c) \) is the set of embeddings from \( k \) atoms nearest the position of the rotamer’s beta carbon.

3.4 TRAINING AND LOSS FUNCTIONS

In all experiments, the energy function is trained to learn the conditional distribution of the rotamer given its context by approximately maximizing the log likelihood of the data.

\[
\mathcal{L}(\theta) = -E_\theta(x, c) - \log Z_\theta(c)
\]

To estimate the partition function, we note that:

\[
\log Z_\theta(c) = \log \int e^{-E_\theta(x, c)} dx = \log (E_{q(x|c)}[e^{-E_\theta(x, c)}])
\]

for some importance sampler \( q(x|c) \). Furthermore, if we assume \( q(x|c) \) is uniformly distributed on supported configurations, we obtain a simplified maximum likelihood objective given by

\[
\mathcal{L}(\theta) = -E_\theta(x, c) - \log (E_{q(x|c)}[e^{-E_\theta(x', c)}])
\]

for some context dependent importance sampler \( q(x|c) \). We choose our sampler \( q(x|c) \) to be an empirically collected rotamer library [Shapovalov & Dunbrack Jr, 2011] conditioned on the amino acid identity and the backbone \( \phi \) and \( \psi \) angles. The rotamer library consists of lists of means and standard deviations of possible \( \chi \) angles for each 10 degree interval for both \( \phi \) and \( \psi \). We sample rotamers uniformly from this library, given by a continuous \( \phi \) and \( \psi \), by sampling from weighted mixture of Gaussians of \( \chi \) angles at each of the four surrounding bins, with weights given by distance to the bins via bilinear interpolation. Every candidate rotamer at each bin is assigned uniform probability. To ensure our context dependent importance sampler effectively samples high likelihood areas in the model, we further add the real context as a sample from \( q(x|c) \).

Training setup   Models were trained for 180 thousand parameter updates using 32 NVIDIA V100 GPUs, a batch size of 16,384, and the Adam optimizer \( (\alpha = 2 \cdot 10^{-4}, \beta_1 = 0.99, \beta_2 = 0.999) \).

4 EXPERIMENTS

4.1 DATASETS

To train our models, we constructed a curated dataset of high-resolution PDB structures using the CullPDB database, with the following criteria: resolution finer than 1.8 Å; sequence identity greater than
than 90%; and R value less than 0.25 as defined in Wang & R. L. Dunbrack (2003). To test the model on rotamer recovery, we use a test set of structures from Leaver-Fay et al. (2013). To prevent training on structures that are similar to those in the test set, we ran BLAST on sequences derived from the PDB structures and removed all train structures with more than 25% sequence identity to sequences in the test dataset. Ultimately, our train dataset consisted of 12,473 structures and our test dataset consisted of 129 structures.

4.2 Baselines

We compared to two simple baselines: a fully-connected network; and the architecture for embedding sets in the set2set paper (Vinyals et al., 2015). For the fully connected network, we map inputs to 256-dimensional vectors per atom and then concatenate all inputs and map through 3 different fully connected layers of 1024 atoms. For the set2set architecture, we use an embedding dimension of 1024 and a total of 6 processing steps before mapping average pooled representations to an energy. All models have around 10 millions parameters.

Results are also compared to Rosetta. We ran Rosetta using score12 and ref15 energy functions using the rotamer trials and rtmin protocols with default settings.

4.2.1 Evaluation

For the comparison of the model to Rosetta in Table 1 we reimplement the sampling scheme that Rosetta uses for rotamer trials evaluation. We take discrete samples from the rotamer library, with bilinear interpolation of the mean and standard deviations using the four grid points surrounding the backbone $\phi$ and $\psi$ angles for the residue. We take discrete samples of the rotamers at $\mu$, except that for buried residues we sample $\chi_1$ and $\chi_2$ at $\mu$ and $\mu \pm \sigma$ as was done in Leaver-Fay et al. (2013). We define buried residues to have $\geq 24$ $C_\beta$ neighbors within 10Å of the residue's $C_\beta$ ($C_\alpha$ for Glycine residues). For buried positions we accumulate rotamers up to 98% of the distribution, and for other positions the accumulation is to 95%. We score a rotamer as recovered correctly if all $\chi$ angles are within 20° of the ground-truth residue.

We also use a continuous sampling scheme which approximates the empirical conditional distribution of the rotamers using a mixture of Guassians with means and standard deviations computed by bilinear interpolation as above. Instead of sampling discretely, the component rotamers are sampled with the probabilities given by the library, and a sample is generated with the corresponding mean and standard deviation. This is the same sampling scheme used to train models, but with component rotamers now weighted by probability as opposed to uniform sampling.

4.3 Rotamer Recovery Results

Table 1 directly compares our EBM model (which we refer to as the Atom Transformer) with two versions of the Rosetta energy function. We run Rosetta on the set of 152 proteins from the benchmark of Leaver-Fay et al. (2013). We also include published performance on the same test set from Leaver-Fay et al. (2013). As discussed above, comparable sampling strategies are used to evaluate the models, enabling a fair comparison of the energy functions. We find that a single model evaluated on the benchmark performs slightly worse than both versions of the Rosetta energy function. An ensemble of 10 models improves the results.

Table 2 evaluates the performance of the energy function under alternative sampling strategies with the goal of optimizing recovery rates. We indicate performance of the Rosetta energy function on recovery rates using the rtmin protocol for continuous minimization. We evaluate the learned energy function with the continuous sampling from a mixture of Gaussians conditioned on the $\phi/\psi$ settings of the backbone angles as detailed above. We note that such a comparison with Rosetta is somewhat unfair, as our models are unable to utilize gradient information for sampling. We find that with ensembling the model achieves similar performance to the Rosetta energy functions. We also compare to two baselines for embedding sets with similar numbers of parameters to the Atom Transformer model and find that they have significantly weaker performance.

Buried residues are more constrained in their configurations by tight packing of the side chains within the core of the protein. In comparison, surface residues are more free to vary. Therefore
Table 1: Rotamer recovery of energy functions under the discrete rotamer sampling method detailed in Section 4.2.1. Parentheses denote value reported by Leaver-Fay et al. (2013).

<table>
<thead>
<tr>
<th>Model</th>
<th>Avg</th>
<th>Buried</th>
<th>Surface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rosetta score12 (rotamer-trials)</td>
<td>72.2</td>
<td>(72.6)</td>
<td>-</td>
</tr>
<tr>
<td>Rosetta ref2015 (rotamer-trials)</td>
<td>73.6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Atom Transformer</td>
<td>70.4</td>
<td>87.0</td>
<td>58.3</td>
</tr>
<tr>
<td>Atom Transformer (ensemble)</td>
<td>71.5</td>
<td>89.2</td>
<td>59.9</td>
</tr>
</tbody>
</table>

Table 2: Rotamer recovery of energy functions under various continuous optimization schemes. Rosetta continuous optimization is performed with the rtmin protocol. Parentheses denote value reported by Leaver-Fay et al. (2013).

<table>
<thead>
<tr>
<th>Model</th>
<th>Avg</th>
<th>Buried</th>
<th>Surface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully-connected</td>
<td>39.1</td>
<td>54.4</td>
<td>30.0</td>
</tr>
<tr>
<td>Set2set</td>
<td>43.2</td>
<td>60.3</td>
<td>31.7</td>
</tr>
<tr>
<td>Atom Transformer</td>
<td>73.1</td>
<td>91.1</td>
<td>58.3</td>
</tr>
<tr>
<td>Atom Transformer (ensemble)</td>
<td>74.1</td>
<td>91.2</td>
<td>59.5</td>
</tr>
<tr>
<td>Rosetta ref2015 (rt-min)</td>
<td>76.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Rosetta score12 (rt-min)</td>
<td>75.4</td>
<td>(74.2)</td>
<td>-</td>
</tr>
</tbody>
</table>

we break out performance on both categories. We find that the ensembled Atom Transformer has a 91.2% rotamer recovery rate for buried residues, as opposed to 59.5% for surface residues. Section 4.3 reports a detailed breakdown of the recovery rates for each residue comparing the Rosetta score12 results reported in Leaver-Fay et al. (2013) to the Atom Transformer model using the Rosetta discrete sampling method. The Atom Transformer model appears to perform well on smaller rotameric amino acids as well as polar amino acids such as glutamate/aspartate while Rosetta performs better on larger amino acids like phenylalanine, tryptophan and lysine.

4.4 Visualizing Energies

In this section, we visualize and understand how Atom Transformer models the energy of rotamers in their native contexts. We explore the response of the model to perturbations in the configuration of side chains away from their native state. We retrieve all protein structures in the test set and individually perturb rotameric $\chi$ angles across the unit circle and plot results in Figures 2, 3, and 4.

Figure 2: The energy function models distinct behavior between core and surface residues. Core residues are more sensitive to perturbations away from the native state in the $\chi_1$ torsion angle. On average, residues closer to the core have a steeper energy well.

Figure 3: There is a relation between the residue size and the depth of the energy well, with larger amino acids (e.g. Trp, Phe, Thr, Lys) having steeper wells.
Table 3: Comparison of rotamer recovery rates of different amino acids of Rosetta and Ensembled Atom Transformer under discrete rotamer sampling. The Atom Transformer model appears to perform well on polar amino acids such as glutamine, glutamate, serine, and threonine, while Rosetta performs better on larger amino acids like phenylalanine and tryptophan.

Core/Surface Energies  Figure 2 shows that steeper response to variations away from the native state is observed for residues in the core of the protein (having ≥24 contacting side-chains) than for residues on the surface (≤16), consistent with the observation that buried side-chains are tightly packed (Richardson & Richardson, 1989), while those on the surface have a more gradual response.

Rotameric Energies  Figure 3 shows relation between the residue size and the depth of the energy well, with larger amino acids having steeper wells (more sensitive to perturbations). Furthermore Figure 4 shows that the model learns the symmetries of amino acids. We find that perturbations of the $\chi_2$ angle for the residues tyr, asp, and phe are symmetric about $\chi_2$. A $180^\circ$ periodicity is observed, in contrast to the non-symmetric residues.

t-SNE Embeddings  Building on the observation of a relation between the depth of the residue and its response to perturbation from the native state, we ask whether core and surface residues are clustered within the representations of the model. To visualize the final hidden representation of the molecular contexts within a protein, we compute for each residue the final vector embedding for the 64 atom context around the carbon-$\beta$ atom (or for glycine, the carbon-$\alpha$ atom). We find that a projection of these representations by t-SNE into 2 dimensions shows a clear clustering between representations of core residues and surface residues. A representative example is shown in Figure 5.

Saliency Map  The dependence of the energy function on individual atoms can be visualized through the saliency map of the model. The 10-residue protease-binding loop in a chymotrypsin inhibitor from barley seeds is highly structured due to the presence of backbone-backbone and backbone-sidechain hydrogen bonds in the same residue (Das, 2011). We compute the energy of the 64 atom context centered around the backbone carbonyl oxygen of residue 39 (isoleucine) in PDB: 2CI2 (McPhalen & James, 1987) and derive the gradients with respect to the input atoms. Figure 6 overlays the magnitude of the gradients on the atom structure, indicating that when centered on a backbone atom, the model is paying attention to other sidechain and backbone atoms, which likely form hydrogen bonds.

5 RELATED WORK

Energy functions have been widely used in the modeling of protein conformations and the design of protein sequences and structures (Boas & Harbury, 2007). Rosetta, for example, uses a combination of physically motivated terms and knowledge-based potentials (Alford et al., 2017) to model proteins and other macromolecules.

Leaver-Fay et al. (2013) proposed optimizing the feature weights and parameters of the terms of an energy function for protein design, however their method used features and physical forces designed with expert knowledge and analysis of data. Our work draws on their development of rigorous
Figure 5: Left: 3-dimensional representation of CcmG reducing oxidoreductase (PDB ID 1KNG; Edeling et al., 2002), a protein from the test set. Atoms are colored dark blue (buried), orange (exposed), or neither (not colored). Middle: t-SNE projection of EBM hidden representation when focused on alpha carbon atom for each residue in the hidden representation. In the embedding space, buried and surface residues are distinguished.

Figure 6: Saliency map of the Atom Transformer applied to the test protein serine proteinase inhibitor (PDB ID: 2CI2; McPhalen & James, 1987). The 64 atom context is centered on the carbonyl oxygen of residue 39 (isoleucine). Atoms in the context are labeled red with color saturation $\propto$ gradient magnitude (interaction strength). Hydrogen bonds with the carbonyl oxygen are shown by dotted lines.

benchmarks for energy functions, but in contrast to Leaver-Fay et al. (2013), our method automatically learns complex features from data without prior knowledge.

Neural networks have been explored for protein folding, in which a model is tasked with predicting 3-dimensional structure. Xu (2018) developed a deep residual network that predicts the pairwise distances between residues in the protein structure from evolutionary covariation information. Senior et al. (2018) used evolutionary covariation to predict pairwise distance distributions, using maximization of the probability of the backbone structure with respect to the predicted distance distribution to fold the protein. Ingraham et al. (2018) proposed learning an energy function for protein folding by backpropagating through a differentiable simulator. AlQuraishi (2019) predicts protein structure from sequence without using co-evolution.

Deep learning has shown practical utility in the related field of small molecule chemistry. Gilmer et al. (2017) achieved state-of-the-art performance on a suite of molecular property benchmarks. Similarly, Feinberg et al. (2018) achieved state-of-the-art performance on predicting the binding affinity between proteins and small molecules using graph convolutional networks. Similarly, In contrast to our work, these methods operate over small molecular graphs and were not applied to large macromolecules, like proteins.

In parallel, recent work proposes that generative models pre-trained on protein sequences alone fine-tune well on downstream supervised tasks (Yang et al., 2019; Bepler & Berger, 2019; Alley et al., 2019; Rives et al., 2019). These methods have also been used for protein design (Wang et al., 2018). By contrast, Atom Transformer models the specific atomic coordinates of the protein.
Relatedly, generative models of protein structures have been proposed for generating protein back-bones (Anand & Huang [2018]) and to explore the inverse protein folding problem (Ingraham et al. [2019]), in which a sequence is returned for a given structure.

6 Discussion

In this work we explore the possibility of learning an energy function to represent protein conformations at an atomic level of resolution. We develop and evaluate the method in the restricted benchmark problem setting of recovering protein side chain conformations from their native context, finding that a learned energy function can perform comparably in this restricted domain to energy functions that have been developed through many years of research into approximation of the physical forces guiding protein conformation and discovery and engineering of statistical terms.

The method developed here models sets of atoms and can discover and represent the energetic contribution of high order dependencies within its inputs. We find that learning an energy function from the data of protein crystal structures automatically discovers features relevant to computing molecular energies; and we observe that the model responds to its inputs in ways that are consistent with an intuitive understanding of protein conformation and energy.

Our work explores methods for generative modeling of protein conformations. High-fidelity generative modeling of proteins can be an enabling tool for generative biology, making possible design of new protein structures and sequences. To create new proteins outside the space of those discovered by nature, it is necessary to use design principles that generalize to all proteins. [Huang et al. [2016] have argued that since the physical principles that govern protein conformation apply to all proteins, encoding knowledge of these physical and biochemical principles into an energy function will make it possible to design de novo new protein structures and functions that have not appeared before in nature.

Learning features from data with generative methods is a possible direction for realizing this goal to enable design in the large space of sequences not visited by evolution. The generalization of neural energy functions to harder problem settings used in the protein design community, e.g. combinatorial side-chain optimization (Tuffery et al. [1991], Holm & Sander [1992], and inverse-folding (Pabo, [1983]), opens up a direction for future work. The methods explored here have the potential for immediate extension into these settings.

References


