# ENERGY-BASED MODELS FOR ATOMIC-RESOLUTION PROTEIN CONFORMATIONS

**Yilun Du***
Massachusetts Institute of Technology
Cambridge, MA
yilundu@mit.edu

**Joshua Meier**
Facebook AI Research
New York, NY
jmeier@fb.com

**Jerry Ma**
Facebook AI Research
Menlo Park, CA
maj@fb.com

**Rob Fergus**
Facebook AI Research & New York University
New York, NY
robfergus@fb.com

**Alexander Rives**
New York University
New York, NY
arives@cs.nyu.edu

## 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 the model, we benchmark on the rotamer recovery task, the problem of predicting the conformation of a side chain from its context within a protein structure, which has been used to evaluate energy functions for protein design. The model achieves performance close to that of 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[1], we investigate learning an energy function for protein conformations directly from protein crystal structure data. To this end, we propose an energy-based model using the Transformer architecture (Vaswani et al., 2017), 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

*Work performed during an internship at Facebook

[1]Data and code for experiments are available at https://github.com/facebookresearch/protein-ebm

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 can circumvent 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. Since energy functions are additive, terms learned by neural energy-based models can be naturally composed with those derived from physical knowledge.

In principle, neural networks have the ability to identify and represent non-additive higher order dependencies that might uncover features such as 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 within a protein 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 model to the Rosetta energy function, as detailed in Leaver-Fay et al. (2013), and find that on the rotamer recovery task, deep learning-based models obtain results approaching the performance of Rosetta. 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 open for future work the more general problem settings of combinatorial side chain optimization for a fixed backbone (Tuffery et al., 1991; Holm & Sander, 1992) and the inverse folding problem (Pabo, 1983) – the recovery of native sequences for a fixed backbone – which has also been used in benchmarking and development of molecular energy functions for protein design (Leaver-Fay et al., 2013).

## 2 BACKGROUND

**Protein conformation** Proteins are linear polymers composed of an alphabet of twenty canonical amino acids (residues), each of which shares a common backbone moiety responsible for formation of the linear polymeric backbone chain, and a differing side chain moiety with biochemical properties that vary from amino acid to amino acid. The energetic interplay of tight packing of side chains within the core of the protein and exposure of polar residues at the surface drives folding of proteins into stable molecular conformations (Richardson & Richardson, 1989; Dill, 1990).

The conformation of a protein can be described through two interchangeable coordinate systems. Each atom has a set of spatial coordinates, which up to an arbitrary rotation and translation of all coordinates describes a unique conformation. In the internal coordinate system, the conformation is described by a sequence of rigid-body motions from each atom to the next, structured as a kinematic tree. The major degrees of freedom in protein conformation are the dihedral rotations (Richardson & Richardson, 1989), about the backbone

bonds termed *phi* ($\phi$) and *psi* ($\psi$) angles, and the dihedral rotations about the side chain bonds termed *chi* ($\chi$) angles.

Within folded proteins, the side chains of amino acids preferentially adopt configurations that are determined by their molecular structure. A relatively small number of configurations separated by high energetic barriers are accessible to each side chain (Janin et al., 1978). These configurations are called *rotamers*. In Rosetta and other protein design methods, rotamers are commonly represented by libraries that estimate a probability distribution over side chain configurations, conditioned on the backbone $\phi$ and $\psi$ torsion angles. We use the Dunbrack library (Shapovalov & Dunbrack Jr, 2011) for rotamer configurations.

**Energy-based models** A variety of methods have been proposed for learning distributions of high-dimensional data, e.g. generative adversarial networks (Goodfellow et al., 2014) and variational autoencoders (Kingma & Welling, 2013). In this work, we adopt energy-based models (EBMs) (Dayan 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 EBMs, 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: $p_\theta(x) = \exp(-E_\theta(x))/Z(\theta)$, where $Z = \int \exp(-E_\theta(x))\, dx$ denotes the partition function.

EBMs are typically trained using the maximum-likelihood method (ML), in which $\theta$ is adjusted to minimize $\mathrm{KL}(p_D(x)||p_\theta(x))$, the Kullback-Leibler divergence between the data and the model distribution. This corresponds to maximizing the log-likelihood of the data under the model:

$$L_{\mathrm{ML}}(\theta) = \mathbb{E}_{x\sim p_D}[\log p_\theta(x)] = \mathbb{E}_{x\sim p_D}[E_\theta(x) - \log Z(\theta)]$$

Following Carreira-Perpinan & Hinton (2005), the gradient of this objective can be written as:

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

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 Markov chain Monte Carlo or Gibbs sampling (Hinton & Salakhutdinov, 2006), possibly accelerated using Langevin dynamics (Du & Mordatch, 2019). Our method uses a simpler scheme to approximate $\nabla_\theta L_{\mathrm{ML}}$, 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. We define an architecture for an energy-based model and describe its training procedure.

### 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 to 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 $k$

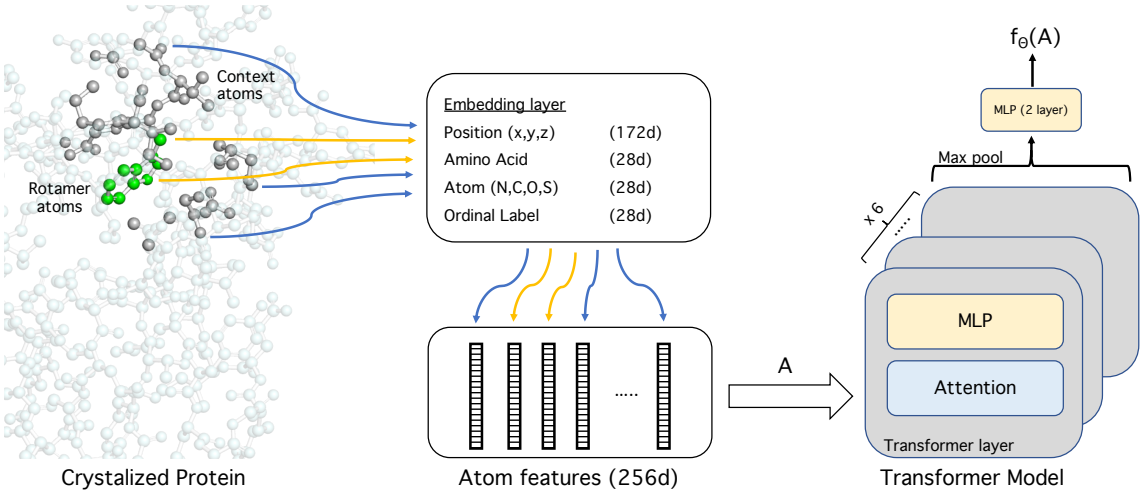


Figure 1: Overview of the model. The model takes as input a set of atoms, $A$, consisting of the rotamer to be predicted (shown in green) and surrounding atoms (shown in dark grey). The Cartesian coordinates and attributes of each atom are embedded. The set of embeddings is processed by Transformer blocks, and the final hidden representations are pooled over the atoms to produce a vector. The vector is passed through a two-layer MLP to output a scalar energy value, $f_\theta(A)$.

atoms. Each categorical feature is embedded into 28 dimensions, and the spatial coordinates are projected into 172 dimensions[2], 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 a 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.

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. 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 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 which are the primary degrees of freedom in the rotamer configurations. The coordinate systems are interchangeable.

[2]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.

### 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 to 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:

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

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

$$\mathcal{L}(\theta) = -E_\theta(x, c) - \log(\mathbb{E}_{q(x^i|c)}[e^{-E_\theta(x^i,c)}])$$

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. We write the importance sampler as a function of atomic coordinates which are interchangeable with the angular coordinates in the rotamer library. The 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 a 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 ($\alpha = 2 \cdot 10^{-4}$, $\beta_1 = 0.99$, $\beta_2 = 0.999$). We evaluated training progress using a held-out 5% subset of the training data as a validation set.

## 4 EXPERIMENTS

### 4.1 DATASETS

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 less 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 the 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.

| Model | Avg | Buried | Surface |
|---|---|---|---|
| Rosetta score12 (rotamer-trials) | 72.2 (72.6) | - | - |
| Rosetta ref2015 (rotamer-trials) | 73.6 | - | - |
| Atom Transformer | 70.4 | 87.0 | 58.3 |
| Atom Transformer (ensemble) | 71.5 | 89.2 | 59.9 |

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).

### 4.2 BASELINES

We compare to three baseline neural network architectures: a fully-connected network, the architecture for embedding sets in the set2set paper (Vinyals et al., 2015); and a graph neural network (Veličković et al., 2017). All models have around 10 million parameters. Details of the baseline architectures are given in Appendix A.1.2.

Results are also compared to Rosetta. We ran Rosetta using score12 and 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 Gaussians 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

| Model | Avg | Buried | Surface |
|---|---|---|---|
| Fully-connected | 39.1 | 54.4 | 30.0 |
| Set2set | 43.2 | 60.3 | 31.7 |
| GraphNet | 69.0 | 94.3 | 54.2 |
| Atom Transformer | 73.1 | 91.1 | 58.3 |
| Atom Transformer (ensemble) | 74.1 | 91.2 | 59.5 |
| Rosetta score12 (rt-min) | 75.4 (74.2) | - | - |
| Rosetta ref2015 (rt-min) | 76.4 | - | - |

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

| Amino Acid | R | K | M | I | L | S | T | V |
|---|---|---|---|---|---|---|---|---|
| Atom Transformer | 37.2 | 31.7 | 53.0 | 93.3 | 82.6 | 79.0 | 96.5 | 94.0 |
| Rosetta score12 | 26.7 | 31.7 | 49.6 | 85.4 | 87.5 | 72.5 | 92.6 | 94.3 |
| Amino Acid | N | D | Q | E | H | W | F | Y |
| Atom Transformer | 67.4 | 76.0 | 40.8 | 49.8 | 65.5 | 83.5 | 80.3 | 77.6 |
| Rosetta score12 | 56.8 | 60.4 | 30.7 | 33.6 | 55.0 | 85.0 | 85.4 | 82.9 |

Table 3: Comparison of rotamer recovery rates by amino acid between Rosetta and the ensembled energy-based model under discrete rotamer sampling. The model appears to perform well on polar amino acids glutamine, serine, asparagine, and threonine, while Rosetta performs better on larger amino acids phenylalanine, tyrosine, and tryptophan and the common amino acid, leucine. The numbers reported for Rosetta are from Leaver-Fay et al. (2013).

above. We find that with ensembling the model performance is close to that of the Rosetta energy functions. We also compare to three baselines for embedding sets with similar numbers of parameters to the Atom Transformer model and find that they have 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 we also report performance separately on both categories. We find that the ensembled Atom Transformer has a 91.2% rotamer recovery rate for buried residues, compared to 59.5% for surface residues.

Table 3 reports recovery rates by 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 and tryptophan and more common ones like leucine.

### 4.4 Visualizing Energies

In this section, we visualize and understand how the 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, plotting results in Figures 2, 3, and 4.

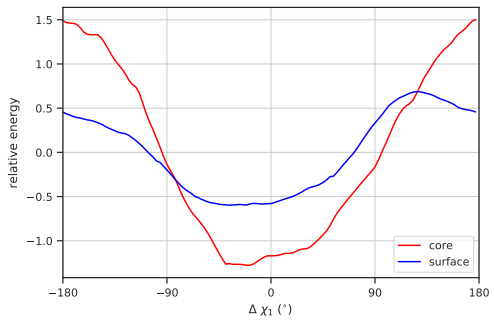


Figure 2: The energy function models distinct behavior between core and surface residues. Core residues are more sensitive to perturbations away from the native state in the $\chi_1$ torsion angle. On average, residues closer to the core have a steeper energy well.

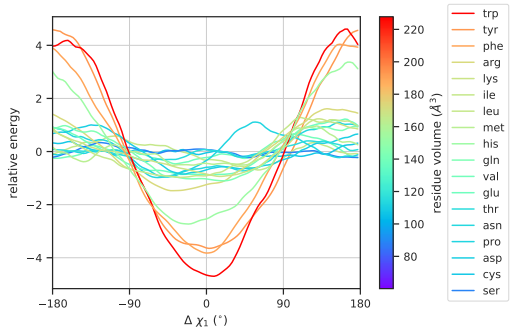


Figure 3: There is a relation between the residue size and the depth of the energy well, with larger amino acids (e.g. Trp, Phe, Thr, Lys) having steeper wells.

**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 $\geq$24 contacting side chains) than for residues on the surface ($\leq$16), consistent with the observation that buried side chains are tightly packed (Richardson & Richardson, 1989).

**Rotameric Energies** Figure 3 shows a 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 responses to perturbations of the $\chi_2$ angle for the residues Tyr, Asp, and Phe are symmetric about $\chi_2$. A $180°$ periodicity is observed, in contrast to the non-symmetric residues.

**Embeddings of Atom Sets** 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 the final vector embedding for the 64 atom context around the carbon-$\beta$ atom (or for glycine, the carbon-$\alpha$ atom) for each residue. We find that a projection of these representations by t-SNE (Maaten & Hinton, 2008) into 2 dimensions shows a clear clustering between representations of core residues and surface residues. A representative example is shown in Figure 5.

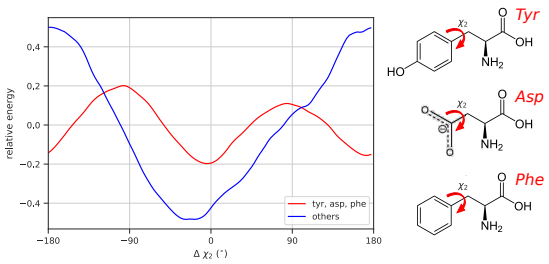


Figure 4: Note the periodicity for the amino acids Tyr, Asp, and Phe with terminal symmetry about $\chi_2$.

**Saliency Map** 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). To visualize the dependence of the energy function on individual atoms, 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 gradient magnitudes on the structure, indicating the model attends to both sidechain and backbone atoms, which participate in hydrogen bonds.

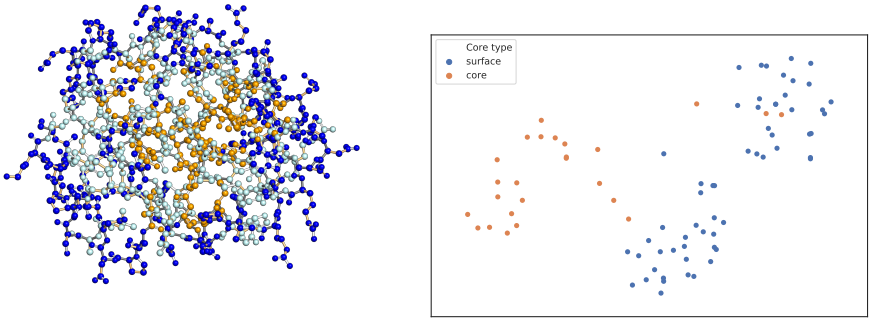


Figure 5: Left: 3-dimensional representation of CcmG reducing oxidoreductase (PDB ID 1KNG; Edeling et al., 2002), a protein from the test set. Atoms are colored dark blue (buried), orange (exposed), or neither (not colored). Right: t-SNE (Maaten & Hinton, 2008) projection of EBM hidden representation when focused on the alpha carbon atom for each residue in the hidden representation. In the embedding space, buried and surface residues are distinguished.

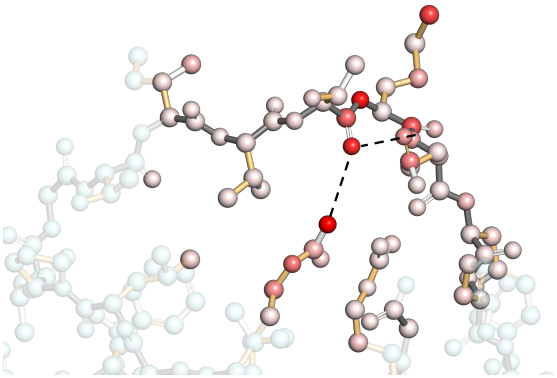

Figure 6: The model's saliency map applied to the test protein, serine proteinase inhibitor (PDB ID: 2CI2; McPhalen & James (1987)). The 64 atom context is centered on the carbonyl oxygen of residue 39 (isoleucine). Atoms in the context are labeled red with color saturation proportional to gradient magnitude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

## 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 physical features designed with expert knowledge and data analysis. Our work draws on their development of rigorous benchmarks for energy functions, but in contrast automatically learns complex features from data.

Neural networks have also been explored for protein folding. 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) investigated predicting 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. Mansimov et al. (2019) used a graph neural network to learn an energy function for small molecules. 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 can transfer knowledge to downstream supervised tasks (Bepler & Berger, 2019; Alley et al., 2019; Yang et al., 2019; Rives et al., 2019). These methods have also been explored for protein design (Wang et al., 2018).

Generative models of protein structures have also been proposed for generating protein backbones (Anand & Huang, 2018) and for the inverse protein folding problem (Ingraham et al., 2019).

## 6 Discussion

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

Generative biology (Rives et al., 2019) proposes that the design principles used by nature can be automatically learned from data and can be harnessed to generate and design new biological molecules and systems. High-fidelity generative modeling for proteins, operating at the level of structures and sequences, can enable generative protein design. 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 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), is a direction for future work. The methods explored here have the potential for extension into these settings.

### Acknowledgments

We thank Kyunghyun Cho, Siddharth Goyal, and Yann LeCun for helpful conversations. Alexander Rives was supported by NSF Grant #1339362.

## REFERENCES


Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation*, 13(6):3031–3048, 2017.

Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. Unified rational protein engineering with sequence-only deep representation learning. *bioRxiv*, 2019. doi: 10.1101/589333. URL https://www.biorxiv.org/content/early/2019/03/26/589333.

Mohammed AlQuraishi. End-to-end differentiable learning of protein structure. *Cell Systems*, 8(4):292 – 301.e3, 2019. ISSN 2405-4712. doi: https://doi.org/10.1016/j.cels.2019.03.006. URL http://www.sciencedirect.com/science/article/pii/S2405471219300766.

Xavier I Ambroggio and Brian Kuhlman. Computational design of a single amino acid sequence that can switch between two distinct protein folds. *Journal of the American Chemical Society*, 128(4):1154–1161, 2006.

Namrata Anand and Possu Huang. Generative modeling for protein structures. In *ICLR*, 2018.

Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. *arXiv e-prints*, art. arXiv:1607.06450, Jul 2016.

Tristan Bepler and Bonnie Berger. Learning protein sequence embeddings using information from structure. In *International Conference on Learning Representations*, 2019.

F Edward Boas and Pehr B Harbury. Potential energy functions for protein design. *Current opinion in structural biology*, 17(2):199–204, 2007.

Michael J Bower, Fred E Cohen, and Roland L Dunbrack Jr. Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: a new homology modeling tool. *Journal of molecular biology*, 267 (5):1268–1282, 1997.

Scott E Boyken, Zibo Chen, Benjamin Groves, Robert A Langan, Gustav Oberdorfer, Alex Ford, Jason M Gilmore, Chunfu Xu, Frank DiMaio, Jose Henrique Pereira, et al. De novo design of protein homo-oligomers with modular hydrogen-bond network–mediated specificity. *Science*, 352(6286):680–687, 2016.

Miguel A Carreira-Perpinan and Geoffrey E Hinton. On contrastive divergence learning. In *Aistats*, volume 10, pp. 33–40. Citeseer, 2005.

Wendy D Cornell, Piotr Cieplak, Christopher I Bayly, Ian R Gould, Kenneth M Merz, David M Ferguson, David C Spellmeyer, Thomas Fox, James W Caldwell, and Peter A Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society*, 117(19):5179–5197, 1995.

R. Das. Four small puzzles that Rosetta doesn't solve. *PLoS ONE*, 6(5):e20044, 2011.

Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

Ken A Dill. Dominant forces in protein folding. *Biochemistry*, 29(31):7133–7155, 1990.

Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv 1903.08689*, 2019.

Melissa A Edeling, Luke W Guddat, Renata A Fabianek, Linda Thöny-Meyer, and Jennifer L Martin. Structure of ccmg/dsbe at 1.14 å resolution: high-fidelity reducing activity in an indiscriminately oxidizing environment. *Structure*, 10(7):973–979, 2002.

Evan N. Feinberg, Debnil Sur, Zhenqin Wu, Brooke E. Husic, Huanghao Mai, Yang Li, Saisai Sun, Jianyi Yang, Bharath Ramsundar, and Vijay S. Pande. Potentialnet for molecular property prediction. *ACS Central Science*, 4(11):1520–1530, Nov 2018. ISSN 2374-7943. doi: 10.1021/acscentsci.8b00507. URL https://doi.org/10.1021/acscentsci.8b00507.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/gilmer17a.html.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Hong Guo and Dennis R Salahub. Cooperative hydrogen bonding and enzyme catalysis. *Angewandte Chemie International Edition*, 37(21):2985–2990, 1998.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Lisa Holm and Chris Sander. Fast and simple monte carlo algorithm for side chain optimization in proteins: application to model building by homology. *Proteins: Structure, Function, and Bioinformatics*, 14(2): 213–223, 1992.

Po-Ssu Huang, Scott E Boyken, and David Baker. The coming of age of de novo protein design. *Nature*, 537 (7620):320, 2016.

John Ingraham, Adam Riesselman, Chris Sander, and Debora Marks. Learning protein structure with a differentiable simulator. 2018.

John Ingraham, Vikas K Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. 2019.

Matthew P Jacobson, George A Kaminski, Richard A Friesner, and Chaya S Rapp. Force field validation using protein side chain prediction. *The Journal of Physical Chemistry B*, 106(44):11673–11680, 2002.

Joel Janin, Shoshanna Wodak, Michael Levitt, and Bernard Maigret. Conformation of amino acid side-chains in proteins. *Journal of molecular biology*, 125(3):357–386, 1978.

Lin Jiang, Eric A Althoff, Fernando R Clemente, Lindsey Doyle, Daniela Röthlisberger, Alexandre Zanghellini, Jasmine L Gallaher, Jamie L Betker, Fujie Tanaka, Carlos F Barbas, et al. De novo computational design of retro-aldol enzymes. *science*, 319(5868):1387–1391, 2008.

William L Jorgensen, David S Maxwell, and Julian Tirado-Rives. Development and testing of the opls all-atom force field on conformational energetics and properties of organic liquids. *Journal of the American Chemical Society*, 118(45):11225–11236, 1996.

Neil P King, Jacob B Bale, William Sheffler, Dan E McNamara, Shane Gonen, Tamir Gonen, Todd O Yeates, and David Baker. Accurate design of co-assembling multi-component protein nanomaterials. *Nature*, 510 (7503):103, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Brian Kuhlman, Gautam Dantas, Gregory C Ireton, Gabriele Varani, Barry L Stoddard, and David Baker. Design of a novel globular protein fold with atomic-level accuracy. *science*, 302(5649):1364–1368, 2003.

Themis Lazaridis and Martin Karplus. Effective energy functions for protein structure prediction. *Current opinion in structural biology*, 10(2):139–145, 2000.

Andrew Leaver-Fay, Matthew J O'Meara, Mike Tyka, Ron Jacak, Yifan Song, Elizabeth H Kellogg, James Thompson, Ian W Davis, Roland A Pache, Sergey Lyskov, et al. Scientific benchmarks for guiding macromolecular energy function improvement. In *Methods in enzymology*, volume 523, pp. 109–143. Elsevier, 2013.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Dennis R Livesay, Dang H Huynh, Sargis Dallakyan, and Donald J Jacobs. Hydrogen bond networks determine emergent mechanical and thermodynamic properties across a protein family. *Chemistry Central Journal*, 2(1):17, 2008.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Alex D MacKerell Jr, Donald Bashford, MLDR Bellott, Roland Leslie Dunbrack Jr, Jeffrey D Evanseck, Martin J Field, Stefan Fischer, Jiali Gao, H Guo, Sookhee Ha, et al. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *The journal of physical chemistry B*, 102(18): 3586–3616, 1998.

Jack B Maguire, Scott E Boyken, David Baker, and Brian Kuhlman. Rapid sampling of hydrogen bond networks for computational protein design. *Journal of chemical theory and computation*, 14(5):2751–2760, 2018.

Elman Mansimov, Omar Mahmood, Seokho Kang, and Kyunghyun Cho. Molecular geometry prediction using a deep generative graph neural network. *arXiv preprint arXiv:1904.00314*, 2019.

CA McPhalen and MNG James. Crystal and molecular structure of the serine proteinase inhibitor ci-2 from barley seeds. *Biochemistry*, 26(1):261–269, 1987.

Carl Pabo. Molecular technology: designing proteins and peptides. *Nature*, 301(5897):200, 1983.

Jasmina S Redzic and Bruce E Bowler. Role of hydrogen bond networks and dynamics in positive and negative cooperative stabilization of a protein. *Biochemistry*, 44(8):2900–2908, 2005.

Jane S Richardson and David C Richardson. Principles and patterns of protein conformation. In *Prediction of protein structure and the principles of protein conformation*, pp. 1–98. Springer, 1989.

Alexander Rives, Siddharth Goyal, Joshua Meier, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, 2019. doi: 10.1101/622803. URL https://www.biorxiv.org/content/early/2019/05/29/622803.

Andrew Senior, John Jumper, and Demis Hassabis. AlphaFold: Using AI for scientific discovery, 12 2018. URL https://deepmind.com/blog/alphafold/.

Maxim V Shapovalov and Roland L Dunbrack Jr. A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. *Structure*, 19(6):844–858, 2011.

Manfred J Sippl. Calculation of conformational ensembles from potentials of mena force: an approach to the knowledge-based prediction of local structures in globular proteins. *Journal of molecular biology*, 213(4): 859–883, 1990.

Seiji Tanaka and Harold A Scheraga. Medium-and long-range interaction parameters between amino acids for predicting three-dimensional structures of proteins. *Macromolecules*, 9(6):945–950, 1976.

P Tuffery, C Etchebest, Serge Hazout, and R Lavery. A new approach to the rapid determination of protein side chain conformations. *Journal of Biomolecular structure and dynamics*, 8(6):1267–1289, 1991.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

G. Wang and Jr. R. L. Dunbrack. Pisces: a protein sequence culling server. *Bioinformatics*, 19:1589–1591, 2003.

Jingxue Wang, Huali Cao, John Z. H. Zhang, and Yifei Qi. Computational protein design with deep learning neural networks. *Scientific Reports*, 8(1):6349, 2018. ISSN 2045-2322. doi: 10.1038/s41598-018-24760-x. URL https://doi.org/10.1038/s41598-018-24760-x.

Jinbo Xu. Distance-based protein folding powered by deep learning. *arXiv preprint arXiv:1811.03481*, 2018.

Kevin K. Yang, Zachary Wu, and Frances H. Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16(8):687–694, 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0496-6. URL https://doi.org/10.1038/s41592-019-0496-6.

## A.1 APPENDIX

### A.1.1 PSEUDOCODE FOR THE TRAINING ALGORITHM

Pseudocode for the training algorithm is given below in Algorithm 1.

**Algorithm 1** Training Procedure for the EBM

**Input:** Rotamer library $q(x|c)$, Training set of proteins $D$
**for** Protein $d_i$ of $D$ **do**
  ▷ *Sample random amino acid from* $d_i$
  $R \sim d_i$
  ▷ *Set positive sample to 64 nearest neighbor atoms of carbon beta of R*
  $c^+ \leftarrow \text{NN}_{64}(R)$
  ▷ *Generate N negative samples from the rotamer library*
  $c^- \leftarrow q(x|c^+)$
  ▷ *Compute loss of model (logsumexp across all negative samples)*
  $L_{ml} = E(c^+;\theta) + \text{logsumexp}(-E(c^+;\theta), -E(c_0^-;\theta), -E(c_1^-;\theta), \ldots, -E(c_N^-;\theta))$
  ▷ *Minimization step of* $L_{ml}$ *using Adam optimizer*
  $\theta \leftarrow \theta - \nabla_\theta L_{ml}$
**end for**

### A.1.2 MODEL ARCHITECTURE

Architectural descriptions are provided below for each the three neural network baselines as well as for the Atom Transformer. For Set2set models, we use 6 processing steps of computation. For graph networks, we add residual connections between each layer.

| |
|---|
| Embed Each Atom to 256 Dim |
| Flatten |
| Dense → 1024 |
| 1024 → 1024 |
| 1024 → 1024 |
| ResBlock down 256 |
| Global Mean Pooling |
| Dense → 1 |

(a) Fully Connected Model

| |
|---|
| Embed Each Atom to 256 Dim |
| Dense → 1024 |
| Repeat (6x): |
| LSTM 2048<br>Attention 2048 → 128 → 1 |
| End Repeat |
| Dense → 1024 |
| 1024 → 1 |

(b) Set2Set Model (6 Permutation Invariant Blocks)

Figure A1: Architectures for Fully Connected and Set2Set Baselines

| |
|---|
| Embed Each Atom to 512 Dim |
| Graph Attention Layer |
| Graph Attention Layer |
| Graph Attention Layer |
| Graph Attention Layer |
| Graph Attention Layer |
| Graph Attention Layer |
| Graph Attention Layer |
| Graph Attention Layer |
| Graph Attention Layer |
| Global Average Pooling |
| dense $\rightarrow$ 1 |

Figure A2: Graph Network Model Architecture (9 Graph Attention Blocks)

| |
|---|
| Embed Each Atom to 256 Dim |
| Transformer Encoder Block (8 heads, feedforward dim 1024, 256 encoder dim) |
| Transformer Encoder Block (8 heads, feedforward dim 1024, 256 encoder dim) |
| Transformer Encoder Block (8 heads, feedforward dim 1024, 256 encoder dim) |
| Transformer Encoder Block (8 heads, feedforward dim 1024, 256 encoder dim) |
| Transformer Encoder Block (8 heads, feedforward dim 1024, 256 encoder dim) |
| Transformer Encoder Block (8 heads, feedforward dim 1024, 256 encoder dim) |
| Global Max Pooling |
| dense $\rightarrow$ 1 |

Figure A3: Atom Transformer Model (6 Transformer Encoder Blocks)