EquiFold: Protein Structure Prediction with a Novel Coarse-Grained Structure Representation

Jae Hyeon Lee[1], **Payman Yadollahpour**[2], **Andrew Watkins**[1], **Nathan C. Frey**[1], **Andrew Leaver-Fay**[1], **Stephen Ra**[1], **Kyunghyun Cho**[1,3,4], **Vladimir Gligorijevic**[1], **Aviv Regev**[2], and **Richard Bonneau**[1]

[1]Prescient Design, Genentech
[2]Genentech Research and Early Development, Genentech
[3]Department of Computer Science, Courant Institute of Mathematical Sciences, New York University
[4]Center for Data Science, New York University
lee.jae-hyeon@gene.com

Abstract

Designing proteins to achieve specific functions often requires *in silico* modeling of their properties at high throughput scale and can significantly benefit from fast and accurate protein structure prediction. We introduce EquiFold, a new end-to-end differentiable, SE(3)-equivariant, all-atom protein structure prediction model. EquiFold uses a novel coarse-grained representation of protein structures that does not require multiple sequence alignments or protein language model embeddings, inputs that are commonly used in other state-of-the-art structure prediction models. Our method relies on geometrical structure representation and is substantially smaller than prior state-of-the-art models. In preliminary studies, EquiFold achieved comparable accuracy to AlphaFold but was orders of magnitude faster. The combination of high speed and accuracy make EquiFold suitable for a number of downstream tasks, including protein property prediction and design.

1 Introduction

Recent studies using deep neural networks to predict protein structure [1, 2, 3, 4, 5] have accelerated the development of structure-based methods for protein property prediction and design. However, some models tend to predict one or few conformations that may not be optimal for properties of interest, such as ligand-binding pockets [6] and protein-protein interfaces [7, 8, 9]. Moreover, these models require either multiple sequence alignments (MSA), protein language model embeddings, or other statistical information distilled from large sequence databases. While complex inputs such as embeddings and MSAs have proven useful for structure prediction, they increase the complexity of properly testing these methods, require significant time to derive, and often scale poorly with respect to sequence library size, and thus may overall limit downstream use. Conversely, the physics of protein dynamics and interactions do not depend directly on these inputs.

Here, we introduce EquiFold, a novel representation of protein structures and an end-to-end differentiable, SE(3)-equivariant neural network that predicts a protein structure given its primary sequence via iterative refinement. EquiFold makes atomically accurate predictions for *de novo* designed mini-proteins with *in silico* predicted structures [10] and on experimental antibody structures from the Protein Data Bank [11]. We focus on this set because small structures (e.g., designed mini-proteins) and flexible loops (e.g., the CDR-H3 loop of IgG based antibodies) have proven to be as difficult to predict as larger structures in recent tests of state-of-the-art methods [12, 13, 14]. The model relies solely on geometrical structure representation and can readily incorporate various energy functions as

Preprint. Under review.

physical priors, which we hypothesize to be instrumental in exploring conformational landscapes towards predicting properties of interest.

2 Related Work

AlphaFold [1] and similar models [2, 3, 4] have been successful at protein structure prediction, by representing the geometry of the chain of backbone atoms as a set of nodes paired with Euclidean transformations and using an iterative refinement procedure that updates the transformations per each block of the structure module. In these models, side-chain geometries are implicitly modeled until they are predicted as a series of torsion angles by the module's final block. However, such implicit modeling of side-chains may make it difficult to model their atoms' placements and interactions in 3D space. For example, to avoid steric clashes, such models must learn from data complex distributions in the high dimensional space of torsion angles that span multiple residues. In contrast, models that represent side-chain degrees of freedom explicitly in 3D would likely need to learn substantially simpler distributions to achieve the same goal.

Coarse-grained structure representations of proteins [15, 16, 17] typically model each residue by one or few nodes with associated positions determined by its atoms' coordinates. These strategies can increase computational efficiency for predictive tasks, such as interacting residue predictions [18] and functional residue prediction [19], and are appropriate for certain generative modeling tasks, such as generating backbone scaffolds [20, 21]. However, these approaches have previously sacrificed all-atom structure resolution needed for design and packing-related tasks, along with much information useful for predicting protein functions.

To address these limitations, we develop a novel coarse-grained representation that retains all-atom structure resolution. In this representation, side-chain degrees of freedom are modeled explicitly in 3D space rather than intrinsically through torsion angles, which we conjecture makes it easier to model geometry and interactions in 3D space.

SE(3)-equivariant neural networks have been used in various 3D object modeling tasks, including atomic potential prediction [22, 23, 24], molecular property prediction [25, 26, 27, 28, 29], protein structure prediction, [2] and docking [30, 31]. They incorporate the symmetry of 3D space and are substantially more data efficient than their non-symmetry-aware counterparts [32, 33]. Input to SE(3)-equivariant networks consists of geometric tensors, or the irreducible representations of the SO(3) group (the group of rotations in $\mathbb{R}^3$), of various degrees l, such as scalars ($l = 0$), vectors ($l = 1$), and higher degree ($l \geq 2$) tensors that transform under a rotation R through multiplication with corresponding Wigner D-matrices $D_l(R)$. Objects with an associated rotation, such as backbone frames [1], can be initially embedded with geometric tensors [15]. We use an SE(3)-equivariant model adapted from Equiformer [29] along with an initial embedding of coarse-grained nodes with geometric tensors.

3 Methods

3.1 A coarse-grained representation

Given a protein sequence $a = (a_1, ..., a_N)$ of length N where a_i is one of the 20 canonical amino acids, its structure is specified by the set of 3D coordinates of its atoms grouped by amino acid, $X = \{X_i^{a_i} | X_i^{a_i} \in \mathbb{R}^{n_{a_i} \times 3}\}_{i=1}^N$ where n_{a_i} is the number of atoms in residue a_i. In a CG representation, each amino acid a_i is represented by a predetermined set of CG *nodes* $\{c_j^{a_i}\}_{j=1}^{n_{CG}^{a_i}}$, where each CG node $c_j^{a_i}$ represents a subset $\sigma_j^{a_i}$ of the amino acid's constituent atoms. The CG nodes are chosen such that 1) their union represents all atoms of the amino acid; 2) each member atom of a node shares at least one covalent bond with another member of the same node; and 3) each node consists of at least three atoms that collectively form a rigid body whose orientation in 3D is uniquely determined. Based on the last property, we define a *forward* CG mapping $\mathcal{F}$ of the 3D coordinates $X_i^{a_i}$ of an amino acid a_i's atoms into its CG representation:

$$\mathcal{F} : X_i^{a_i} \mapsto \{(T_j^{a_i}, c_j^{a_i})\}_{j=1}^{n_{CG}^{a_i}},$$

where each tuple in the set consists of a CG node identity $c_j^{a_i}$ and a corresponding Euclidean transformation $T_j^{a_i} = (\vec{t}_j^{a_i}, R_j^{a_i}) \in SE(3)$ that maps the predefined template coordinates for atoms

in $\sigma_j^{a_i}$ to the corresponding input atom coordinates (see Appendix A.1). Table 3 illustrates an example of a CG scheme used in this work that respects the above properties while minimizing the redundancy of atom representations across CG nodes. Figure 1 gives illustrations of this CG scheme for select amino acids.

We define a *reverse* CG mapping $\mathcal{G}$ for an amino acid a_i that maps its CG representation to the 3D atom coordinates $X_i^{a_i}$ as follows: for each atom in the amino acid, we average the 3D coordinates associated with the atom specified by any of its CG nodes.

3.2 Geometric tensor features and initial embedding

To each CG node, we assign a set of geometric tensor features of degree $l = 0, ..., l_{max}$ with n_c channels per degree [25, 28, 29]. As there are $2l + 1$ features associated with an l-degree tensor, there are $n_c \times (l_{max} + 1)^2$ features in total per node. We define the initial embedding of the nodes $\mathcal{E} : C \times SO(3) \to \mathbb{R}^{n_c \times (l_{max}+1)^2}$ where C is the pre-determined set of the CG node types:

$$\mathcal{E} : (c_j^{a_i}, R_j^{a_i}) \mapsto e_j^{a_i} = D(R_j^{a_i}) \cdot \textsc{LookUp}(c_j^{a_i})$$

where $\textsc{LookUp} : C \to \mathbb{R}^{n_c \times (l_{max}+1)^2}$ is a typical embedding function and $D(R_j^{a_i})$ is the direct sum of the Wigner D-matrices D_l corresponding to the tensor features of various degrees l:

$$D(R_j^{a_i}) = \bigoplus_{l=0}^{l_{max}} \left[\bigoplus_{c=1}^{n_c} D_l(R_j^{a_i}) \right].$$

3.3 Structure prediction via iterative refinement using an SE(3)-equivariant neural network

Given an input sequence, its CG representation is instantiated with Euclidean transformations whose translations and rotations are sampled from a normal distribution with zero mean and unit variance and a uniform distribution over the SO(3) group, respectively. The output structure is predicted via iterative refinement using an SE(3)-equivariant model that consists of N_{blocks} blocks sharing the same architecture, where each block is composed of N_{sub} equivariant sub-blocks. The block architecture is adapted from Equiformer [29], as detailed in Appendix A.2.

Each block takes as input either the initial CG representation or the output of the previous block. A block outputs two $l = 1$ tensors for each node, one of which is used as the vector part of a non-unit quaternion to compute an update R' to the node's rotation R_{in} and the other to compute an update $\vec{t'}$ to its translation $\vec{t}_{in}$ [1] (we drop amino acid and CG node indices for clarity). The input Euclidean transformation $T_{in} = (\vec{t}_{in}, R_{in})$ is then updated to $T_{new} = (\vec{t}_{new}, R_{new})$ via the following update rules:

$$R_{new} = R' R_{in} \text{ and } \vec{t}_{new} = \vec{t'} + \vec{t}_{in}.$$

A block can optionally transform or simply copy the input embedding of each node, but either way, it is multiplied by the direct sum of the Wigner D-matrices corresponding to the update rotation R'. The proof of equivariance of these update rules is provided in Appendix A.3.

3.4 Loss functions

To train the model, the Frame Aligned Point Error (FAPE) loss and the structure violation loss introduced in [1] are computed based on the CG node Euclidean transformations and the reverse CG mapped structures output from each block, respectively. Unlike in [1], all atom FAPE loss is computed for every block. We observed that the structure violation loss is important, since EquiFold is more susceptible to predicting non-physical bond lengths and angles and non-bonded atom distances, due to its modeling of CG nodes in extrinsic 3D space, compared to other models that predict torsion angles representing internal degrees of freedom [1, 2, 3]. However, we do not observe strong instabilities in EquiFold training dynamics like those reported in [1], when the loss is used from the beginning of training. More details on the losses, including their relative weights, other hyper-parameter choices, and training strategies are found in Appendix A.4.

Table 1: Performance of EquiFold on *de novo* designed mini-proteins from the test set broken down by fold. All RMSDs are reported in angstroms. "RMSD" and "C_α RMSD" report all atom and backbone C_α atom RMSDs based on the comparison of EquiFold predictions and Rosetta predicted structures, respectively. The last column "C_α RMSD (train)" reports the averaged backbone C_α atom RMSD based on the comparison of each test example to all examples of the same fold and length in the training set, using Rosetta predicted structures.

Fold	Train	Test	RMSD	C_α RMSD	C_α RMSD (train)
$\alpha\beta\beta\alpha$	92	3	2.20	1.76	4.36
$\beta\beta\alpha\beta\beta$	533	8	1.12	0.45	0.91
$\beta\alpha\beta\beta$	787	15	1.00	0.43	0.99
$\alpha\alpha\alpha$	1330	24	1.14	0.53	2.79

Table 2: Performance of EquiFold on antibody structures over the test set used in [13] broken down by framework and variable regions. Performance of other models are as reported in [13]. RMSD (in Å) over N, C_α, and C backbone atoms are reported for valid comparison. Column "Time" reports approximate inference time for EquiFold and approximate run times for other models as reported in [13] for predicting all atom structures.

Model	H Fr	H1	H2	H3	L Fr	L1	L2	L3	Time
EquiFold	**0.41**	**0.74**	0.70	**2.92**	**0.39**	**0.78**	**0.34**	**1.02**	$\sim$ 1 second
AlphaFold-Multimer	0.43	0.75	**0.69**	3.02	**0.39**	0.82	0.41	1.13	$\sim$ 1 hour
IgFold	0.45	0.80	0.75	2.99	0.45	0.83	0.51	1.07	$\sim$ 1 minute

4 Results

We report EquiFold's performance on two structure datasets we curated to focus on structures that present key challenges to protein structure prediction methods: designed mini-proteins and antibody loops. In each challenge, we are primarily predicting structures that are small, have high error in recent blind benchmarks, and have significant potential applications in biotechnology and protein design. These sets also are comprised of regions that elude homology detection and traditional concepts that underpin MSA (as do, by construction, *de novo* designed sequences). EquiFold achieved high accuracy and speed over these sets, demonstrating that it can enable new downstream protein design and engineering.

4.1 *De novo* designed mini-proteins

We first tested EquiFold on a set of *de novo* designed mini-proteins from [10], each having one of four different folds ($\alpha\alpha\alpha, \alpha\beta\beta\alpha, \beta\alpha\beta\beta$, and $\beta\beta\alpha\beta\beta$), with associated *in silico* structures predicted using Rosetta [34]. After filtering for sequences with stability score greater than 1, we retained 2,842 sequences whose lengths range from 43 to 50. We randomly split the sequences into train, validation, and test sets of size 2,742, 50, and 50. Test sequences have nearest training sequence similarity ranging from 44% to 80% with mean of 67.2%.[1] Table 1 shows all atom and C_α RMSD based on the test set broken down by fold. Average inference speed over the entire test set was 0.03 seconds per sequence, with the model containing 2.3M trainable parameters (see Table 4). Figure 2 shows test example predictions overlaid with ground truth structures. Rather than serving as a benchmark relative to other structure prediction models, since the ground truth here are *in silico* predicted structures, this result illustrates the ability of EquiFold to learn a variety of distinct protein topologies with atomic resolution all-atom accuracy.

4.2 Antibodies

We obtained all antibody (Ab) experimental structures from the PDB [11] that were listed in The Structural Antibody Database (SAbDab) [35], as accessed on January 12, 2022. We processed this

[1] Sequence similarity is defined as $(l_{query} - n_{edit})/l_{query}$ where l_{query} is the length of the query sequence and n_{edit} the Levenshtein edit distance between query and target sequences.

dataset to obtain the variable fragment portions of the structures and annotated the sequences with Chothia numbering using ANARCI [36]. We obtained 6,789 structures with a resolution better than 4Å and deposited before July 1, 2021 as training set, of which 50 structures were used for validation. We used the same test set structures as in [13] that have resolution better than 3 Å and deposited after the aforementioned date. Compared to other models, EquiFold achieves similar or better accuracy in backbone atom RMSD across different sequence regions (see Table 2) and results in all atom RMSD of 1.52 Å averaged over the test set. Importantly, the model has fast inference speeds of approximately 1 second per Ab on average on a single A100 GPU for predicting all atom structures and contains 7.38M trainable parameters (see Table 4), compared to 559.6M of IgFold (including 558M of the antibody language model AntiBERTy) [13] and 93.2M of AlphaFold [1] that requires time-consuming input preparation steps. Given its high speed and accuracy, it is practical to predict structures at high-throughput scale for millions of antibodies observed in deep sequencing data sets [37] and integrate the model in a design workflow [16, 21]. Fig 3 shows selected test set predictions overlaid with ground truth structures.

5 Conclusion and Future Work

We introduced EquiFold, a new end-to-end differentiable protein structure prediction model that uses a novel coarse-grained (CG) representation of proteins. The model achieves high test set accuracy on two datasets, while running at a substantially faster speed. Notably, it is trained on significantly less data and does not rely on multiple sequence alignments [1, 2] or protein language model embeddings [3, 4, 13]. Its accuracy and speed make it practical to integrate EquiFold as a sub-component of a larger model. For instance, EquiFold can be combined in an end-to-end differentiable fashion with another neural network that predicts various molecular properties based on the structure output by its last block.

We leave to future work training EquiFold on more general classes of proteins from the PDB [11] and examining its generalizability to novel folds unseen in training data. Scaling to larger proteins will require addressing the quadratic complexity of the message passing layers, possibly using similar strategies in earlier works [1]. To integrate physical priors, we will extend the CG representation to include hydrogens and implement various energy functions such as the Rosetta all-atom energy function [34]. With such utilities, EquiFold can be adapted to generate conformational ensembles within an energy band, perform flexible docking, and fit structural models to experimental data such as that from cryogenic electron microscopy (cryo-EM) and hydrogen-deuterium exchange (HDX). Lastly, the CG representation could be used in generative modeling of protein sequence and structure together, rather than modeling them sequentially as done in recent works [16, 20, 21].

References

[1] J. Jumper et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 2021.

[2] M. Baek et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 2021.

[3] R. Wu et al. High-resolution de novo structure prediction from primary sequence. *bioRxiv*, 2022.

[4] Z. Lin et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

[5] R. Chowdhury et al. Single-sequence protein structure prediction using language models from deep learning. *bioRxiv*, 2021.

[6] A. Meller et al. Predicting the locations of cryptic pockets from single protein structures using the PocketMiner graph neural network. *bioRxiv*, 2022.

[7] P. Gainza et al. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nat Methods* 2020.

[8] M. L. Fernández-Quintero et al. Paratope states in solution improve structure prediction and docking. *Structure,* 2022.

[9] B. Jing, S. Eismann, P. N. Soni, and R. O. Dror. Equivariant graph neural networks for 3d macromolecular structure. *arXiv*, 2022.

[10] G. J. Rocklin et al. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science,* 2017.

[11] H. Berman, K. Henrick, and H. Nakamura. Announcing the worldwide Protein Data Bank. *Nat Struct Mol Biol*, 2003.

[12] J. A. Ruffolo, J. Sulam, and J. J. Gray. Antibody structure prediction using interpretable deep learning. *Patterns,* 2022.

[13] J. A. Ruffolo, L.-S. Chu, S. P. Mahajan, and J. J. Gray. Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. *bioRxiv*, 2022.

[14] R. Das. Four small puzzles that rosetta doesn't solve. *PLOS ONE*, 2011.

[15] C. Hsu et al. Learning inverse folding from millions of predicted structures. *bioRxiv*, 2022.

[16] J. Dauparas et al. Robust deep learning based protein sequence design using ProteinMPNN. *Science*, 2022.

[17] K. T. Simons et al. Ab initio protein structure prediction of CASP III targets using ROSETTA. *Protein*, 1999.

[18] A. Del Vecchio et al. Neural message passing for joint paratope-epitope prediction. arXiv, 2021.

[19] V. Gligorijević et al. Structure-based protein function prediction using graph convolutional networks. *Nat Commun*. 2021.

[20] N. Anand and T. Achim. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv*, 2022.

[21] J. Wang et al. Scaffolding protein functional sites using deep learning. *Science,* 2022.

[22] S. Batzner et al. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat Commun*, 2022.

[23] I. Batatia et al. The Design Space of E(3)-Equivariant Atom-Centered Interatomic Potentials. *arXiv*, 2022.

[24] I. Batatia et al. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. *arXiv*, 2022.

[25] N. Thomas et al. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds. *arXiv*, 2022.

[26] F. B. Fuchs et al. SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks. *arXiv*, 2020.

[27] V. G. Satorras, E. Hoogeboom, and M. Welling. E(n) equivariant graph neural networks. *arXiv*, 2022.

[28] J. Brandstetter et al. Geometric and physical quantities improve E(3) equivariant message passing. *arXiv*, 2022.

[29] Y.-L. Liao and T. Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. *arXiv*, 2022.

[30] O.-E. Ganea et al. Independent SE(3)-equivariant models for end-to-end rigid protein docking. In *International Conference on Learning Representations*, 2022.

[31] W. Jin, D. Barzilay, and T. Jaakkola. Antibody-antigen docking and design via hierarchical structure refinement. *In Proceedings of the 39th International Conference on Machine Learning*, 2022.

[32] N. Frey et al. Neural Scaling of Deep Chemical Models. *chemRxiv*, 2022.

[33] A. Musaelian et al. Learning local equivariant representations for large-scale atomistic dynamics. *arXiv*, 2022.

[34] J. K. Leman, et al. Macromolecular modeling and design in Rosetta: recent methods and frameworks. *Nat Methods*, 2020.

[35] C. Schneider, M. Raybould, and C. M. Deane, SAbDab in the age of biotherapeutics: updates including SAbDab-nano, the nanobody structure tracker, *Nucleic Acids Research*, 2022.

[36] J. Dunbar and C. M. Deane. ANARCI: antigen receptor numbering and receptor classification. *Bioinformatics*, 2016.

[37] T. H. Olsen, F. Boyles, and C. M. Deane. *Protein Science*, 2021.

[38] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica*, 1976.

[39] M. Geiger and T. Smidt. e3nn: Euclidean neural networks. *arXiv*, 2022.

[40] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv*, 2017.

[41] A. Paszke et al. PyTorch: An imperative style, high-performance deep learning library. *arXiv*, 2019.

A Appendix

A.1 Computing coarse-grained node template coordinates and ground-truth Euclidean transformations

For each coarse-grained (CG) node defined in Table 3, we compute the template coordinates of all the atoms comprising the node as follows. Given a protein structure dataset $\mathcal{D}$, we compute the Euclidean transformation T_q corresponding to each CG node q in $\mathcal{D}$ using Algorithm 21 of [1] with the 3D coordinates of the first three atoms of the CG node q's atom group as input. Next, we apply the inverse Euclidean transformation T_q^{-1} to the 3D coordinates of all the atoms in the node q into the corresponding local frame. Lastly, we average the observed coordinates in the local frames across all instances in the dataset $\mathcal{D}$ grouped by CG node type defined in Table 3.

To compute the ground truth Euclidean transformation for a given CG node to be used in the FAPE loss [1], we use the Kabsch algorithm [38] to determine the transformation from the template coordinates to the observed coordinates that minimizes the root mean square error (RMSE) of the node's constituent atoms. To elaborate, for the j-th CG node $c_j^{a_i}$ containing M atoms of the i-th amino acid a_i, we apply the Kabsch algorithm to the template coordinates $W = [\vec{w}_1^{c_j^{a_i}}, \cdots, \vec{w}_M^{c_j^{a_i}}] \in \mathbb{R}^{3\times M}$ and the corresponding input coordinates $X = [\vec{x}_1^{c_j^{a_i}}, \cdots, \vec{x}_M^{c_j^{a_i}}] \in \mathbb{R}^{3\times M}$. The algorithm factorizes the covariance matrix $H = W_c X_c^T$ into eigenvectors and values using singular vector decomposition,

$$H = USV^T$$

where W_c and X_c are mean centered template and input coordinates. The resulting rotation and translation are

$$R_j^{a_i} = VU^T, \quad \vec{t}_j^{a_i} = \vec{x}_\mu^{c_j^{a_i}} - R_j^{a_i}\vec{w}_\mu^{c_j^{a_i}},$$

where $\vec{w}_\mu^{c_j^{a_i}} \in \mathbb{R}^3$ and $\vec{x}_\mu^{c_j^{a_i}} \in \mathbb{R}^3$ are the mean coordinates of W and X respectively.

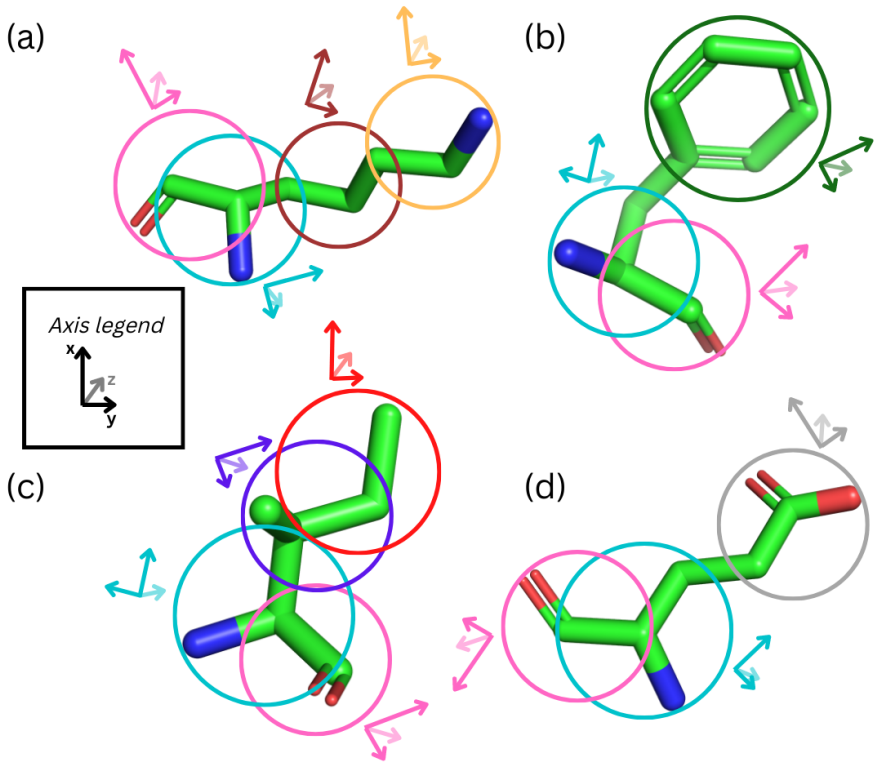

Figure 1: A representation of EquiFold's coarse-grained node scheme as applied to four amino acids: a) lysine, b) phenylalanine, c) isoleucine, and d) glutamate. Atoms belonging to each node are circled and axes representing the node's specification (by rotation-transformation) are shown in a corresponding color. Backbone nodes, which are common to each amino acid, are shown in consistent colors across panels.

Table 3: The amino acid coarse-graining (CG) scheme used in this work. Each column with "CG" prefix defines a group of atoms belonging to a CG node for each amino acid.

AA	CG1	CG2	CG3	CG4
ALA	C, C_α, C_β, N	C, C_α, O	-	-
ARG	C, C_α, C_β, N	C, C_α, O	$C_\beta, C_\gamma, C_\delta$	$N_\epsilon, N_{\eta1}, N_{\eta2}, C_\zeta$
ASN	C, C_α, C_β, N	C, C_α, O	$C_\gamma, N_{\delta2}, O_{\delta1}$	-
ASP	C, C_α, C_β, N	C, C_α, O	$C_\gamma, O_{\delta1}, O_{\delta2}$	-
CYS	C, C_α, C_β, N	C, C_α, O	$C_\alpha, C_\beta, S_\gamma$	-
GLN	C, C_α, C_β, N	C, C_α, O	$C_\gamma, C_\delta, O_{\epsilon1}, N_{\epsilon2}$	-
GLU	C, C_α, C_β, N	C, C_α, O	$C_\gamma, C_\delta, O_{\epsilon1}, O_{\epsilon2}$	-
GLY	C, C_α, N	C, C_α, O	-	-
HIS	C, C_α, C_β, N	C, C_α, O	$C_\gamma, C_{\delta2}, C_{\epsilon1}, N_{\delta1}, N_{\epsilon2}$	-
ILE	C, C_α, C_β, N	C, C_α, O	$C_\beta, C_{\gamma1}, C_{\gamma2}$	$C_\beta, C_{\gamma1}, C_{\delta1}$
LEU	C, C_α, C_β, N	C, C_α, O	$C_\gamma, C_{\delta1}, C_{\delta2}$	-
LYS	C, C_α, C_β, N	C, C_α, O	$C_\beta, C_\gamma, C_\delta$	$C_\delta, C_\epsilon, N_\zeta$
MET	C, C_α, C_β, N	C, C_α, O	$C_\gamma, C_\epsilon, S_\delta$	-
PHE	C, C_α, C_β, N	C, C_α, O	$C_\gamma, C_{\delta1}, C_{\delta2}, C_{\epsilon1}, C_{\epsilon2}, C_\zeta$	-
PRO	C, C_α, C_β, N	C, C_α, O	$C_\beta, C_\gamma, C_\delta$	-
SER	C, C_α, C_β, N	C, C_α, O	$C_\alpha, C_\beta, O_\gamma$	-
THR	C, C_α, C_β, N	C, C_α, O	$C_\beta, C_{\gamma2}, C_{\gamma1}$	-
TRP	C, C_α, C_β, N	C, C_α, O	$C_\gamma, C_{\delta1}, C_{\delta2}, C_{\epsilon2}, C_{\epsilon3}, C_{\zeta2}, C_{\zeta3}, C_{\eta2}, N_{\epsilon1}$	-
TYR	C, C_α, C_β, N	C, C_α, O	$C_\gamma, C_{\delta1}, C_{\delta2}, C_{\epsilon1}, C_{\epsilon2}, C_\zeta, O_\eta$	-
VAL	C, C_α, C_β, N	C, C_α, O	$C_\beta, C_{\gamma1}, C_{\gamma2}$	-

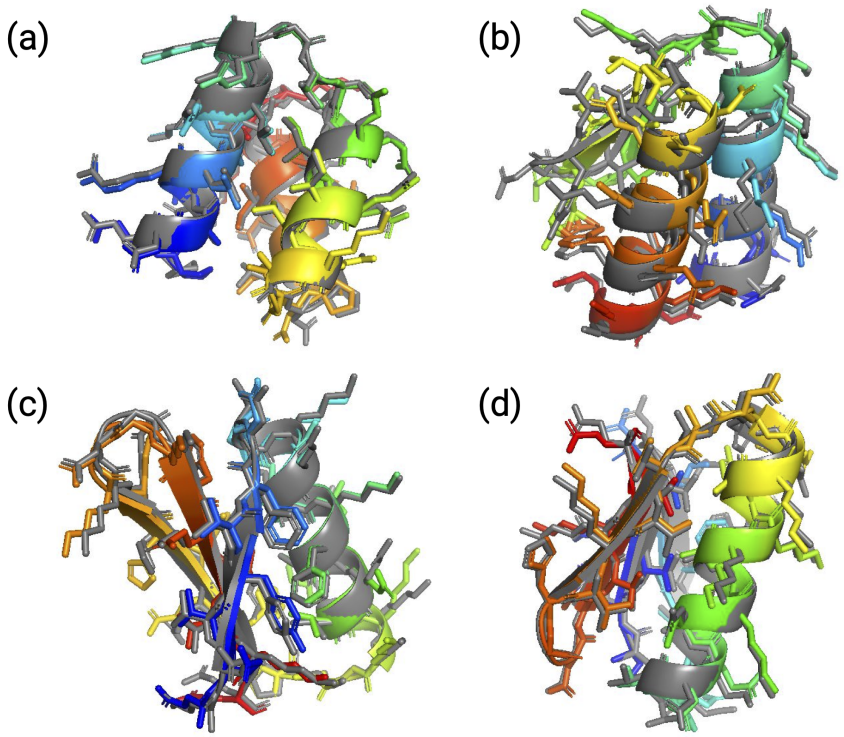

Figure 2: EquiFold's predictions (rainbow) on test set examples are shown overlaid with the ground truth structures (gray), one for each of the four folds [10]: a) HHH_rd3_0030 ($\alpha\alpha\alpha$, 0.88Å, 62%), b) HEEH_rd2_0948 ($\alpha\beta\beta\alpha$, 1.75Å, 51%), c) EHEE_rd3_0145 ($\beta\alpha\beta\beta$, 1.15Å, 60%), and d) EEHEE_rd3_0132 ($\beta\beta\alpha\beta\beta$, 1.12Å, 60%), where the last two numbers in parentheses correspond to all-atom RMSD and sequence similarity to the nearest training example. EquiFold achieves high accuracy on sequences that have low similarity to any training examples.

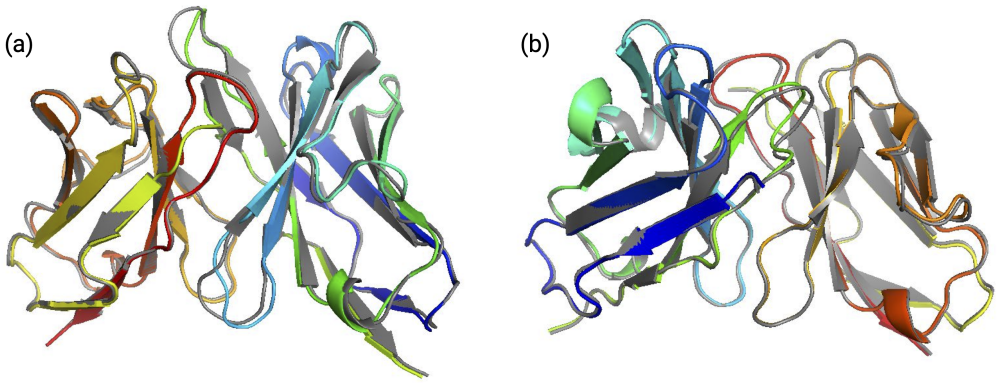

Figure 3: EquiFold's predictions (rainbow) on two examples from the test set [13] are shown overlaid with the ground truth structures (gray): a) 7lfb_HL (1.24Å, 71%) and 7oo2_CD (1.00Å, 72%), where the two numbers in parentheses correspond to all-atom RMSD and sequence similarity to the nearest training example.

A.2 SE(3)-equivariant neural network architecture details

Each block of the neural network consists of N_{sub} sub-blocks that share the same architecture, where each sub-block is an adapted version of the Equiformer's "Transformer block" [see 29, Figure 1]. As the general theory of SE(3)-equivariant neural networks and the implementation of the Transformer block are well-described in [29], here we describe only the modifications in our adapted version.

Given the input set of coarse-grained (CG) nodes and their Euclidean transformations, the block initially computes pairwise distances r_{ij} and normalized distance vectors $\hat{r}_{ij}$, where i and j index CG nodes. r_{ij} is projected onto d_{bessel} radial Bessel basis with learnable weights [22] and a cutoff distance r_c, which are used in radial functions that parameterize tensor products in the Equivariant Graph Attention module. Instead of a polynomial envelope function used in [22], we apply `e3nn.math.soft_unit_step` from [39] with $10(1 - r_{ij})/r_c$ as input. $\hat{r}_{ij}$ is used to compute spherical harmonics SH$(\hat{r}_{ij})$ input to tensor products. When training the network, gradients do not propagate through r_{ij} and $\hat{r}_{ij}$.

In our adapted version of the Equivariant Graph Attention module, after the application of the initial linear layers to input node embeddings, instead of element-wise summation, channel-wise fully connected tensor products are applied to the embeddings of every CG-node pair ij, which is followed by another linear layer to produce output tensors x_{ij} with the same number of channels as the input. Next, Depth-wise Tensor Product (DTP) is applied to x_{ij} and SH$(\hat{r}_{ij})$ with a radial function that takes as input the Bessel basis mentioned above and additionally a scalar *edge* embedding vector corresponding to the primary amino acid sequence distance for the CG-pair ij, clamped at maximum absolute distance of 32; for input proteins with multiple chains, sequence distances across chains are set at the maximum distance. The edge embedding is implemented via a simple look-up table with learnable weights and has the same dimension as the number of channels in input tensors. The output of the DTP layer is uniformly shuffled and grouped by N_{head} attention heads and a linear layer is applied to produce tensors of various degrees with appropriate channel numbers for the remainder of the module.

The output of each sub-block except for the last one are updated node embeddings corresponding to the input CG nodes. The last sub-block outputs only two $l = 1$ vectors per CG node as mentioned in Section 3.3. Edge embeddings are shared across sub-blocks of a given block.

A.3 Proof of equivariance of the Euclidean transformation update

Under a global Euclidean transformation $\tilde{T} = (\tilde{\vec{t}}, \tilde{R})$, the Euclidean transformation $T = (\vec{t}, R)$ corresponding to a CG node that specifies the mapping between the observed and template coordinates transforms as

$$(\vec{t}, R) \mapsto (\tilde{R}\vec{t} + \tilde{\vec{t}}, \tilde{R}R),$$

and the update transformation $T' = (\vec{t'}, R')$ output by a block transforms as

$$(\vec{t'}, R') \mapsto (\tilde{R}\vec{t'}, \tilde{R}R'\tilde{R}^T).$$

We show that the update rules given in Section 3.3 are equivariant by applying a global transformation $\tilde{T}$ to the input $T_{in} = (\vec{t}_{in}, R_{in})$ and update T' transformations and showing that their composition is equivalent to the global transformation of the output $T_{new} = (\vec{t}_{new}, R_{new})$:

$$\vec{t'} + \vec{t}_{in} \mapsto \tilde{R}\vec{t'} + (\tilde{R}\vec{t}_{in} + \tilde{\vec{t}}) = \tilde{R}(\vec{t'} + \vec{t}_{in}) + \tilde{\vec{t}} = \tilde{R}\vec{t}_{new} + \tilde{\vec{t}}$$

and

$$R'R_{in} \mapsto \tilde{R}R'\tilde{R}^T\tilde{R}R_{in} = \tilde{R}R'R_{in} = \tilde{R}R_{new}.$$

A.4 Hyper-parameters and training details

Table 4 provides the hyper-parameter values used for the two experiments described in Section 4. Both models were trained using the Adam optimizer [40] with PyTorch [41] default parameters, including `beta=(0.9, 0.999)` and the initial learning rate of 10^{-3}. We used a warm-up phase of 10,000 steps, where the learning rate was linearly increased from 0 to the final value and input structures to the model were linearly interpolated between the ground truth structures and random initializations; for rotations, we used quaternion spherical linear interpolation [20].

Table 4: Hyper-parameter values used for the experiments described in Section 4.

Parameter	Mini-protein	Antibodies
Number of channels, n_c	64	96
Number of bessel basis, d_{bessel}	64	96
Number of blocks, N_{blocks}	4	6
Number of sub-blocks, N_{sub}	3	3
Number of attention heads, N_{head}	2	3
Cut-off distance in Å, r_c	32	64
Maximum tensor degree, l_{max}	1	1
Gradient clip value	0.1 per example	0.1 per example
FAPE loss clip value	10.0	10.0
Number of trainable parameters	2.30M	7.38M

Both models were trained with the FAPE and structure violation losses computed on the output structure of each block with equal weights of 1. A mini-batch size of 8 was used with 8 A100 GPUs in Pytorch's Distributed Data Parallel mode [41]. Model training for the miniprotein experiment was stopped after one day as the validation loss converged. The antibody model was trained for approximately 3.5 days before being trained for additional 7 days at a reduced learning rate of 2×10^{-4}.

To circumvent large memory requirement originating from the quadratic computational complexity of attention layers, we computed gradient updates of the learnable weights of the models on a block-by-block basis, stopping gradient propagation through rotations and translations after each block. Early experiments on mini-proteins showed that this model weight update algorithm did not substantially affect training dynamics. Similarly, we did not observe a significant benefit to using tensors of degrees higher than 1. We leave to future work a more careful benchmarking of different model weight update schemes and other hyper-parameters.