Protein Structure Accuracy Prediction with Deep Learning and its Application to Structure Prediction and Design

Naozumi Hiranuma

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:

David Baker, Chair

Frank Dimaio

Larry Ruzzo

Program Authorized to Offer Degree:

Computer Science

University of Washington

Abstract

Protein Structure Accuracy Prediction with Deep Learning and its Application to Structure Prediction and Design

Naozumi Hiranuma

Chair of the Supervisory Committee:
The Henrietta and Aubrey Davis Endowed Professor in Biochemistry, David Baker
Department of Biochemistry

Understanding the rules of protein structure folding has always been one of the central goals in computational biology. Deep learning is gaining popularity in protein machine learning due to its ability to learn complex functions on large amounts of protein geometry data. To help understand the rules of protein folding better, we developed neural networks (**DeepAccNet** and **Pluto**) that estimate the error in protein models. In other words, these networks estimate how much a computationally modeled protein structure deviates from its experimentally determined conformation. Approximately two million conformations from 21000 protein sequences located at different local energy minima with a large diversity of errors were sampled and used for training. The network uses 3D convolutions to evaluate local atomic environments followed by 2D convolutions to provide their global contexts and outperforms other methods that similarly

predict the accuracy of protein structure models. Overall accuracy predictions for X-ray and cryoEM structures in the PDB correlate with their resolution. The network should be broadly helpful in assessing the accuracy of both predicted structure models and experimentally determined structures and identifying specific regions likely to be in error. The **DeepAccNet** methods were selected as top-performing methods for the estimation of model accuracy (EMA) category in CASP14. We extended the accuracy prediction models for proteins to more general chemistry by training graph neural networks on a wide variety of protein and non-protein datasets. We showed that the resulting framework (**GAAP**) successfully estimates the accuracy of non-protein molecules, such as peptides and Protein-DNA complexes. Our results illustrate how deep learning can impact the efficiency and accuracy of large-scale simulations for both modeling and designing of molecules.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would first like to acknowledge Professor David Baker for guiding my protein machine learning research. I would also like to acknowledge Hahnbeom Park for mentoring me through my Ph.D. program and being a massive contributor to the development of DeepAccNet, especially for its refinement framework. I want to thank Professor Larry Ruzzo and Professor Frank Dimaio for being a part of my reading committee and Professor Eric Klavins for being my GSR. Minkyung Baek, Ivan Anishchenko, and Justas Dauparas significantly contributed to the DeepAccNet development with no particular order of importance. Brian Coventry and Nate Bennett contributed to the analysis of DeepAccNet on protein binder design. Professor Gaurav Bhardwaj, Guangfeng Zhou, and Stephen Rettie contributed significantly to the development and analysis of GAAP on macrocycle peptides. Ryan McHugh contributed to the development and analysis of GAAP on Protein-DNA complexes. David Juergens was helpful for bouncing research ideas. Professor Su-In Lee and the members of Lee lab contributed to the early part of my Ph.D. Finally, I would like to acknowledge Elise Dorough, Erin Kirschner, Luki Goldschmidt, and Lance Stewart for facilitating my research.

DEDICATION

This work is dedicated to my parents and sister, without whose constant support this thesis work was not possible. They are always inspirational to me. At the same time, my gratitude goes to my institution mentors, especially to Hahnbeom Park, whose constant guidance was essential to my thesis work.

Chapter 1. INTRODUCTION

Understanding the folding patterns of protein structures and accurately simulating them has immense implications for many critical real-world problems, such as drug design. Traditionally, the successful approaches predicted protein structures by aligning sequences to evolutionarily related sequences with known structures and then borrowing structural information, such as inter-residue distance, to guide subsequent 3D modeling [1]. In recent years, methods based on deep learning became more popular in the field; in particular, the use of distance map predictions generated by deep neural networks based on amino acid co-evolution data have considerably advanced protein structure prediction by providing more accurate guidance [2]–[4]. In particular, AlphaFold2 and RoseTTAFold took more direct and successful approaches to explicitly predict 3-dimensional atomic coordinates of protein structures using large transformer-based networks [5], [6]. However, these methods cannot predict the structures of non-protein molecules, and some parts of predicted protein structures still deviate from the actual structure [7].

The major challenge in protein structure prediction, design, and refinement is sampling; the space of possible structures that must be searched is vast [8], [9]. For example, suppose it were possible to accurately identify what parts of the computationally modeled structure of a protein or non-protein model are most likely to be in error and how we can alter these regions. In that case, it should be possible to considerably improve the search process by constraining the structural search space.

The methods for estimation of model accuracy (EMA) tackle this problem by assigning accuracy scores for computationally modeled structures. Many methods have been described, including approaches based on deep learning such as ProQ3D (based on per-residue Rosetta energy

terms and multiple sequence alignments with multilayer perceptrons [10]) and Ornate (based on 3D voxel atomic representations with 3D convolutional networks [11]). Non-deep learning methods such as VoroMQA compare a Voronoi tessellation representation of atomic interactions against pre-collected statistics [12]. These methods typically focus only on predicting per-residue structural accuracy and do not suggest how predicted accuracy can be improved.

Some studies have sought to guide refinement using deep-learning-based accuracy predictions directly [13]. However, the most successful refinement protocols in the recent blind 13th Critical Assessment of Structure Prediction (CASP13 and CASP14) test still either utilized very simple ensemble-based error estimations [8] or none at all [14], [15]. This is likely because of the low specificity of most current accuracy prediction methods, which only predict which residues are likely to be inaccurately modeled but not how they should be moved. Hence, they are less helpful in guiding search.

One obvious shortcoming of the current protein structure prediction and accuracy estimate method is that they do not generalize to chemistry outside of proteins. Apart from the apparent reason that the methods are exclusively trained on monomeric protein structures, they usually operate at the resolution of residues, not atoms. This makes it hard to directly apply them to datasets such as Protein-DNA complexes, small molecules, and RNAs; some parts of these datasets do not have amino acid residues. On the other hand, Peptides have similar chemistry to protein monomeric structures. However, they often contain noncanonical or N-methylated residues; these residues cannot be handled by the frameworks expecting 20 canonical amino acids as input. A method that can work with both protein and non-protein molecules is needed.

This document describes our journey to develop state-of-the-art frameworks for estimating the accuracy of protein structures and general non-protein molecules. Chapter 2 describes the

development of deep-learning-based frameworks (**DeepAccNet**, **DeepAccNet-MSA**, and **DeepAccNet-Bert** [16]) that estimate the signed error in every residue-residue distance along with the local residue contact errors. Chapter 3 describes the analysis of the DeepAccNet variants on various performance metrics and compares them to other EMA methods. We show that the DeepAccNet methods are state of the art. Chapters 4 and 6 describe the application of DeepAccNet methods to downstream refinement and protein design processes. Chapter 5 extends the DeepAccNet methods to full-atom resolution using rotationally and translationally invariant graph transformers (**Pluto**). And finally, in Chapter 7, we extend the DeepAccNet methods to general chemistry outside of proteins (**GAAP**), and we show its successful application to peptide macrocycle chemistry and Protein-DNA complexes.

Chapter 2. DEVELOPING THE DEEPACCNET METHODS

This section describes the process of building the DeepAccNet method and its variants. I want to acknowledge Hahnbeom Park, Ivan Anishchanka, Minkyung Baek, and Justas Dauparas for consulting and pitching ideas.

2.1 UNDERSTANDING THE TRAINING OBJECTIVES

We sought to develop model accuracy predictors that provide both global and local information. We developed network architectures that make the following three types of predictions given a protein structure model:

- Local measures of structure accuracy are measured by per residue C_β local distance difference test (referred to as C_β **l-DDT**) scores [17]. C_β l-DDT measures how well the environment in a reference crystal structure is replicated in a decoy structure in distance space. To calculate C_β l-DDT for *i*-th residue R_i, we iterate over all residues R_j within 15Å

of Ri in the reference structure and calculate the fractions of distances between Ri and R_j that are conserved. A distance is considered conserved if it is within a certain tolerance threshold (0.5, 1, 2, 4Å). The final C_β l-DDT score is the average of four fractions. For example, if a residue has 0.9 C_β l-DDT, it means that 90% of distances around the residue are conserved. Global l-DDT is calculated by simply taking an average across all residues.

- A native C_β contact map thresholded at 15Å (referred to as **mask**) is predicted. This is necessary for calculating C_β l-DDT.
- Per residue-pair distributions of signed C_β-C_β distance error against corresponding native structures (referred to as **estograms**; histogram of errors) are also predicted. C_α is taken for GLY. Rather than predicting single error values for each pair of positions, we instead predict histograms of errors (analogous to the distance histograms employed in the structure prediction networks of AlphaFold [4] and TrRosetta [3]). They provide more detailed information about the distributions of possible structures and better represent the uncertainties inherent to error prediction. Estograms are defined over categorical distributions with 15 binned distance ranges; the boundary of bins are at -20.0Å, -15.0Å, -10.0Å, -4.0Å, -2.0Å, -1.0Å, -0.5Å, 0.5Å, 1.0Å, 2.0Å, 4.0Å, 10.0Å, 15.0Å, 20.0Å. The thresholds for middle bins (-4~4Å) were chosen to match the thresholds for the definition of the C_β l-DDT score. The thresholds for outer bins were arbitrarily chosen to give a rough idea of how much a pair of residues should move. In hindsight, it might have been more appropriate to predict bins with higher resolution (e.g., 0.5 Å intervals from -20Å to 20Å) as seen in more recent structural prediction methods (AlphaFold2, RoseTTaFold).

2.2 PREPARING SEQUENCES FOR MONOMERIC PROTEIN STRUCTURES

Training and test sets for protein model structures (called decoys) were generated to most resemble starting models of real-case refinement problems. We reasoned that a relevant decoy structure should meet the following conditions: i) has template(s) not too far or close in sequence space; ii) does not have strong contacts to other protein chains, iii) should contain minimal fluctuating (i.e., missing density) regions. To this end, we picked a set of crystal structures from the PISCES server (deposited by May 1, 2018) containing 20,399 PDB entries with maximum sequence redundancy of 40% and a minimum resolution of 2.5 Å [18]. We further trimmed the list to 8,718 chains by limiting their size to 50-300 residues and requiring that proteins are either monomeric or have minimal interaction with other chains (weaker than 1 kcal/mol per residue in Rosetta energy). HHsearch [19]was used to search for templates; 50 templates with the highest HHsearch probability, sequence identity of at most 40%, and a sequence coverage of at least 50% were selected for model generation.

For the versions of DeepAccNet extended with graph neural networks (discussed later in Chapter 5&7, referred to as Pluto and GAAP), we added new sequences and clusters from the Protein Data Bank (deposited by Feb 18, 2020) [18]; the clusters were formed with the same similarity cutoff, and the resolution cutoff was 4.0 Å this time. This process added an extra 14210 clusters to the original DeepAccNet dataset.

2.3 GENERATING DECOY STRUCTURES WITH VARIOUS PIPELINES

Decoy structures are generated using four methods: 1) comparative modeling, 2) native structure perturbation, 3) deep learning guided folding based on TrRosetta [3], and 4) RoseTTaFold [6].

- For comparative modeling of each protein chain, we repeated RosettaCM 500 [1] times in total, every time randomly selecting a single template from the list. For target proteins lacking templates with GDT-TS > 50 [20], we provided 40% trimmed native structure as templates and generated 500 additional models to increase the coverage of decoy structures at mid-to-high accuracy. We only included the decoy set for a protein chain to the training/test data if the total number of decoys at medium accuracy (GDT-TS to native ranging from 50 to 90) is more than 50. Maximum 15 lowest scoring decoys at each GDT-TS bin (ranging from 50 to 90 with bin size 10) are collected, then the rest with the lowest energy values are filled to make the set contain approximately 90 decoys.
- Native structures are perturbed to generate high-accuracy decoys. 30 models were generated by RosettaCM either by a) combining a partial model of a native structure with high-accuracy templates (GDT-TS > 90) or b) inserting fragments at random positions of the native structure.
- Deep learning guided folding is done using trRosetta [3]. For each protein, 5 subsampled multiple sequence alignments (MSAs) are generated with various depths (i.e., number of sequences in MSA) ranging from 1 to maximum available. The standard trRosetta modeling is run 45 times for each subsampled MSAs.
- The trunk module of RoseTTAfold (`bff_8_4_384_288_last.pt`) was used to generate distograms with log MSA subsampling for the extensions of DeepAccNet (discussed in Chapter 5&7) [6]. This was followed by minimization through the trRosetta minimization script (`m=0, s=0.15`) [3]. We decided to use the script because this

combination was shown to be fast and equally accurate before the recent update of RoseTTaFold.

For DeepAccNet variants, we only used decoys from the first three procedures. The resulting set consisted of about 150 structures (90 from comparative modeling, 30 from native perturbation, and 30 from deep learning guided folding) per each of 7,314 protein chains (6,749, 280, 285 for training, validation, and test datasets), are thoroughly relaxed by Rosetta dual-relax [21] before the usage. For the extensions of DeepAccNet, we added the decoys from the last procedure. The distribution of the starting global l-DDT values of the test proteins is shown in Figure 2.1. The objective was to collect model structures with global l-DDT more than 0.4. All Decoy structures generated for the DeepAccNet models training are available at the GitHub repository https://github.com/hiranumn/DeepAccNet.

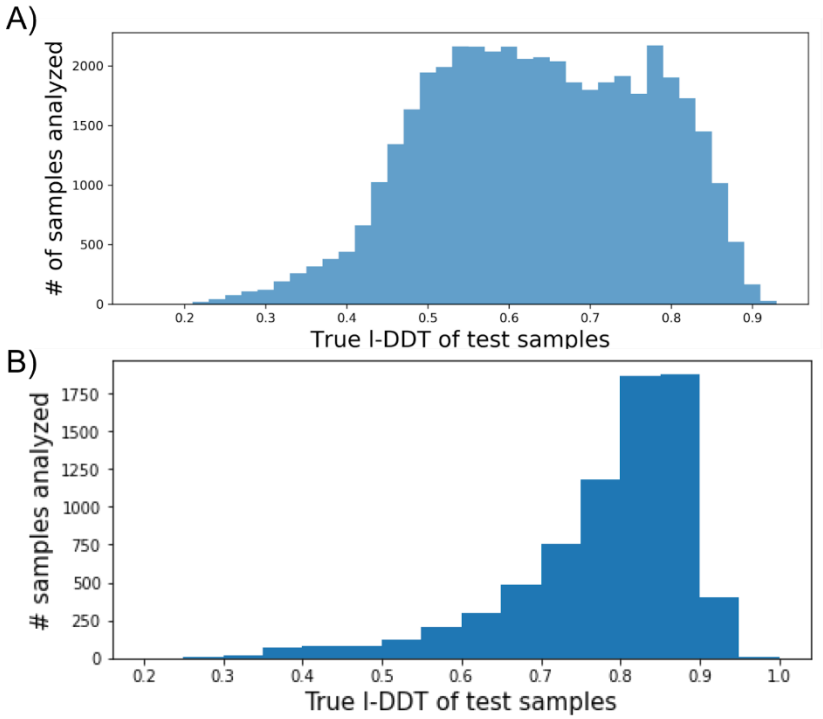

Figure 2.1. Distribution of global l-DDT values in the test set.

A) The test set for the original DeepAccNet, generated for the DeepAccNet methods. B) The test set for the extension of DeepAccNet (described in Chapters 5 and 7). The training, validation, and test splits were selected uniformly. Hence they all have very similar distributions.

2.4 DEFINING MODEL ARCHITECTURES AND INPUT FEATURES

In our original DeepAccNet framework, convolution operations are performed in several dimensions, and different classes of features come in at different entry points of the network (Figure 2.2). Here, we describe the network architecture and classes of features we use.

The first set of input features to the network are voxelized Cartesian coordinates of atoms per residue, generated like Ornate [22]. Voxelization is performed individually for every residue in the corresponding local coordinate frame defined by backbone N, C_α, and C atoms. This representation is translationally and rotationally invariant because projections onto local frames are independent of the global position of the protein structure in 3D space. The second set of inputs are per residue 1D features (e.g., amino acid sequence and properties, backbone angles, Rosetta intra-residue energy terms, and secondary structures) and per residue pair 2D features (e.g., residue-residue distances and orientations, Rosetta inter-residue energy terms).

In the first part of the neural network, the voxelized atomic coordinates go through a series of 3D convolution layers whose parameters are shared across residues. The resulting output tensor is flattened to become a 1D vector per residue, which is concatenated to other 1D features. The second part of the network matches the dimensionality of the features and performs a series of 2D convolution operations. Let us now denote that we have n residues, f_1 1D features, and f_2 2D features. Then, the input matrix of the 1D features M_1 has the shape of n by f_1, and the input matrix of the 2D features M_2 has the shape of n by n by f_2. We tile M_1 in the first and second axis of M_2, concatenating them to produce a feature matrix of size n by n by $2f_1+f_2$. The third axis of the resulting matrix represents vectors of size $2f_1+f_2$, which contain the 2D features and 1D features of i-th and j-th residues. This data representation allows us to convolve over neighboring residues in primary sequence space and pairwise interactions.

The concatenated feature matrix goes through a residual network with 20 residual blocks, with cycling dilation rates of 1, 2, 4, and 8 (see Table 2.1). Cycling dilation allows the network to capture long-range interactions. Then, the network branches off to two arms of 4 residual blocks. These arms separately predict estograms and masks.

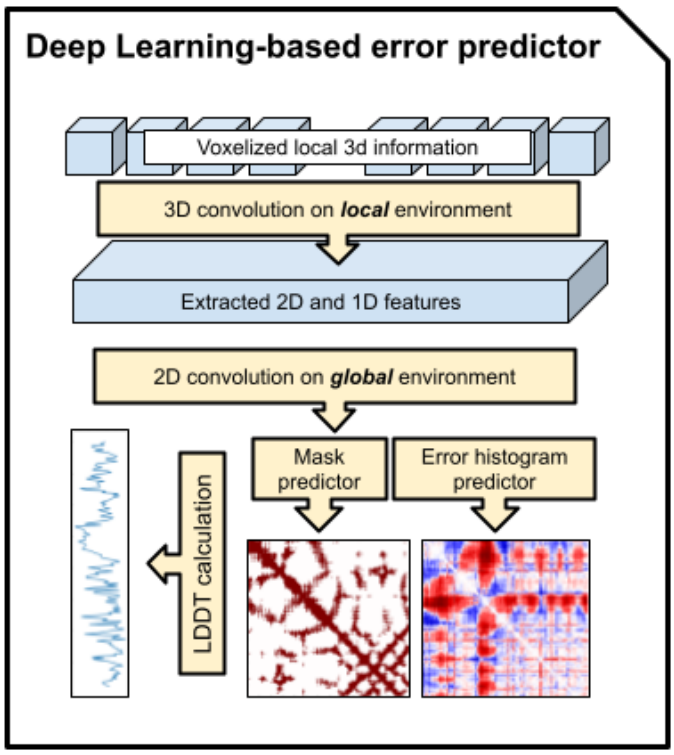

Figure 2.2. The DeepAccNet approach overview.
The deep learning network (DeepAccNet) consists of 3D and 2D convolution operations. The networks are trained to predict i) the signed C_β-C_β distance error distribution for each residue pair (error histogram or **estogram** in short), ii) the native C_β contact map with a threshold of 15Å (referred to as **mask**), iii) the C_β l-DDT score per residue; Cα is taken for GLY. Input features to the network include a) distance maps, b) amino acid identities and properties, c) local atomic environments scanned with 3D convolutions, d) backbone angles, e) residue angular orientations, f) Rosetta energy terms, and g) secondary structure information.

The following tables describe the features and model parameters in more detail. Please refer to the code available on GitHub for further details on the implementation.

Table 2.1. Model architectures details for the DeepAccNet methods.

Layers groups	**Descriptions**
3D convolution layers	This group has four layers of 3D convolution operations with 20, 20, 30, and 20 filters with sizes of 1, 3, 4, 4, respectively. Elu activation is used. Mean pooling of filter size 4 with stride 4 was performed at the end.
Feature merging	This operation merges flattened 3D convolution outputs, 2D, and 1D features. One layer of 2D convolution with 32 filters of size 1 and instance normalization is applied. Elu activation is then used. Finally, the output is upsampled to 256 channels for the following ResNet operations.
Residual blocks 1	Each residual block consists of (i) elu activation, (ii) projection down to 128 channels, (iii) elu activation layer (iv) 3 by 3 convolution, (V) elu activation, (vi) projection up to 256 channels. Instance normalization operations are applied. Residual connection adds inputs to (i) with outputs of (vi). 20 residual blocks are stacked. Dilation is applied to (iv) with a cycling dilation size of 1,2,4,8.
Residual blocks 2 for estograms and masks	Two arms of four residual blocks are applied to predict estograms and masks. The same numbers of channels (256-->128-->256) are used.
C_β l-DDT calculation layers	C_β l-DDT values are calculated within GPU memory based on predicted estograms and masks.
Loss	(i) Estograms are evaluated with categorical cross-entropy loss. (ii) Masks are evaluated with binary cross-entropy loss. (iii) C_β l-DDT values are evaluated with mean squared loss.

Table 2.2. Description of all 9 major feature classes.

Some features are scaled and normalized to a reasonable range. Please refer to the code available on GitHub for further details on the normalization scheme.

Distance-based	i) C_β to C_β distance map, C_α is taken for GLY, ii) C_α to Tip-atom distance map and its transpose, iii) Tip-atom to Tip-atom distance map, and iv) sequence separation map. The distance maps (i~iv) go through a variance reduction process with arcsinh(x).
Amino acid properties	i) One-hot encoded amino acids. ii) Blosum62 scores [23]. iii) Per amino-acid feature sets from Meiler et al [24].
Rosetta energy terms	i) Two-body energy terms: fa_atr, fa_rep, fa_sol, lk_ball_wtd, fa_elec, hbond_bb_sc, and hbond_sc. ii) One-body energy terms: p_aa_pp, rama_prepro, omega, fa_dun. iii) Presence of backbone-to-backbone hydrogen bonds.
Backbone angles and lengths	i) Phi, Psi, and Omega angles. ii) Standardized length between backbone atoms.
residue-residue orientations	i) Full 6 degrees of freedom of translation and rotation. ii) cosine and sine of Dihedral and planar angles defined by Yang et al [3].
Secondary structures	1-hot encoded representation of three state secondary structures given by DSSP solver.
Local atomic environments	24 by 24 by 24 voxels of size 0.8Å. In total, it covers an area of size 19.2Å by 19.2Å by 19.2Å. There are 20 channels for 20 atom types defined by Rosetta. The coordinate frame is fixed based on backbone N, C_a, and C atoms [11].
Multiple sequence alignment	Inter-residue distance (30 by N by N, where N is protein size) predictions from trRosetta give indirect access to evolutionary multiple sequence alignments [3].
Bert embeddings	Attention heads from the last attention layer of the ProtBert-BFD100 model (16 by N by N, where N is protein size)

Table 2.3. Definition of tip atoms.

amino acid	**ALA**	**CYS**	**ASP**	**ASN**	**GLU**	**GLN**	**PHE**	**HIS**	**ILE**	**GLY**
tip atom	CB	SG	CG	CG	CD	CD	CZ	NE2	CD1	CA
amino acid	**LEU**	**MET**	**ARG**	**LYS**	**PRO**	**VAL**	**TYR**	**TRP**	**SER**	**THR**
tip atom	CG	SD	CZ	NZ	CG	CB	OH	CH2	OG	OG1

2.5 CALCULATING PROBABILISTIC L-DDT BASED ON ESTOGRAMS

In the standard calculation of a C_β l-DDT score of *i*-th residue of a model structure, all pairs of C_β atoms that include the *i*-th residue and are less than 15Å in a reference structure are examined. 0.5Å, 1.0Å, 2.0Å, and 4.0Å cutoffs are used to determine the fractions of preserved C_β distances across the set of pairs. The final C_β l-DDT score is calculated by computing the arithmetic mean of all fractional values [17].

However, we do not have access to reference native structures in our setup. Instead, a C_β l-DDT score of *i*-th residue is predicted by combining the probabilistic predictions of estograms and masks as follows:

$$perResLDDT = 0.25 * \frac{p_0 + p_1 + p_2 + p_3}{p_4}$$

p_0 is the mean probability that the magnitudes of C_β distance errors are less than 0.5Å across all residue pairs that have *i*-th residue involved and predicted to be less than 15Å in its corresponding native structure. The former C_β distance errors are obtained from estogram predictions, and the latter native distance information is directly obtained from mask predictions. $p_1 ... p_3$ are similar quantities with different cutoffs for errors; 1.0Å, 2.0Å, and 4.0Å,

respectively. p_4 is the mean probability that native distance is within 15Å, and it is again directly obtained from mask predictions.

2.6 EXTENDING THE NETWORK WITH MSA-BASED FEATURES

As is evident from recent CASP experiments, co-evolution information derived from multiple sequence alignments provides detailed structure information. To this end, we include this as inter-residue distance predictions from the trRosetta network [3]. This gives DeepAccNet indirect access to MSA information. This is an optional input to our network (**DeepAccNet-MSA**) for two reasons: first, all available homology and co-evolutionary information are typically already used in generating the input models for protein structure refinement, and second, in applications such as *de novo* protein design model evaluation, no evolutionary multiple sequence alignment information exists.

DeepAccNet-Bert includes the Bert embeddings from ProtBert-BFD100 model (or Bert, in short), a large language model trained on millions of publicly available protein sequences [25]. Specifically, we used the attention pattern of the last layer, which can be used as embeddings for residue pairs. These embeddings were generated with a single sequence without any evolutionary alignments. Both MSA and Bert features are optionally provided as 2D features.

2.7 OPTIMIZING THE NETWORKS

The networks were trained to minimize categorical cross-entropy between true and predicted estograms and masks. Additionally, as noted, we calculated C_β l-DDT scores based on estograms and masks, and we used a small amount of mean squared loss between true and predicted scores as an auxiliary loss. Therefore, the following weights on the three terms of losses are used.

$$globalLoss = EstogramLoss + 10.0 * lDDTLoss + 0.25 * MaskLoss$$

The weights are selected so that the highest loss generally comes from *EstogramLoss* since estograms are the richest source of information for the downstream refinement tasks. We selected a single decoy from decoy sets of a randomly chosen training protein without replacement at each training step. The decoy sets include native structures, in which case the target estograms ask networks not to modify any distance pairs. An epoch consists of a complete cycle through training proteins, and the training processes usually converge after 100 epochs (Figure 2.3). Our predictions are generated by an ensemble of four models in the same training trajectory with the best validation performance. We used an ADAM optimizer with a learning rate of 0.0005 and a decay rate of 0.98 per epoch [26]. Training and evaluation of the networks were performed on RTX2080 and Titan GPUs.

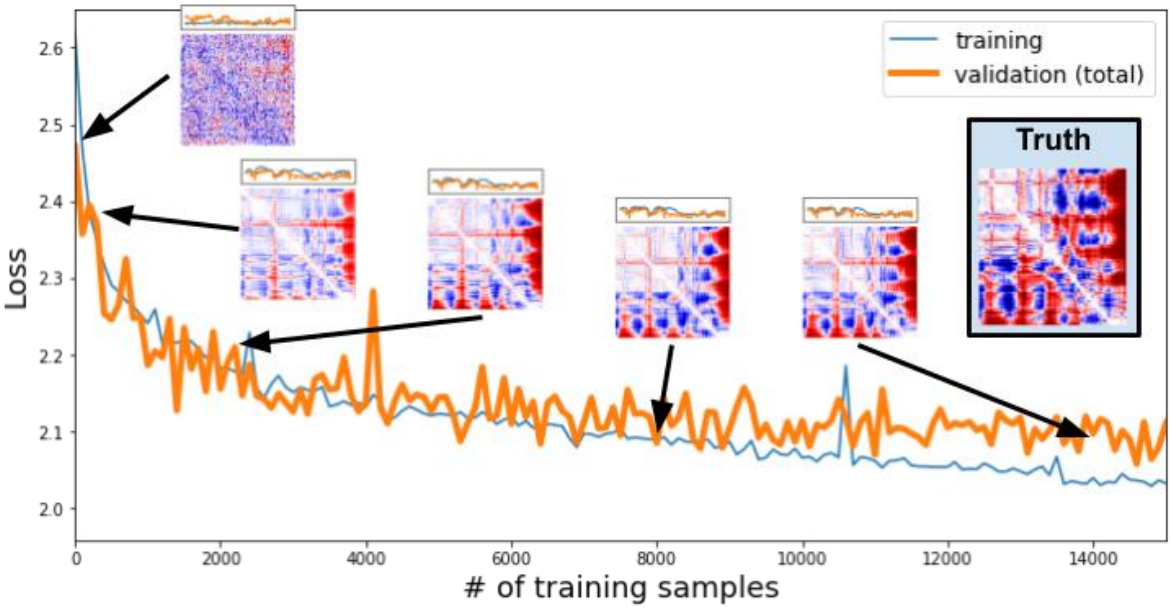

Figure 2.3. The training and validation loss of DeepAccNet-Standard training process. The lines show total loss values, the weighted sum of *EstogramLoss*, *lDDTLoss*, and *MaskLoss* values. The images show the validation prediction of a randomly selected sample along different epochs.

Chapter 3. ANALYZING THE DEEPACCNET METHODS

3.1 DEEPACCNET METHODS ACCURATELY PREDICT STRUCTURE OFFSETS

We first qualitatively look into the prediction from the learned network. Figure 3.1 shows examples of the predictions of DeepAccNet without MSA or Bert embeddings (referred to as **DeepAccNet-Standard**) on two randomly selected decoy structures for each of three target proteins (3lhnA, 4gmqA, and 3hixA) not included in the training process [16]. The images were generated by calculating the expected values of estograms by taking weighted sums of central error values from all bins. For the two bins that encode for errors larger than 20.0 Å and smaller than -20.0 Å, we define the central distance at their boundaries of 20.0 Å and -20.0 Å. In each case, the network generates different signed residue-residue distance error maps for

the two decoys that qualitatively resemble the actual patterns of the structural errors (rows of Figure 3.1). The network also accurately predicts the variations in per residue model accuracy (C_β l-DDT scores) for the different decoys. For example, the left sample from 4gmqA (second row) is closer to the native structure than the other samples are, and the network correctly predicts the location of the smaller distance errors and global l-DDT scores closer to 1.0. Overall, while the detailed predictions are not pixel-perfect, they provide considerable information on what parts of the structure need to move and how to guide refinement.

Predictions from the variants with the MSA (referred to as **DeepAccNet-MSA**) and Bert features (referred to as **DeepAccNet-Bert**) are visualized in Figure 3.2. Qualitatively, predictions from DeepAccNetp-MSA and -Bert are crispier and match the ground truth better in more detail. This observation agrees with the quantitative analysis we conduct in the later sections.

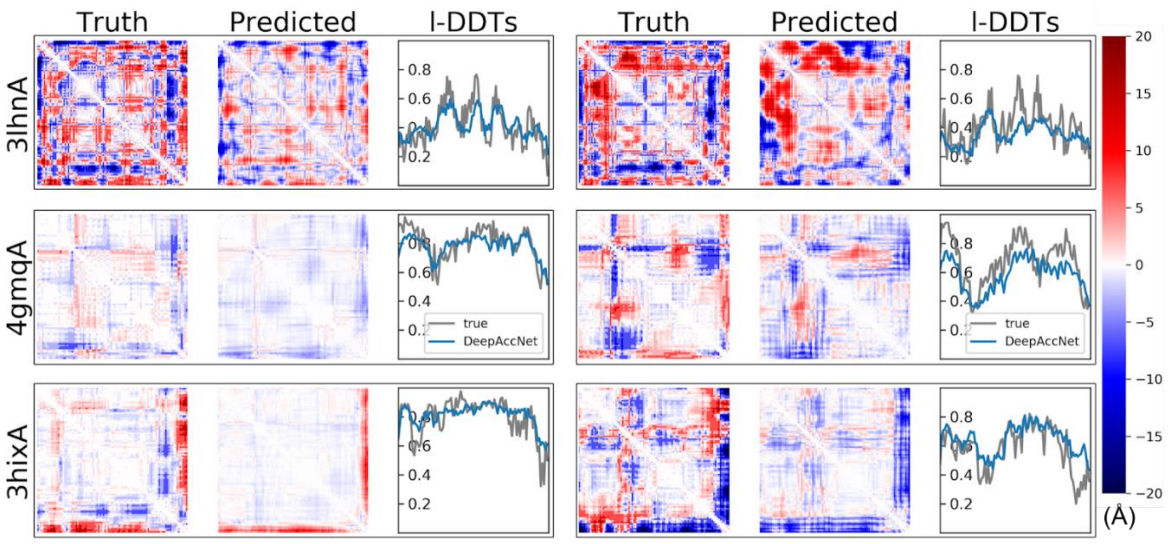

Figure 3.1. Example estograms and C_β l-DDT score prediction.

Model predictions for two randomly selected decoys for three test proteins were randomly selected (3lhnA, 4gmqA, 3hixA; size 108, 92, and 94, respectively; black rectangular boxes delineate results for single decoy). The first and fourth columns show true maps of errors, the second and fifth columns show predicted maps of errors, and the third and sixth columns show predicted and true C_β l-DDT scores. The *i*, *j*-th element of the error map is the expectation of actual or predicted estograms between *i*-th and *j*-th residues in the model and native structure. Red and blue indicate that the pair of residues are too far apart and too close, respectively. The color density shows the magnitude of expected errors.

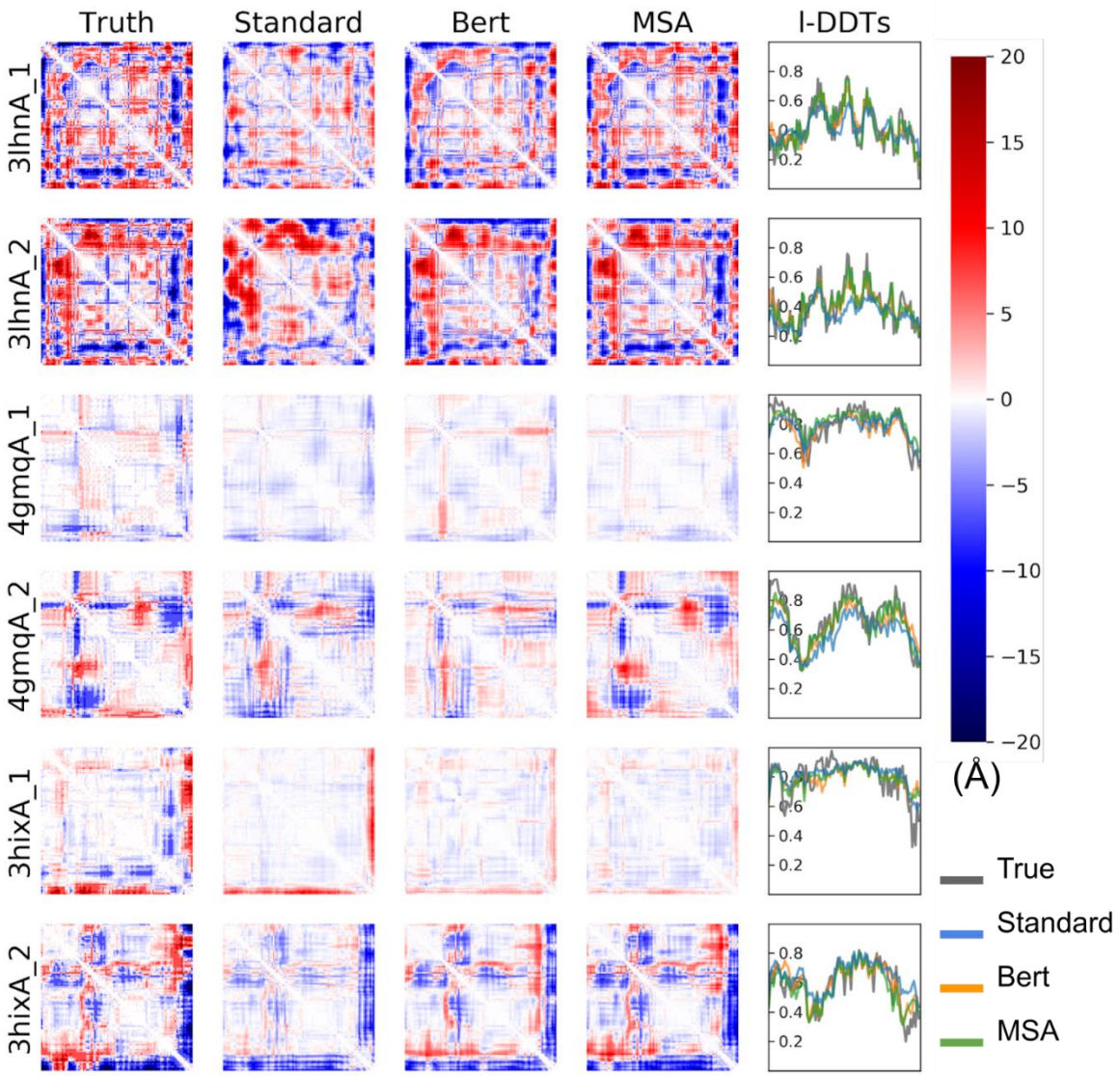

Figure 3.2. Example estograms and C_β l-DDT score prediction from DeepAccNet standard, Bert and MSA.

Model predictions for the same set of decoys from Figure 2 (3lhnA, 4gmqA, 3hixA; size 108, 92, and 94, respectively). The first column shows true maps of errors, the second to fourth columns show predicted maps of errors, and the last column shows predicted and true C_β l-DDT scores. The *i*, *j*-th element of the error map is the expectation of actual or predicted estograms between *i*-th and *j*-th residues in the model and native structure. Red and blue indicate that the pair of residues are too far apart and too close, respectively. The color density shows the magnitude of expected errors.

3.2 LOCAL ATOM INFORMATION, MSA, AND BERT FEATURES ARE THE DOMINANT CONTRIBUTORS TO DEEPACCNET.

We compared the performance of the DeepAccNet networks to that of a baseline network trained only on residue-residue C_β distances. There is no information about residue identities or chemical properties in this baseline network. Thus, it should effectively only capture the generic over/under-packing of average decoy structures from Rosetta.

The performance of the DeepAccNet networks is considerably better on average for almost all the test set proteins (Figure 3.3A; Figure 3.4). They outperform the baseline C_β distance model in predicting estograms for residue pairs across different sequence separations and input distances (Figure 3.3B). The extra MSA or Bert information improves accuracy, particularly for less accurate models and residues (Figure 3.3CD). For all networks and the distance-only network, C_β l-DDT score prediction does not decline substantially with increasing size (Spearman correlation coefficient, or Spearman-r, of -0.04 with p-value > 0.05 for protein size vs. DeepAccNet-Standard performance). However, estogram prediction performance significantly declines for larger proteins (Spearman-r of 0.57 with p-value < 0.00001) (Figure 3.3E). Estimating the direction and magnitude of errors for larger proteins with more interactions over long distances is a much more challenging task since l-DDT scores only consider local changes at short distances; they degrade less with increasing size.

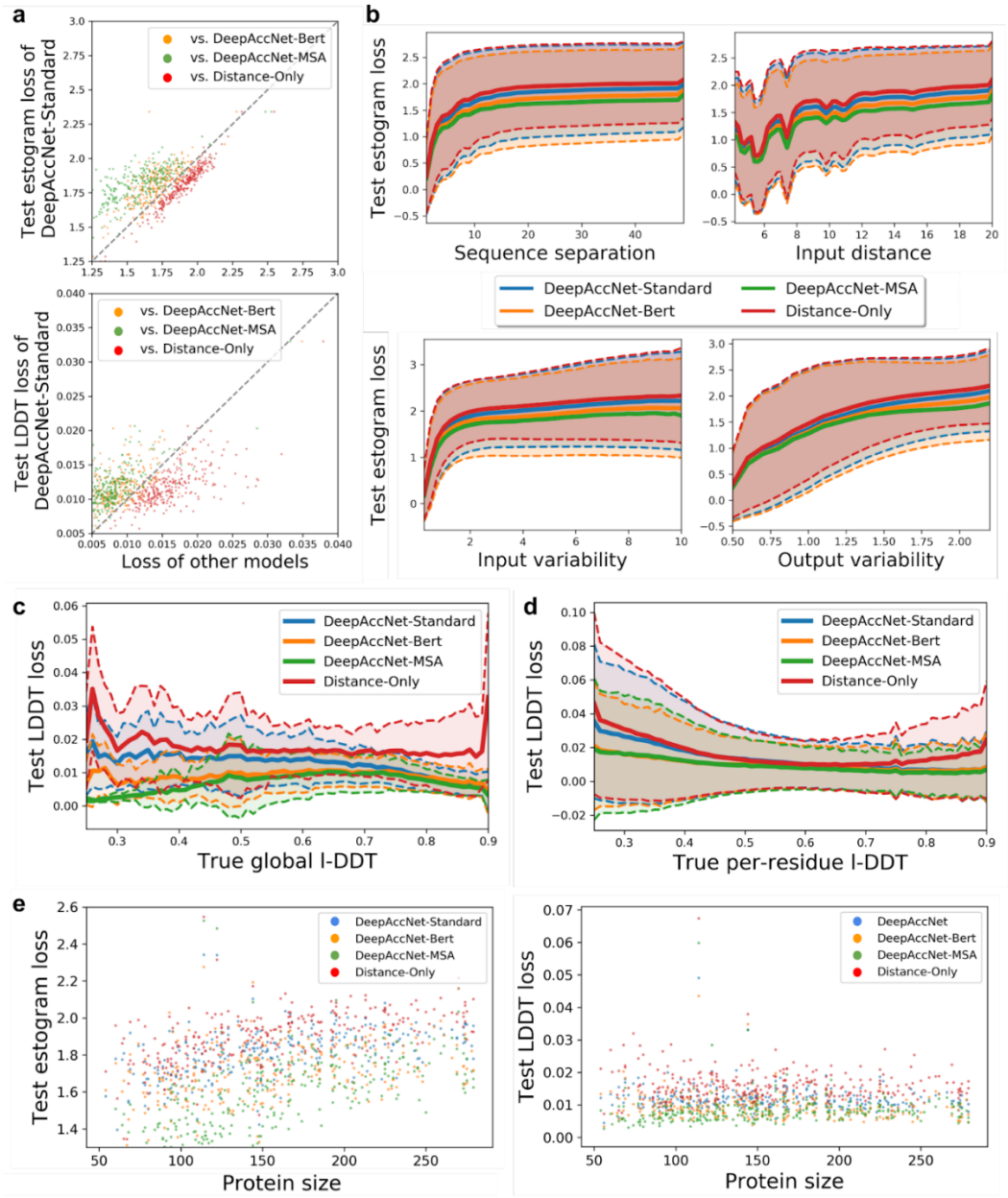

Figure 3.3. Detailed analyses of the DeepAccNet method.

A) Comparison of the variants of DeepAccNet and distance-only network on predicted estograms (top) and C_β l-DDT scores (bottom). Each dot represents the loss for a single protein averaged over all decoys. Lower loss values indicate better performance. Estograms are evaluated by cross-entropy loss, and per residue l-DDT scores are evaluated by mean-squared error. B) Test estogram loss plotted against four conditions: sequence separation, input distance, input variability (standard deviation of input distance across decoys from the same target), and output variability (entropy of true estogram across decoys from the same

target). The loss values are binned in terms of x-axis properties. The mean value at each bin is shown on the y-axis. The range of one z-score is shown with the shaded area. CD) Dependence of l-DDT score loss on true l-DDT per-model (C) and per residue (D). Loss values are binned in terms of the true l-DDT scores. The mean of loss values at each bin is shown on the y-axis as a solid line. The range of one Z-score is shown with the shaded area. E) Dependence of estogram (left) and l-DDT score per residue (right) loss on protein size. Each dot is an average loss value for a single target protein over all its decoys.

In addition to the distance map features, the DeepAccNet and its variants take as input a) amino acid identities and properties, b) local atomic 3D environments for each residue, c) backbone torsion angles and residue-residue orientations, e) Rosetta energy terms, f) secondary structure information, g) MSA, and h) Bert information. To investigate the contributions of each of these features to network performance, we combined each with distance maps one at a time during training. Then, we evaluated performance through the cross-entropy loss of estograms and the mean squared error of C_β l-DDT scores on test sets (Figure 3.4, Table 3.4).

Specifically, we combined each feature class with a distance map one at a time during training (or removed them in one case) and analyzed the loss of predictions on a held-out test protein set. In addition to the DeepAccNet-Standard, -Bert, and -MSA, we trained 8 types of networks: i) distance map only, ii) distance with local atomic environments scanned with 3D convolution, iii) distance with Bert embeddings, iv) ii and iii combined, v) distance with Rosetta energy terms, vi) distance with amino acid identities and their properties, vii) distance with secondary structure information, and iv) distance with backbone angles and residue-residue orientations. We took an ensemble of four models with the best validation performance from the same trajectory for each network to reduce noise. We are aware that more sophisticated feature attribution methods for deep networks exist [27]; however, these methods

attribute importance scores to features per output per sample. Since we have approximately a quarter-million outputs and near a million inputs with a typical 150 residue protein, these methods were not computationally feasible and tractable to analyze.

Apart from the MSA features, the most significant contributions were from the 3D convolution-based features and the Bert embeddings (compare (v), (vi), and (vii)). In addition, there is a statistically significant difference between the network (ii) and (vii), suggesting that the features other than 3D-convolution and Bert facilitate them to work together (p-value < 0.0001 with Wilcoxon signed-rank test for estogram loss between network (ii) and (vii)).

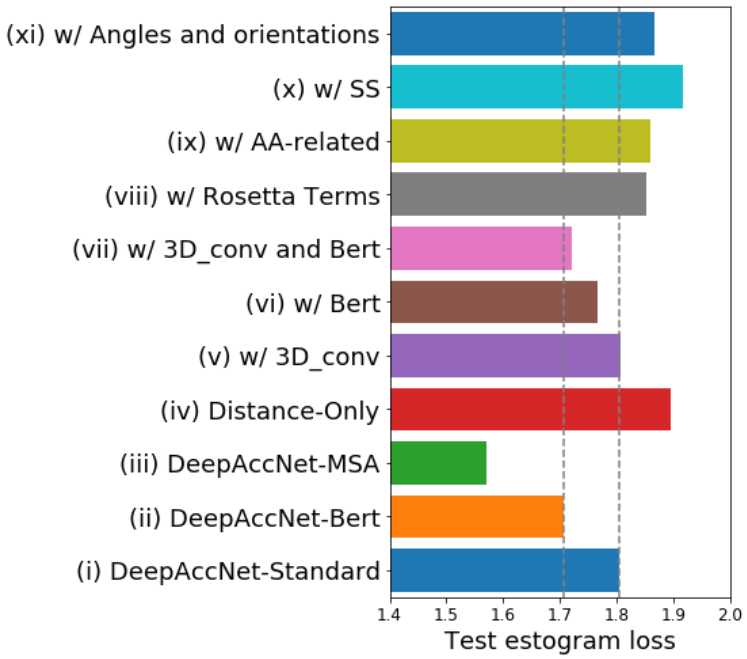

Figure 3.4. Contribution of individual features to network performance
All models include the distance matrix features. The most significant contribution is from the features generated by 3D convolutions on local environments, Bert embeddings, and MSA information. Estogram (cross-entropy) loss values averaged over all decoys for each test protein are shown as one data point. The gray dotted line shows the values from predictors (i) and (ii).

Table 3.4. Result of the ablation study for the DeepAccNet models.

Performance is measured by cross-entropy for estograms and masks and mean squared error for C_β l-DDT scores. We ensembled the prediction from four models with the best validation performance from the same training trajectory for each setting. Columns 2-4 report the quality of the three predictions averaged over all held-out decoy structures. Columns 5-7 report the quality of the predictions on decoys with low true quality (global l-DDT < 0.7). Columns 8-10 report the quality of the predictions on decoys with high true quality (global l-DDT > 0.7).

Models	**Held-out proteins (# proteins=285)**			**True global l-DDT < 0.7**			**True global l-DDT > 0.7**		
	Esto	**Mask**	**l-DDT**	**Esto**	**Mask**	**l-DDT**	**Esto**	**Mask**	**l-DDT**
(i) DAN-Standard	1.805	0.200	0.012	1.939	0.250	0.014	1.567	0.110	0.009
(ii) DAN-Bert	1.697	0.171	0.009	1.781	0.208	0.010	1.548	0.106	0.009
(iii) DAN-MSA	1.557	0.135	0.008	1.594	0.158	0.009	1.489	0.094	0.008
(iv) C_β distance	1.901	0.217	0.017	2.022	0.270	0.017	1.685	0.123	0.016
(v) 3D conv	1.808	0.200	0.012	1.936	0.250	0.013	1.581	0.111	0.010
(vi) Bert	1.761	0.181	0.012	1.836	0.217	0.012	1.628	0.115	0.012
(vii) 3D+Bert	1.714	0.175	0.010	1.794	0.211	0.010	1.570	0.110	0.010
(viii) Rosetta	1.854	0.209	0.013	1.986	0.262	0.015	1.617	0.115	0.011
(ix) AA-related	1.863	0.208	0.014	1.977	0.258	0.014	1.659	0.119	0.014
(x) Sec struct	1.922	0.222	0.017	2.049	0.275	0.018	1.695	0.127	0.015
(xi) Angles and orientations	1.870	0.212	0.015	2.006	0.266	0.017	1.627	0.117	0.012

As is evident from recent CASP experiments, co-evolution information derived from multiple sequence alignments provides detailed structure information; we only include this as an optional input to our network (DeepAccNet-MSA) for two reasons: first, all available homology and co-evolutionary information is typically already used in generating the input models for protein structure refinement and second, in applications such as *de novo* protein design model evaluation, no evolutionary multiple sequence alignment information exists. DeepAccNet-Bert includes the Bert embeddings generated with a single sequence without evolutionary alignments. It outperformed the DeepAccNet-MSA on the EMA tasks for proteins with no homologous sequence information (Figure 3.5). It should be noted that DeepAccNet-MSA would still be a more robust choice when multiple sequence alignment information is available (Table 3.5).

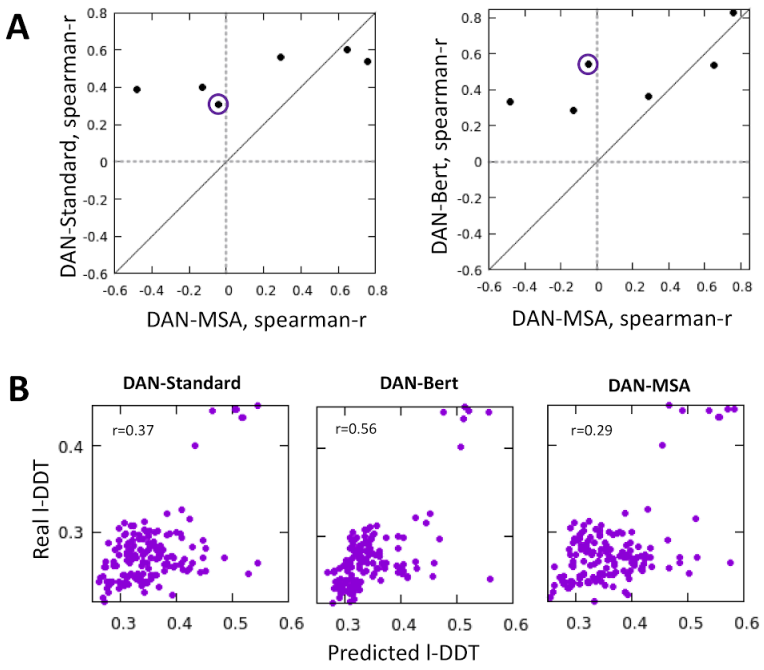

Figure 3.5. DeepAccNet -Bert and DeepAccNet -Standard outperform DeepAccNet -MSA when the protein has no homologous sequence information.

A) Global EMA results of 6 targets from CASP14, which had no homologous sequence (UniClust30 37 January 2020) [28]. Spearman-r between the predicted and the actual global l-DDT across 150 models generated by CASP14 participants is shown for each target. left) DeepAccNet-MSA versus DeepAccNet -Standard, right) DeepAccNet -MSA versus DeepAccNet -Bert; DeepAccNet -MSA on the y-axis and the other on the x-axis. B) Scatter plots of EMA results by DeepAccNet -variants on a CASP14 EMA target T1043 (highlighted by purple circles in panel A).

Table 3.5. Significant tests comparing the DeepAccNet variants and ablation models. Wilcoxon signed-rank test was used to analyze *1~6 as the distribution of the difference between two variants' means is not assumed to be normally distributed. All differences in arithmetic means are statistically significant between variants. For *7, we only have one r-value per variant, unlike *6. Thus, we applied Fisher's Z transformation and analyzed the statistical significance based on the observed z test statistic.

	Standard vs. Bert	**Bert vs. MSA**	**Standard vs. MSA**
Test set (MSE loss of lddt) [*1]	p-value < 0.0001	p-value < 0.0001	p-value < 0.0001
Test set (Cross-entropy loss of Estogram) [*2]	p-value < 0.0001	p-value < 0.0001	p-value < 0.0001
CASP13 (ROC AUC) [*3]	p-value < 0.0001	p-value < 0.0001	p-value < 0.0001
CASP13 (Spearman r) [*4]	p-value < 0.0001	p-value < 0.0001	p-value < 0.0001
CAMEO (ROC AUC) [*5]	p-value < 0.0001	p-value < 0.0001	p-value < 0.0001
CAMEO (PR AUC) [*6]	p-value < 0.0001	p-value < 0.0001	p-value < 0.0001
CAMEO (Spearman r) [*7]	p-value = 0.069	p-value = 0.0003	p-value = 0.080

3.3 DEEPACCNET PREDICTIONS CORRELATE WITH THE RESOLUTION OF EXPERIMENTAL STRUCTURES DETERMINED BY X-RAY CRYSTALLOGRAPHY AND ELECTRON MICROSCOPY.

An effective accuracy prediction method should help evaluate and identify potential errors in experimentally determined structures and computationally modeled structures. We investigated the performance of the networks on experimental structures determined by X-ray crystallography, nuclear magnetic resonance spectroscopy (NMR), and electron microscopy (EM) that were not included in the training set (Chapter 2.2).

Specifically, native structures that were i) not used for model training and validation, ii) monomeric, iii) larger than 40 residues, iv) smaller than 300 residues for the X-ray and NMR structures, and (v) smaller than 600 residues for EM structures were obtained from the PDB [18]. For Figure 3.6C, samples with a resolution larger than 4Å and 5Å were ignored for the X-ray and EM structures, respectively. The histograms in Figure 3.6D are made with all samples. 23,672 X-ray structures, 88 EM structures, and 2,154 NMR structures are in the histograms. For NMR structures, regions highly varying across the models were trimmed. We discarded structures whose remaining residues after trimming were less than 40 or half of the original chain length.

The predicted global l-DDT values by the DeepAccNet variants are close to 1.0 for high-resolution crystal structures, as expected for nearly error-free protein structures. It decreases for lower resolution structures (Figure 3.6A, left panel for DeepAccNet-Standard, Figure 3.6C for DeepAccNet-MSA, and Figure 3.6E for DeepAccNet-Bert). A similar correlation between predicted accuracy and resolution holds for X-ray structures of membrane proteins (Figure 3.6A, middle panel; Spearman-r 0.64 with p-value < 0.0001) and cryoEM structures (Figure

3.6A, right panel; Spearman-r 0.87 with p-value < 0.0001). Note that the good correlation found within the membrane proteins can be simply due to the difference in core packing; whether the network is aware of the membrane environment is unclear from the result. A list of X-ray structures with low predicted global l-DDT despite their high experimental resolution is listed in Table 3.6. Many of these are hemeproteins. As the network does not consider bound ligands, the regions surrounding them are detected as atypical for folded proteins, suggesting that the network may also be useful for predicting cofactor binding and other functional sites from apo-structures. NMR structures have lower predicted accuracies than high-resolution crystal structures (Figure 3.6BDF), which is not surprising given i) they were not included in the training set and ii) they represent solution averages rather than crystalline states. Despite their differences in structural aspects, it is an interesting direction to train an accuracy network with NMR structures in the future.

Table 3.6. List of PDB X-ray native structures with low DeepAccNet scores.
Monomeric proteins in Protein Data Bank with less than 300 residues with low global l-DDT despite their high experimental resolution are shown. Whether a structure has global l-DDT to its experimental resolution was arbitrarily determined by drawing the line at $\mathrm{lddt} = -0.175 * \mathrm{resolutoin} + 0.9$ and capturing all samples that fall under this line.

6B17, 3URO, 3TWG, 5DYR, 6HR0, 1P9G, 4G4L, 6EWN, 4HB6, 5JQF, 4U2W, 4HB8, 1MBN, 4HAJ, 1CYC, 1VXB, 3H4N, 2SBT, 1NXB, 4HBF, 1G7V, 2EWI, 1J0O, 2SNS, 4HDL, 3SJ4, 3H34, 4D5M, 1MBS, 1OS6, 2EWU, 1LWK, 1LYZ, 3TRV, 3SJ0, 4Z0W, 1ACX, 1PMK, 3TJW, 1HH5, 1M1R, 6DK5, 2ZVS, 3D6T, 2AOA, 3SEL, 6FM8, 5YP8, 4EFX, 1TGL, 3SJ1, 1TIA, 2EWK, 2XJI, 5HDD, 6CDX, 5VBD, 4HC3, 3NIR, 2YYX, 1HGU

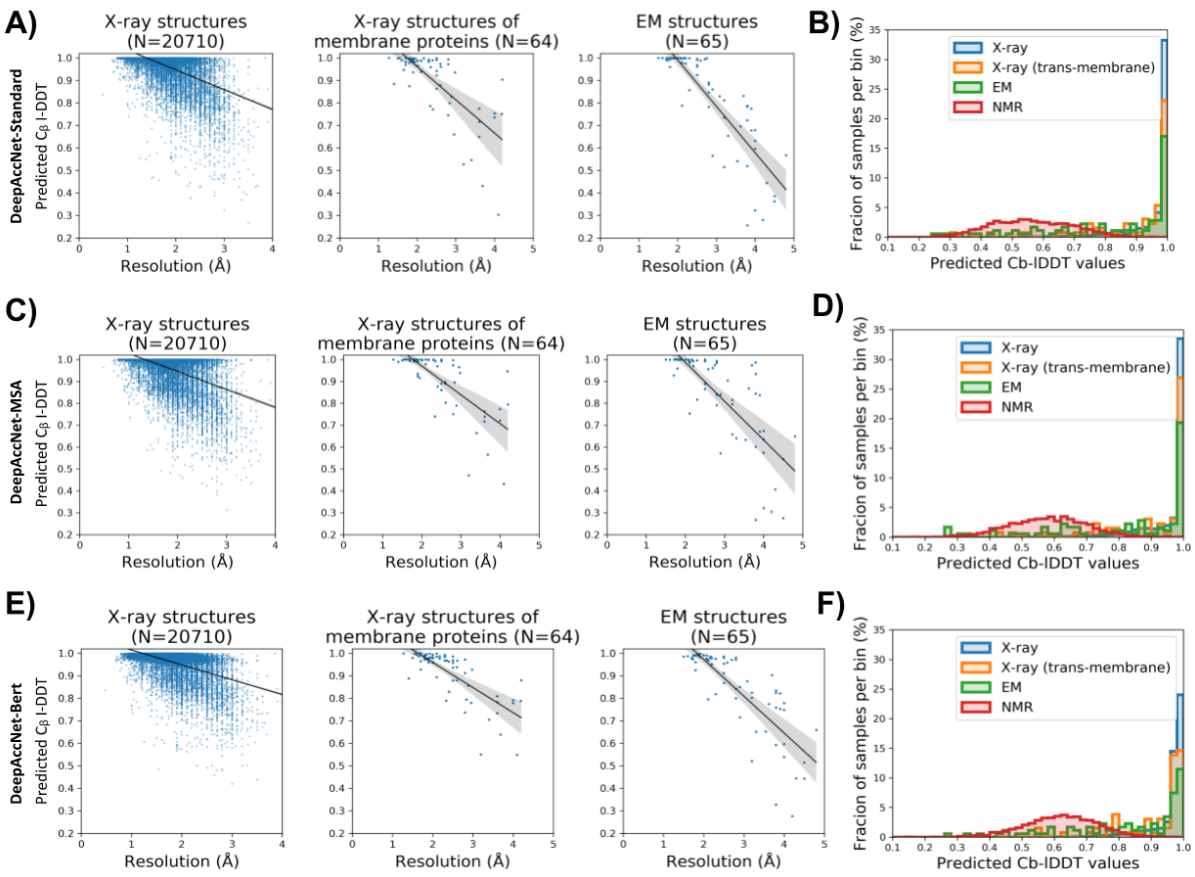

Figure 3.6. DeepAccNet predictions on native monomer structures from PDB.
A) Predicted C_β I-DDT by DeepAccNet-Standard correlates with resolution for X-ray structures (left; Spearman-r 0.48 with p-value < 0.0001), X-ray structures of transmembrane proteins (middle; Spearman-r 0.64 with p-value < 0.0001), and cryoEM structures (right; Spearman-r 0.87 with p-value < 0.0001). B) X-ray structures have higher predicted I-DDT values by DeepAccNet-Standard than NMR structures. CE) Predicted C_β I-DDT by DeepAccNet-Bert (C) and DeepAccNet-MSA (E) correlates with resolutions for X-ray structures (left; Spearman-r 0.43 and 0.44 with p-value < 0.0001 for the Bert and MSA variants, respectively), X-ray structures of transmembrane proteins (middle; Spearman-r 0.73 and 0.74 with p-value < 0.0001 for the Bert and MSA variants, respectively), and cryoEM structures (right; Spearman-r 0.82 and 0.84 with p-value < 0.0001 for the Bert and MSA variants, respectively). DF) X-ray structures have higher predicted I-DDT values by DeepAccNet-Bert and -MSA than NMR structures.

3.4 THE DEEPACCNET METHODS SHOW STATE-OF-THE-ART PERFORMANCE ON THE EMA TASKS.

We compared the performance of the DeepAccNet variants on the CASP13 EMA data (76 targets with approximately 150 decoy models each) to that of the methods that similarly estimate the error of a single computationally modeled structure. These methods include Ornate (group name 3DCNN) [11], a method from Lamoureux Lab [29], VoroMQA [12], ProQ3 [10], ProQ3D, ProQ3D-lDDT [10], and MODFOLD7 [30]; the former two use 3D convolutions similar to those used in our single residue environment feature calculations.

Specifically, we downloaded these competitors' submissions for the CASP 13 accuracy estimation category. The latter six methods submitted their predictions for 76 common targets, whereas Ornate only submitted for 55 target proteins. Thus, we decided to analyze predictions on the 76 common target proteins from all methods except for Ornate, which only evaluated 55 target proteins. The evaluation was performed in two metrics; i) Spearman-r of predicted quality scores across decoys of each target, and ii) area under the ROC curve for predicting mismodeled residues of each sample (C_β l-DDT< 0.6). The latter metric is one of the official CAMEO metrics for local accuracy evaluation [31]. Samples whose residues are all below or above 0.6 C_β l-DDT are omitted. For assessing the performance of methods other than ours, their submitted estimations of global quality scores were evaluated against the true full-atom global l-DDT scores.

According to both metrics, DeepAccNet-Standard and DeepAccNet-Bert outperformed the other methods that do not use any evolutionary information; DeepAccNet-MSA also outperformed the other methods that use evolutionary multiple sequence alignment information (Figure 3.7).

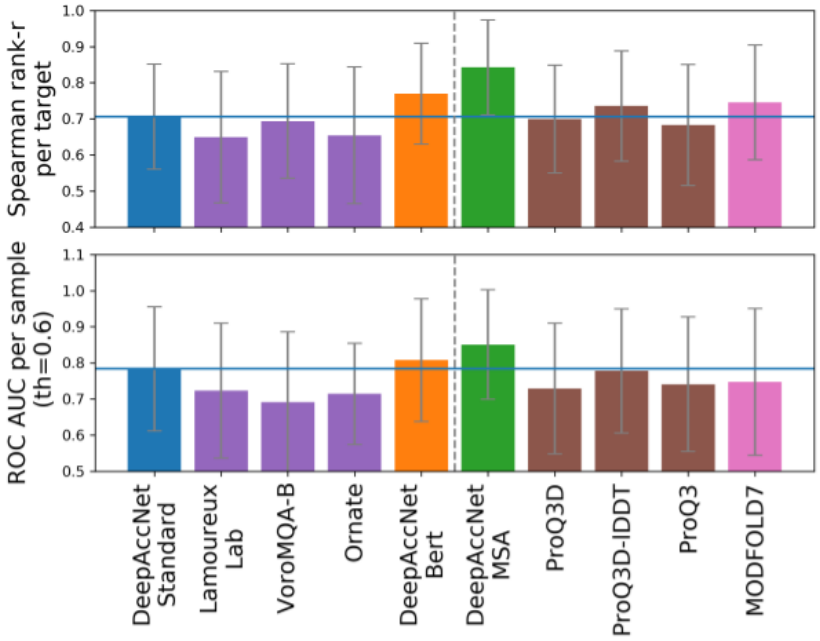

Figure 3.7. Comparison of the performance of single model accuracy estimation (EMA) methods on CASP13 data.

(top) Performance of global accuracy estimation measured by the Spearman correlation coefficient (r-value) of predicted and actual global l-DDT scores per target protein. (bottom) Performance of local accuracy estimation measured by area under receiver operator characteristic (ROC) curves for predicting mismodeled residues per sample (C_β l-DDT< 0.6). The blue horizontal lines show the value of DeepAccNet-Standard. The methods to the left of the dotted line do not use coevolutionary information. The quasi-single EMA method is shown in pink. Error bars show standard deviation.

While this improved performance is very encouraging, it must be noted that our predictions are made after rather than before the CASP13 data release, so the comparison is not entirely fair. Future blind accuracy prediction experiments are necessary to compare methods on an even footing. As a step in this direction, we tested performance on structures released from the PDB after our network architecture was finalized. This corresponds to the CAMEO (Continuous Automated Model EvaluatiOn) experiment between 2/22/2020 to 5/16/2020 [31]. We collected 206 targets with approximately 10 modeled structures on average. We downloaded submissions of "Baseline potential", EQuant2, ModFOLD4, ModFOLD6, ModFOLD7_LDDT, ProQ2, ProQ3, ProQ3D, ProQ3D_LDDT, QMEAN3, QMEANDisco3, VoroMQA_sw5, and VoroMQA_v2. Some methods did not submit their predictions for all samples, and those missing predictions were ignored from the analysis. The DeepAccNet methods generate predictions for all samples. We consistently observed that DeepAccNet-Standard and DeepAccNet-Bert improved on other methods that do not use evolutionary information, -- namely, VoroMQA [12], QMEAN3 [32], and EQuant2 [33] in both global (entire model) and local (per residue) accuracy prediction performance (Figure 3.8). DeepAccNet-MSA also showed state-of-the-art performance among the methods that use multiple sequence alignment. We could not compare signed residue-pair distance error predictions because the other methods do not predict this.

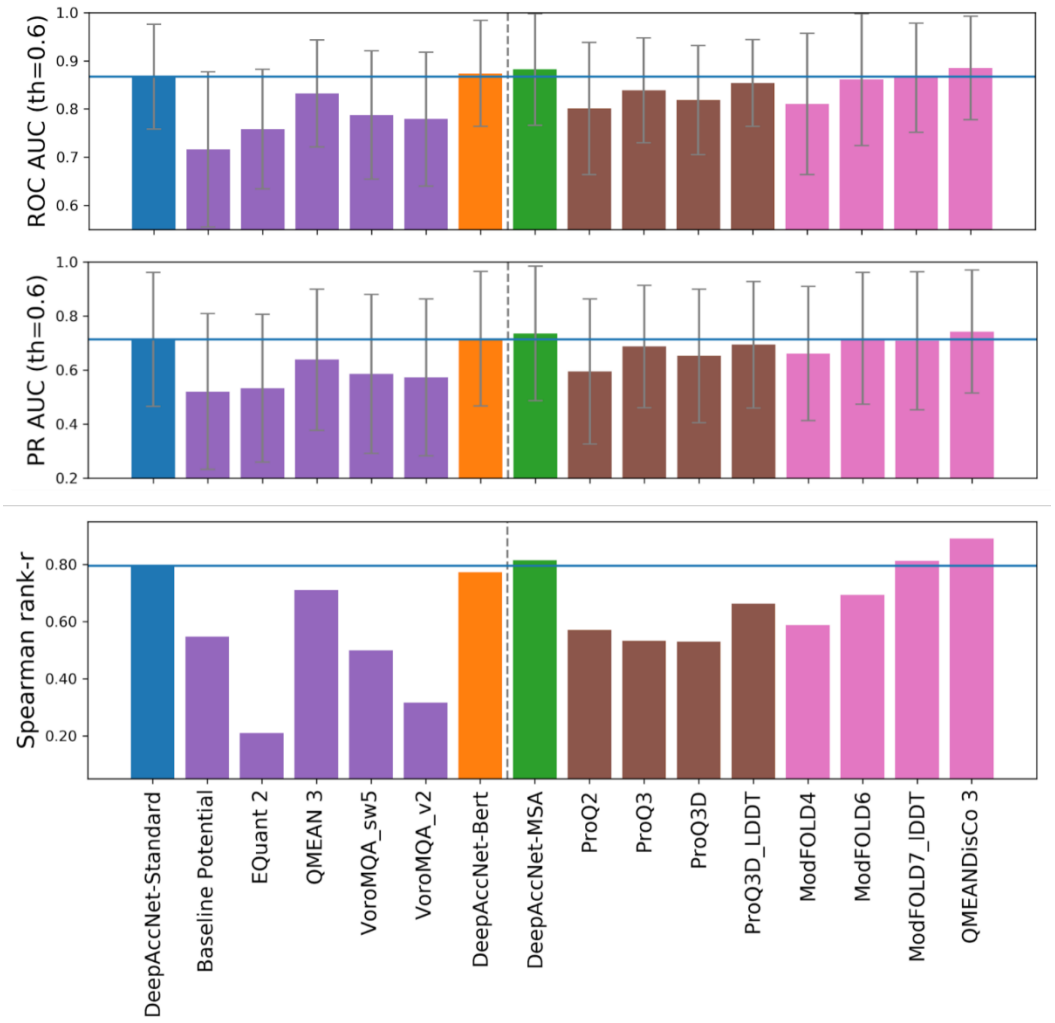

Figure 3.8. Comparison of the performance of single model accuracy estimation (EMA) methods on CAMEO data.

(Top, middle) Performance of local accuracy estimation measured by area under receiver operator characteristic (ROC, top) curve and precision-recall curve (PR, middle) for predicting mismodeled residues per sample (C_β l-DDT< 0.6). Error bars show standard deviation. (Bottom) Performance of global accuracy estimation measured by the Spearman correlation coefficient (r-value) of predicted and actual global l-DDT scores. Since the number of models per target was small, the correlation was measured globally across all targets. The blue horizontal lines show the value of DeepAccNet-Standard. The methods to the left of the dotted line do not use coevolutionary information. The quasi-single models are shown in pink.

As a rigorous blind test of the DeepAccNet methods, we entered them in the CASP14 structure prediction experiment. Out predictions were analyzed by the independent assessors and presented at the CASP14 meeting. In the EMA category, we registered DeepAccNet-standard as "BAKER-experimental (group403)" and DeepAccNet-MSA as "BAKER-ROSETTASERVER (group209)". DeepAccNet-Standard and DeepAccNet-MSA were the top single-model methods for global QA (top1 loss). In addition, DeepAccNet-MSA was the best single-model method for local QA, and DeepAccNet-Standard was the best single-model local QA method that does not use any co-evolutionary information [34], [35]. These results suggest that our DeepAccNet methods improve over the previous state-of-the-art EMA methods.

3.5 SUMMARY

Representations of the input data are critical for the success of deep learning approaches. In the case of protein structures, a complete description is the full Cartesian coordinates of all atoms. However, these are not invariant to translation and rotation, and they are not optimal for predicting rotationally invariant quantities such as error metrics. Hence most previous accuracy prediction methods based on machine learning have not used the full atomic coordinates [10], [12], [36]. The previously described Ornate method does use atomic coordinates to predict accuracy and solves the rotation dependence by setting up local reference frames for each residue [11]. As in the Ornate method, DeepAccNet carries out 3D convolutions over atomic coordinates in frames defined per residue. We go beyond Ornate by integrating this detailed residue information with additional individual residue and residue-residue level geometric and energetic information by 2D convolutions over the full N x N residue-residue distance map. DeepAccNet-Bert further employs the sequence embeddings

from the ProtBert language model [25], which provides a higher-level representation of the amino acid sequence more directly relatable to 3D structures.

Performance evaluation on CASP13, CAMEO, and CASP14 datasets shows that the DeepAccNet networks make state-of-the-art accuracy predictions. They are the first to our knowledge to predict signed distance errors for protein structure refinement. Model quality estimations on X-ray crystal structures correlate with resolutions, and the network should also help identify errors in experimentally-determined structures (Figure 3.6). DeepAccNet performs well on both cryoEM and membrane protein structures, and it could be beneficial for low-resolution structure determination and modeling of currently unsolved membrane proteins (Figure 3.6). We also anticipate that the network is handy in evaluating protein design models.

3.6 IMPLEMENTATION AND CODE AVAILABILITY.

Decoy structures generated for the training of the DeepAccNet models and their raw predictions on the held-out test, CASP13, and CAMEO set are available at the GitHub repository https://github.com/hiranumn/DeepAccNet. Code and accompanying scripts for the model accuracy predictors (DeepAccNet-Standard, DeepAccNet-MSA, and DeepAccNet-Bert) are implemented and made available at https://github.com/hiranumn/DeepAccNet

Chapter 4. APPLYING THE DEEPACCNET METHODS TO PROTEIN REFINEMENT.

We next experimented with incorporating the DeepAccNet accuracy predictions into the Rosetta refinement protocol [8], [37], which was already one of the top methods tested in CASP13 [38]. I want to acknowledge Hahnbeom Park for significantly contributing to this section.

4.1 INTEGRATING DEEPACCNET TO THE REFINEMENT PROTOCOL.

Rosetta's high-resolution refinement starts with a single model. The first diversification stage explores the energy landscape around the model using a set of sampling operators. Then, in a subsequent iterative intensification stage, it hones in on the lowest energy regions of the space. Search is controlled by an evolutionary algorithm that maintains a diverse but low energy pool through many iterations/generations. The bottleneck to improving refinement has essentially become sampling close to the correct structure, with improvements in the Rosetta energy function in the last several years [39], [40]. The original protocol utilized model consensus-based accuracy estimations (i.e., regional accuracy estimated as inverse of fluctuation within an ensemble of structures sampled around the input model) to keep the search focused on the relevant region of the space. These have the obvious downside of limiting exploration in regions that need to change substantially from the input model but are in deep false local energy minima.

To guide the search, estograms and I-DDT scores were predicted and incorporated at every iteration in the Rosetta refinement protocol at three levels (Figure 4.1, details in Appendix). First and most importantly, the estograms were converted to residue-residue interaction

potentials with weight for each pair defined by a function of its estogram prediction confidence. These potentials were added to the Rosetta energy function as restraints to guide sampling. Second, the per-residue l-DDT predictions were used to decide which regions to sample intensively or recombine with other models. Third, the global l-DDT prediction was used as the objective function during the selection stages of the evolutionary algorithm and to control the model diversity in the pool during iteration.

For modeling a single structure at both diversification and intensification stages, first, unreliable regions in the structure are estimated from accuracy prediction. Then, structural information is removed in those regions and fully reconstructed from scratch. Next, fragment insertions are carried out in a coarse-grained broken-chain representation of the structure, focusing more on unreliable regions (5 times more frequently than the rest), followed by repeated side-chain rebuilding and minimization in all-atom representation. Finally, both coarse-grained and all-atom stage modeling are guided by distance restraints derived from accuracy predictions in addition to Rosetta energy. The Appendix reports the details of unreliable region predictions, recombination iteration, and restraints.

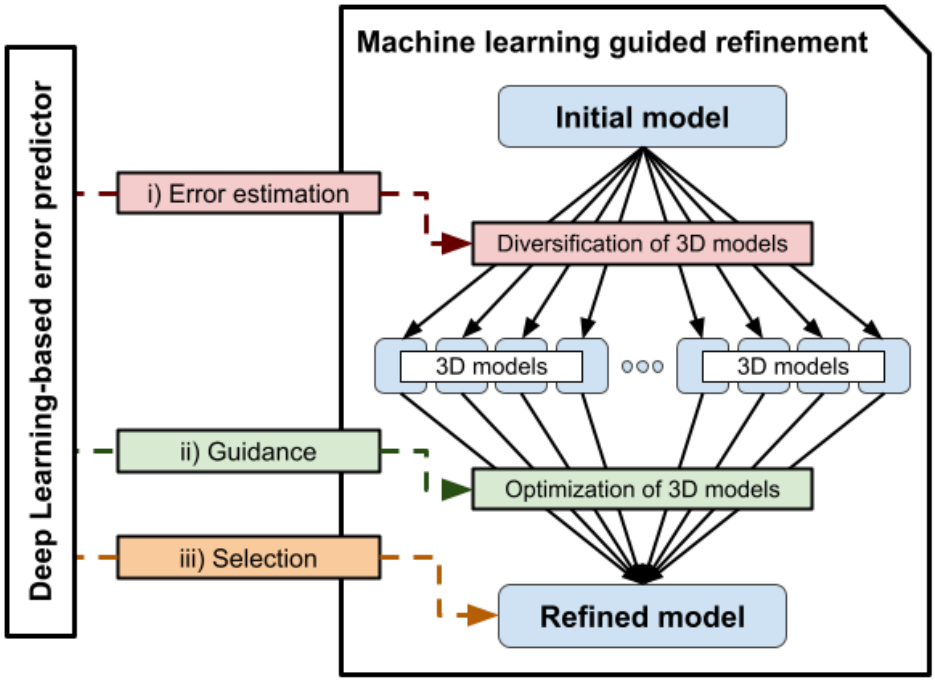

Figure 4.1. The refinement approach overview.

The refinement protocol tested in this work inherits the framework from previous study 5. The overall architecture consists of two stages: the first initial model diversification stage, followed by iterative model intensification stages where a pool of structures is maintained during optimization by an evolutionary algorithm. At the diversification stage, following accuracy estimation of the single starting model, two thousand independent Rosetta modeling are attempted using RosettaCM. In the iterative annealing stage, a series of accuracy estimation, new structure generation, and pool selection steps are repeated iteratively. At each iteration, 10 model structures are selected from the current pool. Individual accuracy predictions are made for each of 10 structures to guide the generation of 12 new model structures starting from each (total 120). A new pool with a size of 50 is selected among 50 previous pool members plus 120 newly generated ones with criteria of i) the highest global l-DDT estimated and ii) model diversity within the pool. This process is repeated for 50 iterations. At every fifth iteration, a recombination iteration is called instead of a regular iteration where model structures are recombined with another member in the pool according to the residue l-DDT values predicted by the network (see Appendix).

4.2 THE DEEPACCNET-GUIDED REFINEMENT PROTOCOL IMPROVES ON THE PREVIOUS STATE-OF-THE-ART

73 protein refinement targets were collected from previous studies [8], [37] to benchmark the accuracy prediction guided refinement protocol. The starting structures were generally the best models available from automated structure prediction methods. A separate 7 targets from Park et al. [8], [37] were used to tune the restraint parameters and excluded from the benchmarking.

We found that network-based accuracy prediction consistently improves refinement across the benchmark examples. In Figure 4.2, refinement guided by the accuracy predictions from DeepAccNet-Standard is compared to our previous protocol in which non-deep learning accuracy estimation was used. Refinement of many proteins in the benchmark set was previously challenging due to their size [37]. However, with the new protocol, consistent improvements are observed over the starting models regardless of protein size (Figure 4.2A, the l-DDT improve by 10% on average) and over the models produced with our previous unguided search (Figure 4.2B; the I-DDT improves by 4% on average). The number of targets with l-DDT improvements of greater than 10% increases from 27% to 47% using DeepAccNet-Standard to guide refinement. These improvements are notable given how challenging the protein structure refinement problem is (comparison to other best predictors on the CASP13 targets is shown in Figure 4.3). For reference, best improvements between successive biannual CASP challenges are typically < 2% [38]. Tracing back through the refinement trajectory reveals that the progress in predicted and actual model quality occurs gradually through the stages and correlates well to each other (Figure 4.4A). Predictions of more detailed per residue model quality also agree with their actual values (Figure 4.2E).

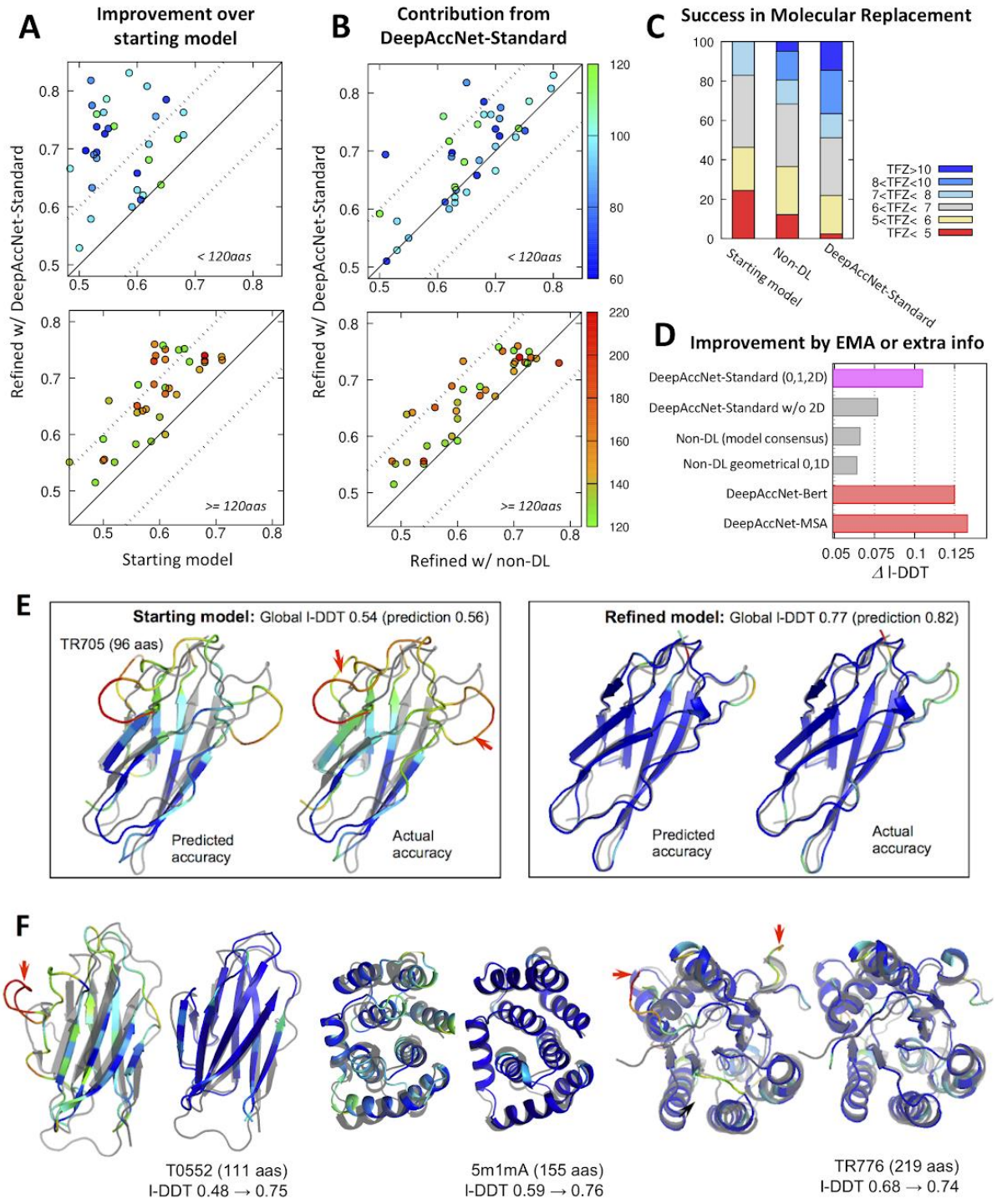

Figure 4.2. Consistent improvement in model structures from refinement runs guided by deep-learning-based accuracy predictions.

Refinement calculations guided and not guided by network accuracy predictions were carried out on a 73 protein target set [8], [37]. A) Network guided refinement consistently improves

starting model. B) Network-guided refinement trajectories produce more significant improvements than unguided refinement trajectories. The accuracy of the refined structure (l-DDT; y-axis) is compared to that of the starting structure in panel A and the final refined structure using non-DL-based model consensus accuracy predictions in panel B. The top and bottom panels show results for proteins less than 120 residues in length and 120 or more residues in length, respectively. Each point represents a protein target with color indicating the protein size (scale shown on the right side of panel B). C) Molecular replacement experiments on 41 benchmark cases using three different sets of models: i) starting models, ii) refined models from the non-deep learning protocol, and iii) guided by DeepAccNet-Standard. Distributions of TFZ (translation function Z-score) values obtained from Phaser software [41] are reported; TFZ values greater than 8 are considered robust MR solutions. D) Model improvements brought about by utilizing DeepAccNet-Standard (magenta), different EMA methods (gray bars), and other DeepAccNet variants trained with Bert or MSA features (red bars). Average improvements tested on the 73 target set are shown. For the "DeepAccNet-Standard w/o 2D" and "geometrical EMA" [12], residue pair distance confidences are estimated by the multiplication of residue-wise accuracy following the scheme in our previous work [8], [37]. E) Example of predicted versus actual per-residue accuracy prediction. Predicted and actual l-DDT values are shown before (left) and after refinement (right) with a color scheme representing local l-DDT from 0.0 (red) to 0.7 (blue). The native structure is overlaid in gray color. Red arrows in the panels highlight major regions that have been improved. F) Examples of improvements in refined model structures. For each target, starting structures are shown on the left and the refined model on the right. The color scheme is the same as E, showing the actual accuracy.

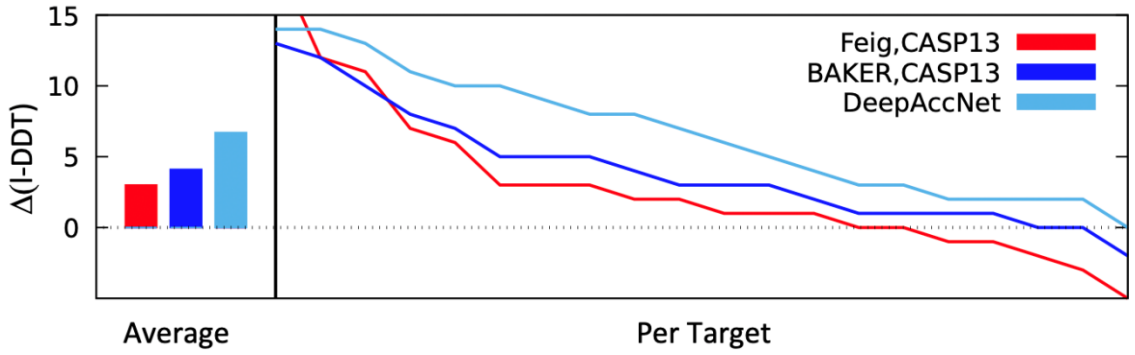

Figure 4.3. Performances of the methods on CASP13 refinement category targets. Improvements in l-DDT scores over starting models are shown. Two leading groups in CASP13, Feig and Baker, are brought in for the comparison against refinement with DeepAccNet; the Feig group ran long MD simulations, while the BAKER group ran the non-DL refinement method presented in the main text with subsequent short MD simulations. Net l-DDT changes for both groups range within 3~4%, compared to 7% by DeepAccNet-guided refinement. 9 targets from the CASP13 refinement category are removed from the analysis for which the native structures contain heavy oligomeric contacts or are determined at low resolutions (>3Å).

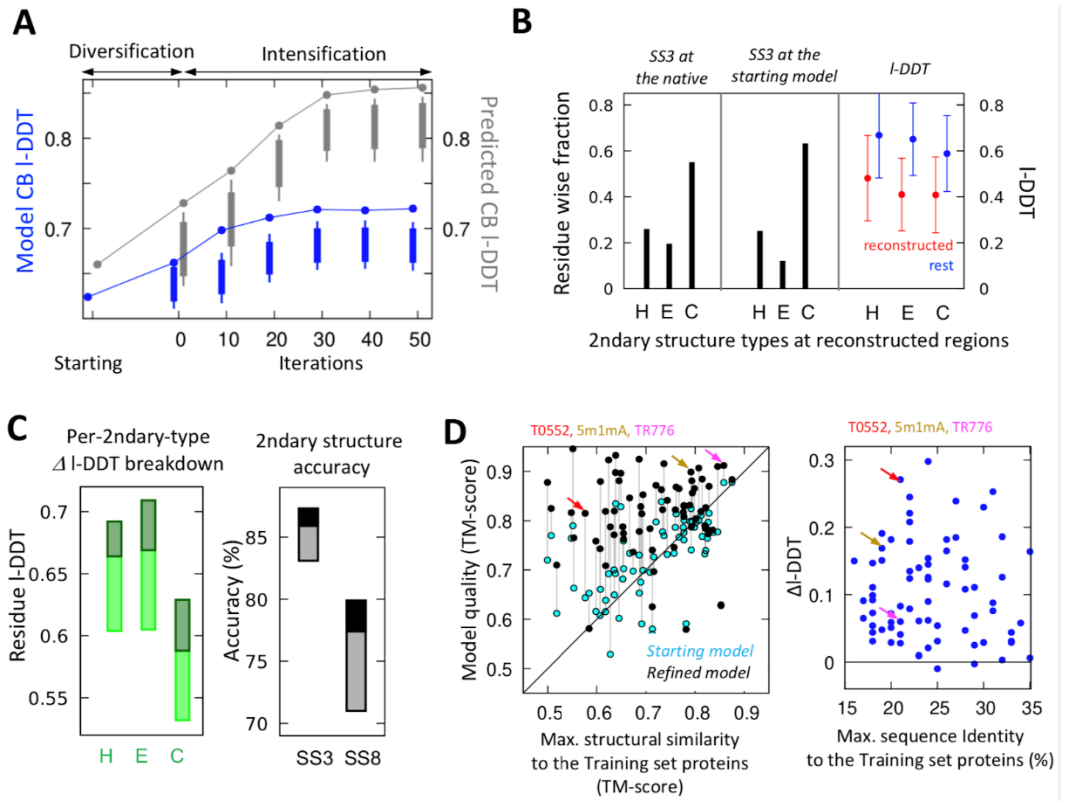

Figure 4.4. Detailed analyses of refinement results.

A) Actual and predicted model accuracy improvements throughout the refinement trajectory. Model quality (actual in blue and predicted in gray, C_β l-DDT is used for direct comparison), averaged over 73 benchmark cases, is shown through the refinement process. Points and bars show the model1 quality and the quality range of 50 models in the pool, respectively. B) 3-state secondary structure type at the reconstructed regions (H: helix, E: extended, C: coil). Residue-wise fractions of each type are plotted according to the native structure (left) and the starting model structure (middle), respectively. (right) Pre-refinement l-DDT values at reconstructed regions and the rest of preserved regions are shown in red and blue colors, respectively (average by circles; standard deviations by error bars). C) Breakdown of accuracy improvements by secondary structure types. Light-colored boxes represent improvements without DeepAccNet-Standard, while darker boxes represent additional improvements gained with DeepAccNet-Standard; these are calculated over the complete benchmark set. (left panel) Similar improvements are observed across secondary structure types. (right panel) Improvements in model secondary structure accuracy are evaluated on 3- or 8-states following DSSP annotations [42]; improvements are evident in both 3 state and 8

state local structure prediction. D) Correlation between refinement performance and highest structural/sequence similarity of the target to the training set proteins. (left panel). Correlation between the maximum structural similarity (x-axis) versus the starting/refined model quality (y-axis) is shown in TM-score [43]. (right panel) Correlation between the maximum sequence identity (%) and the refinement performance (in l-DDT change). In both panels, targets highlighted in Figure 4.2 are shown in colored arrows.

4.3 REFINED MODELS WITH DEEPACCNET RESULT IN MORE ROBUST MOLECULAR REPLACEMENT HITS.

We evaluated the practical impact of improved quality of model refinement using the accuracy predictions by carrying out molecular replacement (MR) trials with experimental diffraction datasets (Figure 4.2C). On 41 X-ray data sets taken from the benchmark set, we obtained robust MR hits for 0%, 20%, and 37% of the cases, using pre-refined models, models refined by the non-deep learning protocol, and models refined using DeepAccNet-Standard, respectively.

Specifically, of 50 target native structures determined by X-ray crystallography in the benchmark set, 41 are tested for MR. 9 targets are excluded as their crystal structures contained other proteins or domains with significant compositions (>50%). Phaser [41] in the Phenix suite version 1.18rc2-3793 is applied with MR_AUTO mode. Terminal residues are trimmed from model structures before MR if they do not directly interact with the rest of the residues. B-factors are estimated by taking residue-wise DeepAccNet predictions: first, u_i, the position error at *i*-th *residue* (in Å), is estimated by using a formula:

$$u_i = 1.5 * \exp\left[4 * (0.7 - PredictedLDDT_i)\right]$$

, where parameters were pre-fit to the training set decoy structures. B-factor at *i*-th residue is calculated as $8\pi^2 u_i^2/3$.

4.4 ESTOGRAMS, MSA, AND BERT ARE ESSENTIAL FOR SUCCESSFUL REFINEMENT RUNS.

Residue-pair restraints derived from the DeepAccNet estogram predictions were crucial for the successful refinement (Figure 4.2D and Figure 4.5A). When only residue-wise and global accuracy predictions (either from DeepAccNet or external EMA tool [12]) were used for the refinement calculations, performance did not statistically differ from our previous work (P-value>0.1). When Bert or MSA inputs were further provided to DeepAccNet (red bars in Figure 4.2D), significant increases in model quality were observed for several targets (Figure 4.5B). Final pool model quality analyses (Figure 4.6) suggest that sampling was improved by those extra inputs (i.e., overall model quality increases). The single model selection was generally reasonable across the three different DeepAccNet variants.

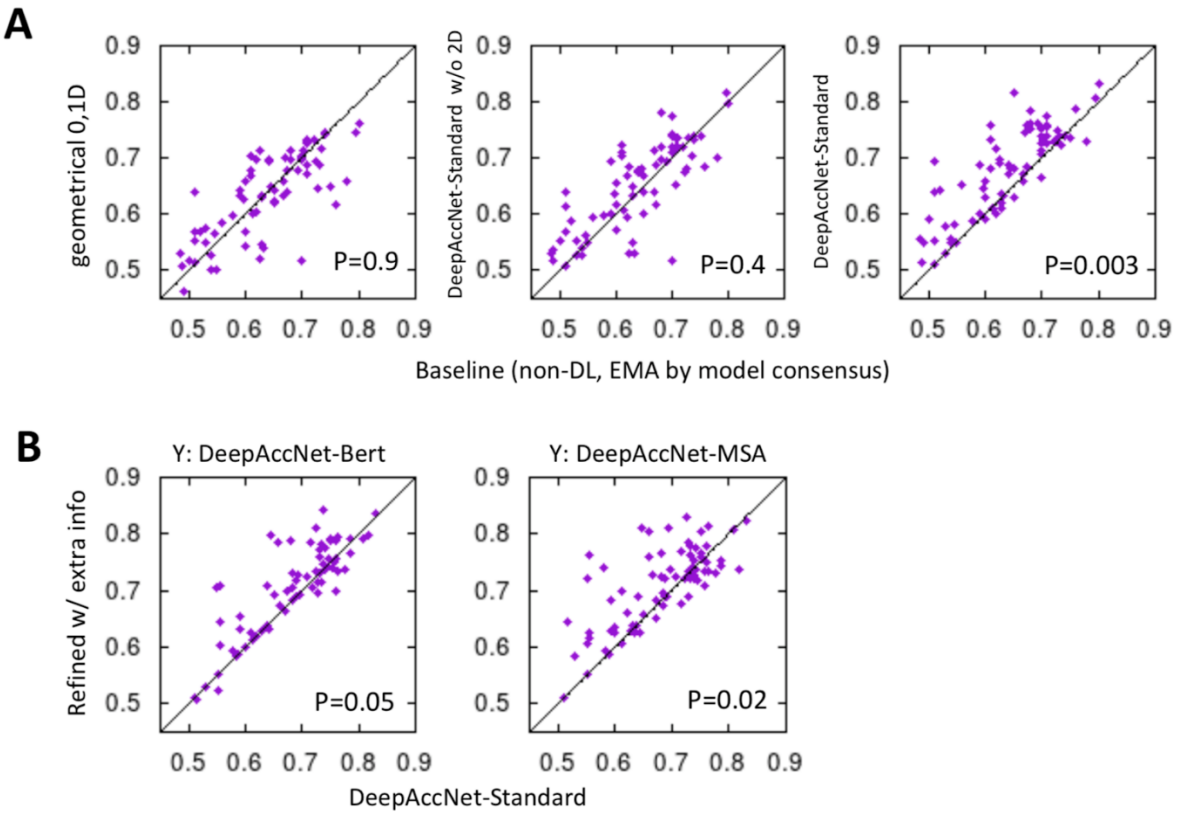

Figure 4.5. Comparison of refinement performances by EMA methods with extra information utilized.

A) Refinement performance with different EMA methods taken during refinement, compared to that of our baseline approach [8], [12] (x-axis) using model consensus for 1D (region detection) and 2D (residue pair confidence) and Rosetta energy for 0D (global ranking). B) Refinement performance gained by providing extra input from Bert and MSA features, compared to DeepAccNet without such extra input features (x-axis)

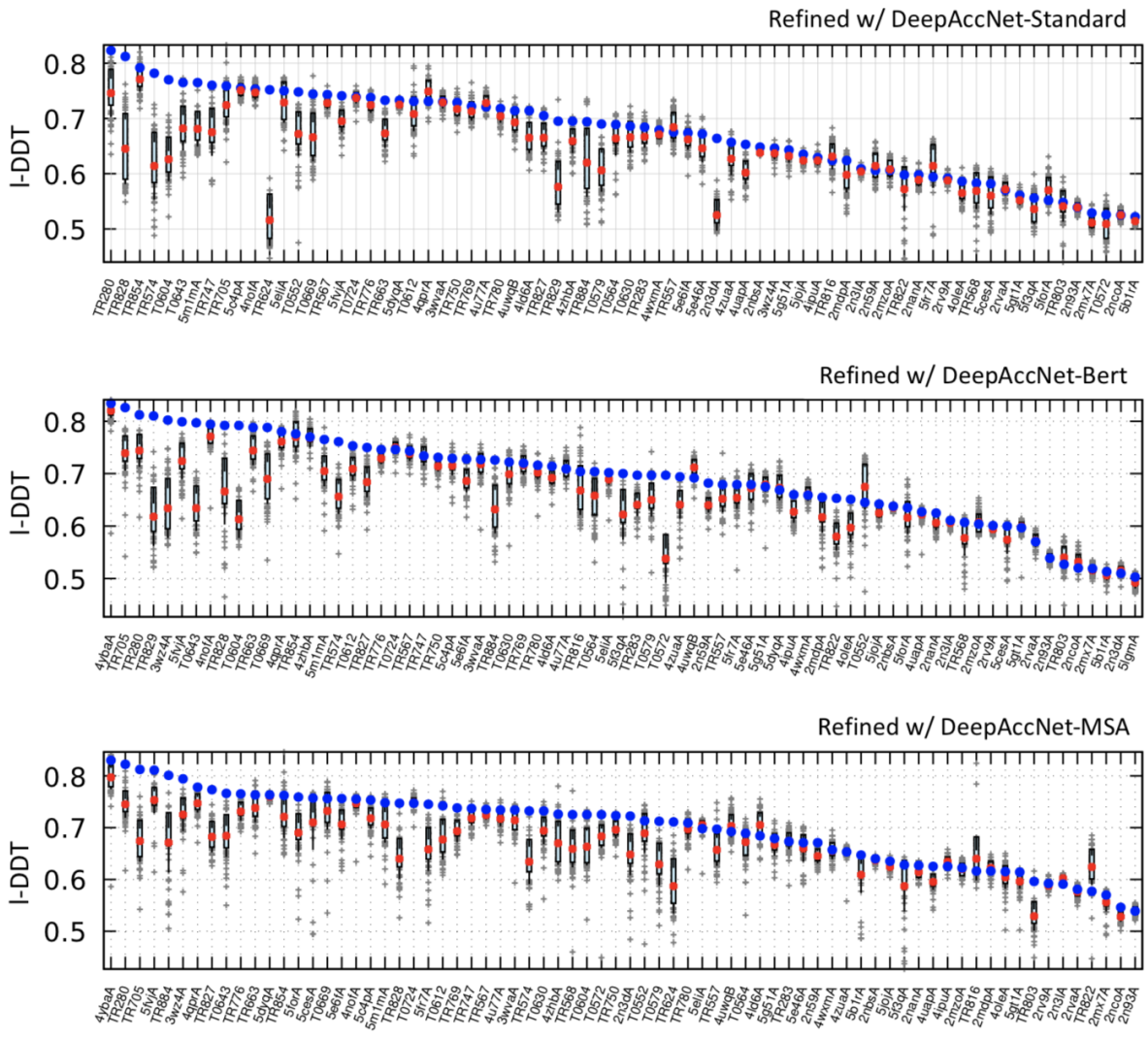

Figure 4.6. The model quality of the final iteration structural pool and the selected one from the refinement runs using DeepAccNet-Standard, -Bert, and -MSA.

Boxplots show the 1st and 3rd quartile of the model qualities in the final iteration models in cyan bars, their mean in red dots, models selected by DeepAccNet (without structural averaging) in blue dots, and individual values for other models in gray crosses.

4.5 SUMMARY

The model accuracy improvements occur across a broad range of protein sizes, starting model qualities, and types of errors. Refinement improved models across various secondary structures to similar extents and corrected secondary structures originally modeled incorrectly, increasing model secondary structure accuracy by almost 10% based on an 8-state definition (Figure 4.4BC) [44]. As shown in Figure 4.2F, improvements involve identifying and modifying erroneous regions when the overall structure is correct (TR776) and overall concerted movements when the core part of the model is somewhat inaccurate (5m1mA). The accuracy prediction network promotes this overall improvement in two ways. First, it provides a more accurate estimation of unreliable distance pairs and regions at every refinement iteration for every model on which sampling can be focused. Second, it provides a means to effectively constrain the search space in the already accurately modeled regions through residue-residue pair restraints. This constraint is essential for the refinement of large proteins. The DeepAccNet networks enable the refinement protocol to adjust how widely to search on a case-by-case basis; this is an advantage over most previous refinement approaches where search has generally been either too conservative or too aggressive [45].

Guiding search using the network predictions improved Rosetta's protein structure refinement over a wide range of protein sizes and starting model qualities (Figure 4.2). However, there is still considerable room for improvement in the combined method. To more effectively use the information in the accuracy predictions, it is necessary to explore sampling strategies that can better utilize the network predictions and more frequent communication between Rosetta modeling and the accuracy prediction network. Currently, the network is fast enough to evaluate the accuracy of many models more frequently, but improving prediction

speed would undoubtedly be beneficial. Also, we find that DeepAccNet often overestimates the quality of models when the network heavily optimizes those through our refinement protocol (Figure 4.4A); adversarial training could help reduce this problem and allow more extensive refinement [46]. There is also considerably more to explore in integrating deep learning methods to guide refinement. For example, we investigate the selection of which of the current sampling operators to use in each situation and the development of new sampling operators using generative models such as sampling missing regions by inpainting. More generally, reinforcement learning approaches should help identify more sophisticated iterative search strategies. Finally, our methods entered CASP14 and performed well, for example the accuracy prediction guided refinement method was the only refinement method at CASP14 able to consistently improve targets greater than 200 amino acids [47].

Chapter 5. FULL-ATOM RESOLUTION WITH GRAPH NEURAL NETWORKS (PLUTO).

5.1 INTRODUCTION

In many fields of computer science, 2D ResNet architectures are known as one of the most robust processors for input features organized in 2D space [4], [48]. Although the underlying inductive bias that pairs of neighboring pixels have more correlated signals than pairs further apart is perfectly accurate for traditional image samples with pixels, this does not always hold true for protein structures represented in distance maps; entries far apart in distance maps often exhibit strongly correlated changes due to their actual proximity in 3D space.

The previous versions of the DeepAccNet methods used a variant of ResNet architectures with 2D convolutions [16]. As a result, they learned reusable kernels over 2-mer interactions among residues, achieving state-of-the-art performance. One practical and effective approach that we used for the DeepAccNet methods was to add 3D convolution layers to scan through local 3D atomic coordinates to capture higher degree interactions [22]. These 3D convolution layers scanned 24x24x24Å volume surrounding each C_α atom. They allowed the network to scan the side-chain rotamers of each residue and its interactions with other residues close in 3D space. The fact that this 3D convolution module significantly improved the performance of the DeepAccNet methods suggests that there may be additional advantages to be gained by more naturally interacting with protein structures in 3D space (Figure 3.4).

Proteins can be represented as point clouds, and some network architectures are more suitable for processing point clouds in 3-dimensional space. In particular, classes of networks called graph convolutional networks [49] and message passing neural networks [50] represent an input point cloud as a graph. They convolve node (i.e., residues/atoms) and edge (i.e.,

residue/atom interactions) signals over a given graph structure. These approaches are often invariant to rotations of input 3D coordinates by construction, suitable for machine learning on point clouds. Several EMA approaches with protein graph representations have been proposed; ProteinGCN by Sanyal et al. represents each atom as a node and makes per residue and global accuracy estimates [51]. GraphQA by Baldassarre et al. represents each residue as a node and relies on MSA co-evolutionary to make per-residue accuracy estimates [52]. However, unlike the DeepAccNet methods, these graph-based EMA networks were not one of the top performers in CASP14 [35]. Nevertheless, they showed promising performance approaching that of the other state-of-the-art EMA methods.

This chapter explores the extension of DeepAccNet architectures using graph-based neural networks. In particular, we use SE3-transformer networks, self-attentive networks that are, by design, invariant and equivariant to 3D rotation and translation in 3D point clouds and graphs [53]. This invariance property is essential for EMA tasks since rotation and translation to input protein structures should not change their outcomes. We explored both residue-based graph (referred to as **RBG**) networks and atom-based graph (referred to as **ABG**) networks. We show that the resulting architecture, named **Pluto**, can make more accurate predictions than its DeepAccNet counterpart.

5.2 DEVELOPING ACCURACY PREDICTORS WITH RESIDUE-BASED GRAPH NEURAL NETWORKS

5.2.1 *Defining residue-based protein graphs.*

The most straightforward graph representation for a protein structure is a linear graph, where nodes represent amino acids and bonds represent chemical connectivity in the backbone primary structure space [52]. We add additional edges to this graph representing proximity in 3D space (i.e., 3D contacts) by selecting the top *k* nearest neighbor residues for every residue. The resulting residue-based graph representation G_{simple} is as follows:

$$G_{simple} = (V_{ca}, E_{ca})$$

$$V_{ca} = \{v | v \in S_{ca}\}$$

$$E_{ca} = \{e_{ij} | \ \|pos(v_i) - pos(v_j)\| < th(top_k)\}$$

, where *v* is a C_α atom, which belongs to S_{ca}, a set of C_α atoms of an input protein sequence. An edge e_{ij} is formed if the distance between C_α coordinates of *i*-th and *j*-th residues is shorter than $th(top_k)$, a threshold determined by the *k* nearest neighbor cutoff. We performed a light hyperparameter search and determined that the value of *k* should be 16. This is the largest choice of *k* while we can fit reasonably well-parametrized models.

Although the above definition of graph structures perfectly captures the complete information about backbone geometry, it provides no explicit information about how side-chains are packed. Therefore, we extended this RBG representation by adding representative side-chain atoms (referred to as "**tip atoms**") as extra nodes. The resulting graph, G_{tip}, is the same as G_{simple} except for each residue having 2 nodes; one for its C_α and another for its tip atom. See Table 2.3 for the definition of Tip atoms.

5.2.2 *Defining model architectures and input features for RBG models*

We trained 4 variants that take in residue-based graphs. Here, we briefly describe their architectures and training process. As previously mentioned, all models use the SE3-transformer framework [53].

All models take in either G_{simple} or G_{tip} that are featurized from an input decoy protein structure. The SE3-transformer modules take L0 and L1 features as inputs. Specifically, L0 features are 1) sinusoidal positional encodings of residue index, which are typically used with transformer models [54], 2) a binary indicator for whether the node represents tip atom or not, and 3) a 1-hot encoded amino acid vector. L1 features are displacement vectors from C_α or tip atom to other heavy atoms in the same residue. These displacement vectors provide full-atom information of side-chain packing to the nodes.

The first model ("per-res-C_α model") processes input G_{simple} through 5 layers of SE3 network parametrized as follows: num_degrees=3, num_channels=64, edge_features=4, div=4, and n_heads=4. The output 2D matrix (N by 64, where N is the number of residues) is then tiled to make a 3D matrix of size N by N by 128, similar to the DeepAccNet methods. An *i*,*j*-th vector along the third axis of this matrix represents an edge between *i*-th and *j*-th nodes. 1D convolutions further process this with kernel and stride size of 1 to predict masks and estograms (see Chapter 2.1 for their definitions). Finally, predicted masks and histograms are combined to calculate C_α l-DDT predictions.

The other three models are slight variants of per-res-C_α. The second model ("per-res-tip model") is the same model of per-res-C_α, except that the input is now G_{tip}. Since G_{tip} has twice the number of nodes compared to G_{simple}, it has an extra step of 1D convolution with kernel

and stride size of 2 to shrink a 2*N by 64 output matrix to N by 64. Then, the resulting N by 64 matrix goes through the same 1D convolution operations as per-res-C_α

We further extend per-res-tip by swapping the last 1D convolution steps with 4 ResNet modules to make the third model ("per-res-tip-RESNET model"). Finally, the last model ("per-res-tip-RESNET-sep" model) has two separate SE3-networks for mask and estogram predictions. A visual description of these residue-based graph networks is shown in Figure 5.1, and the parameters are listed in Table 5.7.

We trained these networks and a baseline network (DeepAccNet-standard) with a subset of the DeepAccNet dataset (3005 proteins per epoch). We used the smaller dataset to assess their performance against the baseline quickly. The same DeepAccNet loss was used to train all networks via Adam with a learning rate of 1e-4 [26]:

$$globalLoss = EstogramLoss + 10.0 * lDDTLoss + 0.25 * MaskLoss$$

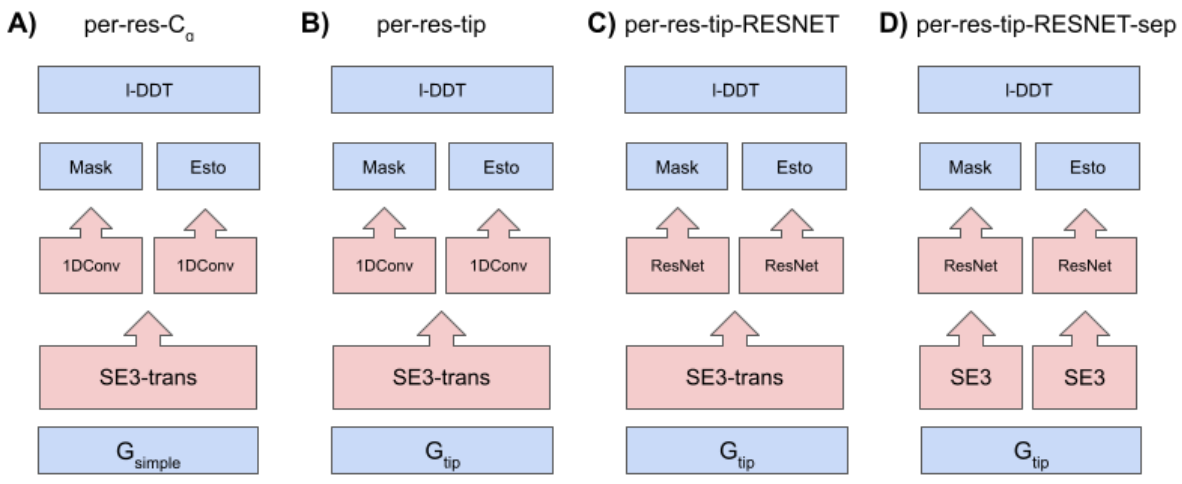

Figure 5.1. The visual representation of the residue-based graph (RBG) models.

Table 5.7. Model architectures details for the RBG models.

Layers groups	**Descriptions**
SE3 transformer network modules	This module is parametrized with num_degrees=3, num_channels=64, edge_features=4, div=4, and n_heads=4.
1D convolution layers aggregator.	The module is parametrized with kernel=2, stride=2, and out_channels=128.
1D convolution	The module is parametrized with kernel=1, stride=1, and out_channels=128.
ResNet module	This module consists of 12 blocks with 3 iterations dilating convolution patterns (8, 4, 2, 1) with the channel sizes of (128 → 64 → 128).
C_α l-DDT calculation layers	C_α l-DDT values are calculated within GPU memory based on predicted estograms and masks.
Loss	(i) Estograms are evaluated with categorical cross-entropy loss. (ii) Masks are evaluated with binary cross-entropy loss. (iii) C_α l-DDT values are evaluated with mean squared loss. Global loss is defined in Section 5.2.2

5.3 RESIDUE-BASED GRAPH MODELS DO NOT OUTPERFORM THE CONVOLUTION-BASED DEEPACCNET METHODS.

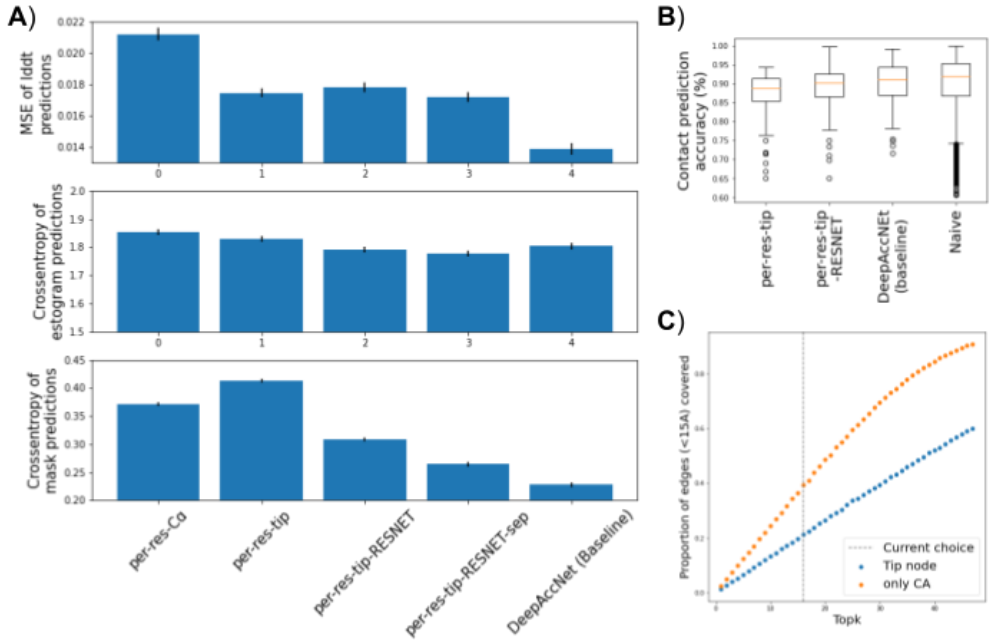

Figure 5.2. Performance of the RBG models.

A) *lDDTLoss*, *MaskLoss*, and *EstogramLoss* of RBG models. *EstogramLoss* and *MaskLoss* are computed with crossentropy, and *lDDTLoss* is computed with mean squared error. Error bars are shown with standard error computed over 500 random validation samples. B) Comparison of contact prediction accuracy with different baselines on high accuracy decoys (global l-DDT > 0.85). The "naive" method simply outputs the contacts in input decoys as contact predictions for their native structures. C) Proportion of edges that are <15Å covered by different top k. k=16 was used to construct all RGB models (dotted line).

Figure 5.2 compares the performance of different RBG models. We use the DeepAccNet-standard method as baseline [16]. All models were trained on the same subset of the DeepAccNet datasets. The accuracy of estogram predictions from the RBG models approach that of DeepAccNet-standard with increasing network complexity (per-res-C_α and per-res-tip) and, in some cases, surpasses DeepAccNet-standard (per-res-tip-RESNET and per-res-tip-RESNET-sep). However, RBG models' predictions on C_α l-DDT values stay significantly worse than l-DDT predictions made with DeepAccNet-standard. This is due to the general lack of accuracy in mask predictions by the RBG models (Figure 5.2A). Panel B shows the accuracy of contact (defined at <15Å, mask) predictions with different models. The naive baseline method outputs the contacts in input decoys as contact predictions for their native structures. This straightforward method performs well because the method is tested on randomly selected 6400 decoys whose global l-DDT scores are above 0.85; these decoys already have good contacts from the start. This simple operation of copying contacts should easily be learnable for all other methods. DeepAccNet-standard performs similarly to the naive method with improved variance in their prediction loss. The distance map features used in DeepAccNet give easy access to distances among all pairs of residues. On the other hand, the RBG models cannot easily learn this because our top k (k=16) forces the models to see only a small fraction of all edges that are shorter than 15Å (Figure 5.2C). For the RGB models to gain direct access to all pairs of residues, it would require a significantly larger choice of k or fully connected graphs over 300 residues at maximum. GPUs would not fit such densely connected large graphs. Another problem is that this RBG representation still does not fully interact with atom point clouds in the most natural way because side-chain atoms are not represented as nodes. They are still just a class of L1 features.

5.4 DEVELOPING ACCURACY PREDICTORS WITH ATOM-BASED GRAPH NEURAL NETWORKS

This section explores the possibility of developing frameworks that define graphs with a node per heavy atom of protein structures. We call them atom-based graph (**ABG**) models.

5.4.1 *Defining atom-based protein graphs.*

The major problem when we define a protein structure as a graph with a node for each heavy atom is GPU memory. While this was later alleviated by Nvidia's implementation of SE3 transformer networks, the original implementation by Fuchs et al was not able to fit large protein structures with reasonably high choice of k for top k neighbor connections [53].

We tackled this problem by two approaches: 1) by only constructing a graph around the residue of interest and making predictions only on those atoms that belong to the residue (Figure 5.3A), and 2) by constructing a graph on a whole protein structure with sparse connections motivated by underlying chemistry (Figure 5.3B).

The graph input for the first approach is defined as follows;

$$G = (V, E)$$

$$V = \{v | \left\| pos(v) - pos(v_{ref}) \right\| < 16\text{Å}\}$$

$$E = \{e_{ij} | \left\| pos(v_i) - pos(v_j) \right\| < th(top_k)\}$$

,where v_{ref} is a C_α atom of the residue of interest, and v is any heavy atoms that are within 16Å of v_{ref}. Then, edges are formed if the two vertices are top k nearest neighbors of each other in the same way as described in Chapter 5.2.1.

We formulated the EMA task as a simple regression problem to each heavy atom's ground truth l-DDT values. We used the mean squared error between true and predicted full-atom l-

DDT values. The predictions were only made for heavy atoms of the residue of interest to prevent atoms located at the boundary of the subgraph defined by the 16Å ball from significantly affecting the prediction.

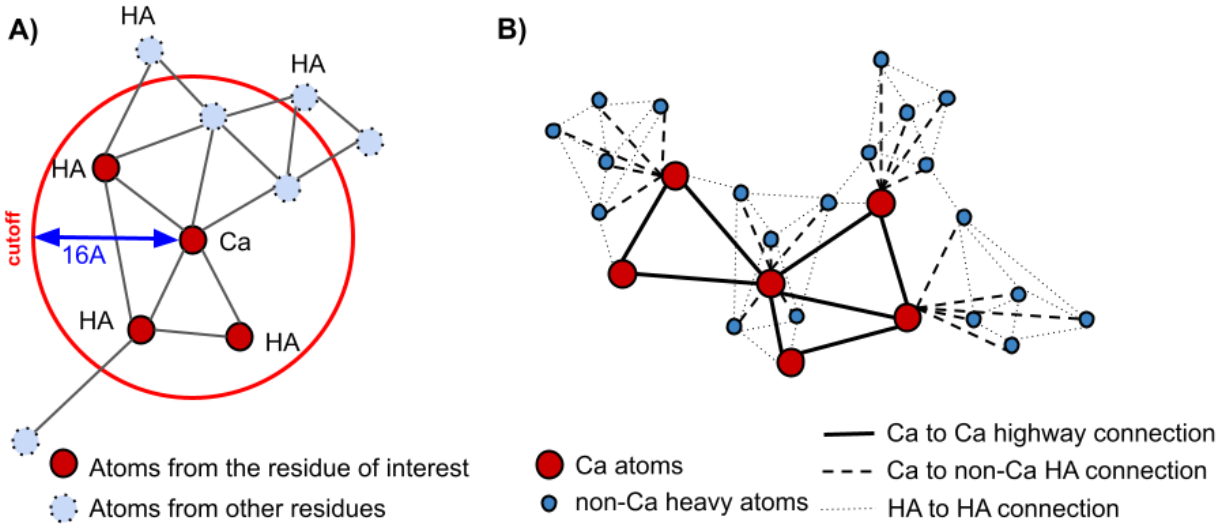

Figure 5.3. The visual representation of the atom-based graph (ABG) model
(A) A graph defined around a single residue, (B) the Pluto architecture.

Table 5.8. Choice of top k and associated validation MSE performances of the ABG models defined around a single residue.

Top k	**k=1**	**k=4**	**k=8**	**k=16**	**k=20**
MSE	0.0356	0.0321	0.0303	0.292	0.294

We picked the SE3-transformer model with the best validation performance by conducting a hyperparameter search through a) the number of neighbors (top k, Table 5.8), b) batch size, and c) learning rates and scheduling (data not shown). We settled with k=16, batch size of 64, and linear decay scheduling (decay=0.999 per epoch) with starting learning rate of 1e-3.

Although this approach has fine-grained access to local atom information, it can only see information within the 16Å ball. Hence it lacks access to the global structure information. Furthermore, even if we increase the radius of this 16Å ball, without unbelievably large choice of k, the network won't be able to reach atoms that are far away with only 5 layers of SE3-networks. The receptive field of these graph nerual networks is too small with 5 layers of graph convolutions. We overcome these shortcomings with the second approach defined in the next Chapter.

5.4.2 *Defining model architectures and input features for ABG models.*

We constructed a graph on a whole protein structure with sparse connections motivated by underlying chemistry for the second approach. Specifically, this approach still has a node defined per heavy atom and have nearest neighbors connected over them. We simply connect a smaller number of nearest neighbors. The difference is that we add sparse skip connections between representative atoms (in this case, C_α) of residues. These "highway" connections make it possible for any atom to reach any other atoms in the structure with a significantly reduced number of hops while maintaining local dense webs of side-chain heavy atoms.

We named this approach **Pluto** and G_{Pluto} is constructed as follows.

$$G_{Pluto} = (V_{Pluto}, E_{Pluto})$$

$$V_{Pluto} = S_{ha} \cup S_{ca}$$

$$E_{Pluto} = \{e_{ij} | \ \|pos(v_i) - pos(v_j)\| < th(top_P), v_i \in S_{ca}, v_j \in S_{ca}, \}$$
$$\cup \ \{e_{ij} | \ \|pos(v_i) - pos(v_j)\| < th(top_Q), v_i \in S_{ha}, v_j \in S_{ha}, \}$$
$$\cup \{e_{ij} | \ \|pos(v_i) - pos(v_j)\| < th(top_8), v_i \in S_{ca}, v_j \in S_{ha}, \}$$

, where S_{ca} is a set of C_α atoms and S_{ha} is a set of heavy atoms that are not C_α. Three types of edges are formed. The C_α to C_α edges provide long-range global information (thresholded with top P). The C_α to heavy atom edges provide the backbone to sidechain information (thresholded with top 8). Finally, the edges among the heavy atoms provide within sidechain information (thresholded with top Q). This formulation allows us to keep connections among heavy atoms relatively sparse, while maintaining the ability to reach any atom in the structure with relative ease. See the visual representation of this in Figure 5.3.

We again formulated this as a simple regression problem per node; we used the mean squared error between true and predicted full-atom l-DDT values. We trained SE3-transformer models by conducting a hyperparameter search for the choice of top P and top Q. We used the subset of the DeepAccNet dataset described in Section 5.2.2 for this hyperparameter search. An Adam optimizer with a learning rate of 1e-5 was used. The hyperparameter search showed that P=32 and Q=12 archive the best validation performance with a relatively light GPU memory load (Table 5.9). Based on this observation, we decided to train a final **Pluto** model, where we parameterized the network with num_layers=8, num_degrees=2, num_channels=48, topP=32, and topQ=12. Finally, this model was trained on the full

DeepAccNet dataset with an Adam optimizer [26] with a learning rate of 1e-5. The model was trained across 8 A100 GPUs.

We later extended the dataset with the predictions from RoseTTaFold [6] to train on decoy structures sampled by state-of-the-art transformer models. See Chapter 2.3 for how these decoys were generated.

Table 5.9. Choice of top P and top Q and associated validation MSE performances of PLUTO.

n_layer	n_degree	n_channel	topP	topQ	MSE
6	3	32	16	6	0.148
6	3	32	16	8	0.146
6	3	32	16	12	0.142
6	3	32	16	16	0.138
6	3	32	32	6	0.146
6	3	32	32	8	0.142
6	3	32	32	12	0.138
6	3	32	32	16	0.139

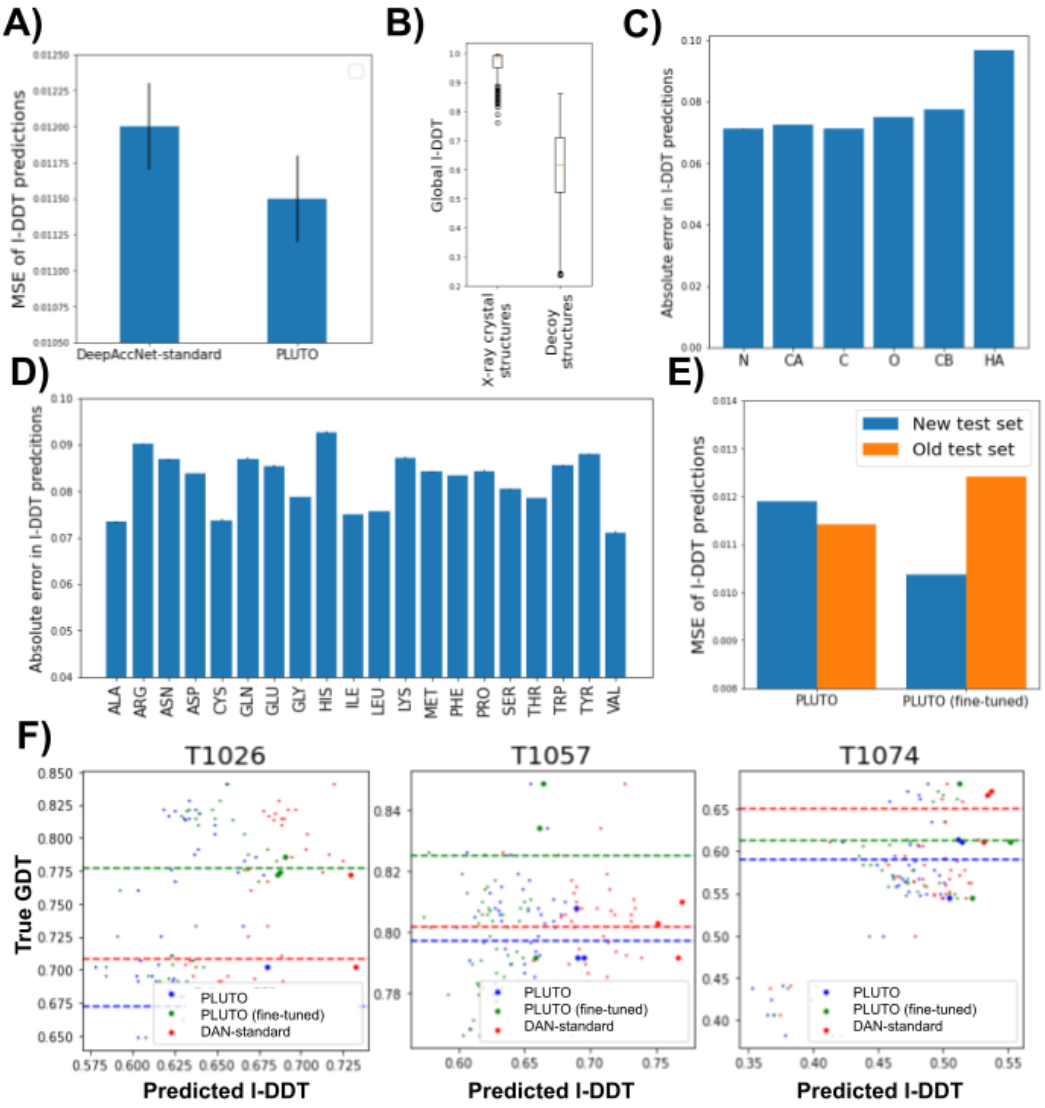

Figure 5.4. Performance of the Pluto models.

A) Performance of Pluto compared to that of DeepAccNet-Standard [16] on the held-out test set. Standard errors are shown as error bars. B) Global l-DDT values predicted by Pluto on experimentally resolved crystal structures vs. decoy structures. CD) Absolute error in full-atom l-DDT predictions grouped by atom types (C) and amino acids (D). Any heavy atoms that are not N, CA, C, O, nor CB are labeled "HA". E) Performance of Pluto and its retuned version on held-out test sets. F) Performance comparison of Pluto, the returned version of Pluto, and DeepAccNet-standard on randomly selected CASP14 targets (T1026, T1057, and T1074). Each dotted line shows the average true GDT [55] values of three decoys that are predicted to be most accurate by the respective methods.

5.5 ATOM-BASED GRAPH MODELS (PLUTO) OUTPERFORM DEEPACCNET

For this final comparison, both Pluto and DeepAccNet-Standard were trained on the full DeepAccNet dataset. The Pluto model exhibited superior performance on the unseen test set regarding the mean squared errors of predicted full-atom l-DDT values (P-value <0.05 with Student's t-test, Figure 5.4A). Similar to DeepAccNet, Pluto successfully recognized native structures and assigned median global l-DDT predictions of 0.976 (Figure 5.4B). No correlation between the b-factor of crystal structures and prediction MSE was observed (data not shown). Pluto performed worse predictions on side-chain heavy atoms than on backbone atoms (Figure 5.4C). The bulky side chains were harder to predict accurately (Figure 5.4D). As seen in Table 5.8, the ABG model defined around a single residue was significantly worse than Pluto (MSE-Loss of 0.0292 for the single-residue ABG vs. 0.011 for Pluto), highlighting the importance of access to global structure information.

We also fine-tuned Pluto to state-of-the-art structure prediction samples from RoseTTAFold. From CASP14 onward, we expect more transformer-based structure predictors to enter the scene. The fine-tuning is necessary to keep Pluto up to date. Figure 5.4E shows the performance of Pluto and its fine-tuned variant to the old DeepAccNet and new extended dataset. While the fine-tuned variant performs significantly better on the newly extended test set, its performance decreases on the old DeepAccNet set. This change in performance shows that the traditional structure predictors and transformer-based methods have different failure modes. The returned Pluto model is simply geared more towards the newer dataset. Figure 5.4F shows the performance of fine-tuned PLUTO (green line) on randomly selected three CASP14 targets. The retuned PLUTO is better at selecting CASP14 decoys than its untuned counterpart.

While we successfully trained a network that outperforms DeepAccNet by representing protein structures in graphs, there are a couple of caveats to the Pluto framework. First, although Pluto is statistically significantly better than DeepAccNet in predicting structural accuracy, the improvement of 0.0005 MSE may not be practically so substantial. Second, due to the nature of EMA methods, Pluto's predictions are always biased to the decoy generation processes to some degree. For Pluto to stay up-to-date, constant retraining with decoys generated by state-of-the-art structure prediction methods is required. The code and datasets used to train the Pluto network are available upon request.

Chapter 6. FILTERING PROTEIN DESIGNS WITH DEEPACCNET AND PLUTO

6.1 INTRODUCTION

After the successful integration of the accuracy prediction networks to the model refinement framework (Chapter 4), we sought another opportunity in protein design, in particular, for small protein binders.

Generally, protein design starts with searching for a set of amino acid sequences that fold roughly into a desired shape via computational modeling [56]. Then, several filters are applied to the modeled structures to find the sequences that exhibit favorable properties. In the case of designing small protein binders to a target protein, the structure of the target protein typically stays constant. Here, the task is to generate sequences for the small binders so that they fold into 3D structures that form the desired interface. The designed sequences are then experimentally validated typically by removing unstructured proteins by protease assays and measuring enrichment for binding activities (Kd) by FACS analysis [57].

There are several hypotheses for the failure modes of the binder design process. One of them is that designed binders are not simply folding into structures that we expect them to fold into. Another possibility is that traditional filtering metrics for interface quality may fail to capture certain aspects of what makes interfaces good for binding. We hypothesize that deep learning models trained on millions of protein geometry may capture these otherwise missed good designs. This chapter briefly explores the use of structural accuracy predictors, such as DeepAccNet [16] or Pluto, as a filter for removing unsuccessful binders.

6.2 OBTAINING PROTEIN-BINDING PROTEIN DATASETS

The designs of protein-binding proteins were obtained from Coventry et al. [56] This set includes binders made for the following target proteins: epidermal growth factor receptor (EGFR), interleukin-7 receptor alpha (IL-7Rα, IL-7Rα-graft), Interleukin 6 Receptor (IL-6R), the SARS-CoV-2 coronavirus spike protein (SARS_CoV2_RBD), and covid_rbd_at3 (Covid receptor binding domain). The first four targets are human cell surface or extracellular proteins, and the last two are pathogen surface proteins. For each target, unsuccessful binders were subsampled by Nate Bennett. We are not disclosing the full dataset at this time because the work is not yet published. The Kd threshold of 10000 was used to decide if there was a significant binding signal or not. The resulting number of samples and successful binders are shown in Table 6.10. Note that, for IL-6R, SARS_CoV2_RBD, and covid_rbd_at3, their scaffolds were already filtered to have good DeepAccNet-standard accuracy scores as it was part of their pipeline.

Table 6.10. The number of samples and successful binders in the dataset.

	EGFR	IL-7Rα	IL-7Rα-graft	IL-6R	SARS_CoV2_RBD	covid_rbd_at3
# samples	272	304	22784	14240	272	1047
# binders	17	19	1424	890	17	151

6.3 MAKING PREDICTIONS WITH THE EMA METHODS.

We used the DeepAccNet-standard network to make three different types of accuracy predictions: complex l-DDT, binder l-DDT, interface l-DDT. The first two metrics are straightforward. DeepAccNet predictions were made on the whole complexes and binders alone, and l-DDT scores were calculated by combining estogram and mask predictions. Because both estogram and mask predictions are defined over pairs of atoms, we can calculate l-DDT only on a subset of edges between binder and target proteins. Recall that the l-DDT score for a residue is calculated over pairs of residues whose distance in the native structure is less than 15Å. When we calculate l-DDT only over edges across an interface, no pair may fall under this 15Å threshold. We do not calculate any interface l-DDT for such residue. Predictions with Pluto were made on the whole complex, including both binders and target proteins, without any special treatment.

6.4 DEEPACCNET AND PLUTO CAPTURE IDENTIFY SUCCESSFUL BINDERS AS WELL AS ROSETTA.

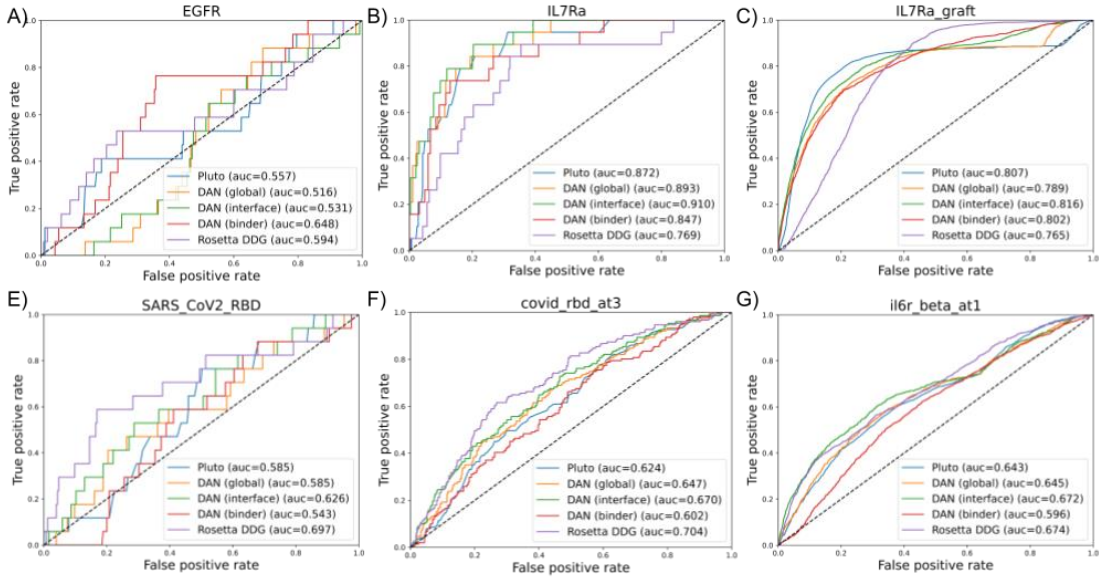

Figure 6.1. Performance of DeepAccNet, Pluto on selecting successful binders.
The receiver-operator curves (ROC) for selecting successful binders from the pool of designs are shown. Rosetta's ddG scores (Rosetta's estimates of the binding energy of a complex) are used as a baseline.

Figure 6.1 showcases the performance of DeepAccNet and l-DDT scores calculated on subsets of designed protein-protein complexes. DeepAccNet metrics outperform Rosetta's ddG scores in IL7Ra and IL7Ra-graft datasets. Rosetta ddG, on the other hand, slightly outperforms DeepAccNet in all other datasets. It is important to note that DeepAccNet was only trained on monomeric protein structures and their decoys. Even then, DeepAccNet's predictions were able to generalize to protein-protein interfaces, suggesting that DeepAccNet captures generalizable properties about hydrophobic protein cores and interfaces. Potential improvement may be achieved if we train an accuracy model on protein-protein interfaces directly.

Chapter 7. GENERAL-PURPOSE ACCURACY ESTIMATOR FOR NON-PROTEIN MOLECULES (GAAP).

7.1 INTRODUCTION

The advent of Alphafold2 [5] and RoseTTaFold [6] in recent years has drastically shifted the research landscape of protein machine learning. These two methods accurately predict 3D coordinates of protein structures at full-atom resolution with the average C_α l-DDT around 0.80~0.90. Although they still struggle in some cases, such as loop modeling, the powerful and flexible use of coevolutionary information from multiple sequence alignment (**MSA**) has made it possible to make structural predictions that are accurate enough to be used as putative ground truth. In fact, the predictions from these models are so accurate that they could be used as a reference native structure for calculating l-DDT values, and the calculated l-DDT values outperform all top-performing solutions in CASP14. In addition, along with their structural predictions, both Alphafold2 and RoseTTaFold make their own accuracy predictions on C_α l-DDT and other metrics (e.g., RMSD and TMScore [55]) after their MSA-processing transformer blocks. These accuracy predictions are shown to be generally more accurate than any existing EMA method. Although one may argue that the EMA methods that do not rely on the MSA information provide orthogonal information to the predictions of AlphaFold2 and RoseTTAfold with heavy reliance on MSA information, the gap in their prediction accuracy is quite significant.

One apparent shortcoming of the structural prediction and accompanying accuracy estimation of AlphaFold2 and RoseTTAfold is that they do not generalize to chemistry outside of proteins. Apart from the obvious reason that they are only trained on protein monomeric structures, these methods internally operate at the resolution of residues, not atoms. This makes

it hard to directly apply them to datasets such as Protein-DNA complexes, small molecules, and RNAs; some parts of these molecules cannot simply be deduced down to amino acid residues. On the other hand, Peptides have similar chemistry to protein monomeric structures. However, they often contain noncanonical or N-methylated residues; these residues cannot be handled by the frameworks expecting 20 canonical amino acids as input. Finally, classes of peptides called minicycles and macrocycles have cyclic backbones with short chains [58]. These small chains are often regulated by similar but slightly different rules of chemistry compared to monomeric protein structures.

We develop a deep-learning framework that can predict structural accuracy scores for non-protein chemical structures. There are several notable previous efforts in this area. For instance, Townshend et al. learned a neural network invariant to input 3D coordinates to predict the structural accuracy of RNA conformations [59]. Although the network was learned on only 18 experimentally determined structures with 1000 decoys each, the network exhibited a state of the art performance on several different metrics for RNA structure assessment. Aimnet by Zubatyuk is another deep learning framework that can predict energy values with state-of-the-art performance on small molecules [60]. However, the framework is limited to relatively small compounds due to its speed not being scalable for designing millions of molecules and screening them rapidly.

Our framework, GAAP (Generalized Atom Accuracy Predictor), is an SE3-transformer neural network whose parameters are learned on several million chemical structures, including 1) protein monomers, 2) macrocyclic peptides, 3) protein-DNA complexes, and 4) protein-ligand complexes. The framework abstracts the notion of amino acids residues and operates directly on atomic coordinates. It learns to predict the structural accuracy of molecules by

assessing the quality of pairs of atoms. This single GAAP network primarily works well for the macrocycle and protein-DNA with state-of-the-art accuracy. Furthermore, this framework can process 10 macrocycle peptide structures in under a second, making it feasible to apply the framework to a large-scale screening of designed structures.

7.2 PREPARING PROTEIN AND NON-PROTEIN DATASETS.

Before starting this section, I would like to acknowledge Hahnbeom Park, Frank Dimaio, Ryan McHugh, Robert Pecoraro, Gaurav Bhardwaj, Stephen Rettie, and Guangfeng Zhou for helping me with preparing decoy structures for various non-protein datasets.

7.2.1 *Protein monomeric structures*

We used protein monomeric structures collected for DeepAccNet [16] and Pluto (see Sections 2.2 and 2.3). To reiterate, in total 22928 sequence clusters were formed after filtering out with a sequence similarity cutoff of 40%, and decoys were generated with 4 different methods; 1) comparative modeling with RosettaCM [1] and selecting lowest-scoring decoys at each GDT-TS bin (ranging from 50 to 90 with bin size 10), 2) perturbing native crystal structures to generate high accuracy structures, 3) folding with trRosetta [3] followed by trRosetta minimization scripts, and 4) folding with RoseTTAfold [6] trunk module with log MSA subsampling followed by trRosetta minimization scripts. See Figure 2.1 for the distributions of these monomeric protein decoy structures in terms of C_α l-DDT scores.

7.2.2 *Peptide macrocycle structures*

Experimentally determined macrocycle structures that are composed of L-amino acids, D-amino acids, and N-methylated amino acids only were collected from the Cambridge Structure

database [61]. Some macrocycle structures were experimentally resolved in-house at the Institute of Protein Design, and they were also added to the mix. Decoy structures were generated by using Rosetta's simple_cycpep_predict application, which uses Generalized kinematic closure (GenKIC [62]) for sampling cyclic backbone conformations from a given sequence and relaxes each cyclic backbone using Rosetta FastRelax. Near-native decoys were also generated by minor perturbations of phi psi angles in Rosetta, followed by Rosetta FastRelax [21]. Additionally, complete random sampling was performed by simply randomizing phi and psi angles and letting FastRelax close the structures. Most of these are very high energy, but this method allows for more cis peptide bond heavy decoys to be generated compared to the GenKIC approach. In total, 97 structures with approximately 5000 decoys each were sampled.

7.2.3 *Protein-ligand complex structures*

18,000 protein-ligand complexes were obtained from PDBbind 2018 [63]. This dataset contains protein-ligand complexes, including short peptides, which come with experimentally validated binding affinity data. Approximately 10k complexes were removed whose ligands ii) are too big (>50 heavy atoms), ii) are covalently bonded to the target, or iii) failed to dock with GALingandDock for various reasons. Metal and protonation states were preserved, and partial charges were assigned using MMFF94 [64]. For each target, GALigandDock [65] was performed, and 30 decoys per target were collected.

7.2.4 *Protein-DNA complex structures*

Protein-DNA complex structures were obtained by scanning through PDB to look for a structure with at least 1 protein and 2 DNA chains. These structures were clustered with a

sequence identity of 80% to remove redundant information between training, validation, and test splits. If more than two protein chains were found, the protein chain with the most contact with DNA was taken. This resulted in 333 ground truth structures. Decoy structures were generated by jittering the protein chain by applying rigid body transformations followed by Rosetta FastRelax of the best 20% by soft Rosetta energy. The best decoys by Rosetta energy were selected while enforcing variety by sampling decoys with a wide range of RMSD values. This decoy generation process does not move the DNA chains at all, and only the protein chains are allowed to move. In total, 335 structures with 30 decoys each were sampled.

7.2.5 *Protein-peptide complexes*

Protein-peptide complexes decoys were generated for testing purposes only. These decoys are not currently included in the training set. Three crystal structures for protein-peptide complexes from the Protein Data Bank [18] were chosen: 5XN3, 1MPO, and 5LSO. For each structure, one residue in the macrocycle with stable interaction to the target protein was selected as a stub. Then, generalized kinematic closure [62] followed by Rosetta FastRelax [21] was performed to sample macrocycle conformations around the selected stub in the context of the target protein.

7.3 FEATURIZATION AND ARCHITECTURE

7.3.1 *Representing chemical structures in graphs.*

In the GAAP framework, an input chemical structure is featurized into a graph structure G_{GAAP}

$$G_{GAAP} = (V_{GAAP}, E_{GAAP})$$

$$V_{GAAP} = S_{ha}$$

$$E_{GAAP} = \{e_{ij} | \ \|pos(v_i) - pos(v_j)\| < th(top_k)\}$$

, where S_{ha} is a set of non-hydrogen heavy atoms of chemical structure. Edges are formed if the distance between the coordinates of *i*-th and *j*-th atoms is shorter than the distance threshold defined by the top k selection. Due to the GPU memory limitation, to fit large samples (primarily protein structures and protein-DNA complexes), we use k=16 for the nearest neighbor counts. This choice of k guarantees that we cover any chemical bonds formed within our training/valid/test datasets with appropriate coverage of non-bonded interaction edges. We may consider increasing this choice on k in the future.

We use the SE3-transformer framework. Our goal is to abstract the notion of amino acid residues from the input and generalize beyond protein chemistry. Hence, the input features do not have any 1-hot amino acid encodings like DeepAccNet and Pluto. Rosetta defined 65 atom types were used for L0 node features (Table 7.11). Edge features include chemical bondedness and their bond orders.

Table 7.11. 65 atom types for the L1 feature.

CS, CS1, CS2, CS3, CD, CD1, CD2, CR, CT, CSp, CDp, CRp, CTp, CST, CSQ, HO, HN, HS, Nam, Nam2, Nad, Nad3, Nin, Nim, Ngu1, Ngu2, NG3, NG2, NG21, NG22, NG1, Ohx, Oet, Oal, Oad, Oat, Ofu, Ont, OG2, OG3, OG31, Sth, Ssl, SR, SG2, SG3, SG5, PG3, PG5, Br, I, F, Cl, BrR, IR, FR, ClR, Ca2p, Mg2p, Mn, Fe2p, Fe3p, Zn2p, Co2p, Cu2p, Cd

7.3.2 *Formulating prediction formats.*

Our network provides estimates of the structural accuracy of an input model in two different metrics: a distribution of full-atom l-DDT [17] and distance offset on a pair of atoms.

For the full-atom l-DDT prediction, l-DDT values for all heavy atoms are calculated between the input decoy structure and experimentally resolved ground truth structure. Specifically, to calculate per-atom l-DDT for *i*-th atom A_i, we iterate over all heavy atoms A_j that are within 15Å of A_i in the reference structure and calculate the fractions of distances between A_i, A_j that are conserved. Distance between a pair of atoms is considered conserved if it is within a certain tolerance threshold (0.5, 1, 2, 4 Å). The final l-DDT score is the average of four fractions. For example, if a heavy atom has 0.9 l-DDT, it means that 90% of distances around the atom are conserved. The full-atom l-DDT values are projected to 50 bins, where each bin covers a 0.02 l-DDT range. Global l-DDT is calculated by taking an average of full-atom l-DDT over all heavy atoms.

Additionally, we predict distributions of distance offsets between pairs of atoms similar to the DeepAccNet methods. Specifically, we calculate distance errors along all connected edges by the top k nearest neighbor threshold. These error values are then projected to bins (-5Å to 5Å with 0.5Å intervals, referred to as **estograms**), and the network predicts them as categorical distributions. During the training and evaluation, our network predicts the categorical distribution of both error values, providing both point estimates and the confidence for the estimates. We use the expectation of the categorical distributions for the point estimates. Both predictions are evaluated with categorical cross-entropy errors during the training process.

7.3.3 *Defiing model architectures and input features for GAAP*.

The SE3-transformer part of the network is parameterized as follows: num_layers=4, num_heads=4, channels_div=4, channels=32, and num_degrees=3. The final layer of the SE3-transformer module has 82 channels, where the first 50 channels go directly to the l-DDT prediction, and the last 32 channels go to edge estogram prediction. For predicting an estogram between a pair of atoms, we concatenate the 32 channel vector outputs of two atoms and process it through three feed-forward layers.

The final loss is a one-to-one combination of cross-entropy errors on full-atom l-DDT and estogram predictions. We use an Adam [26] optimizer with a learning rate of 1e-4 with linear decay of 0.999 per epoch. Each epoch consists of a full pass through all sequences in our datasets. In other words, at least 1 decoy structure for each experimentally resolved structure is seen by the network per epoch. Note that protein-DNA datasets and macrocycle datasets are under-represented in terms of the number of experimentally resolved structures. Hence, we oversample them during the training by a factor of 40 and 5, respectively.

Decoys are sampled uniformly in terms of their true global l-DDT values. Precisely, we determine the minimum and maximum global l-DDT of the decoys for each target structure and uniformly draw a random float number in between. Then, we pick the closest decoy to the drawn float number. We do this because simple uniform sampling of decoys results in minimization focused around the mean l-DDT (e.g., 0.74 for macrocycle test sets). Performing well on mediocre structures is somewhat pointless because what we practically care about is being able to accurately separate high-accuracy decoys and crystal structures from other less accurate decoys. All training runs were terminated at epoch 120, where reasonable convergence is observed.

7.3.4 *Predicting structure accuracy with GAAP*

Since the GAAP framework makes predictions per edge between neighboring atoms, users can select a subset of edges and focus on a particular part of structures. This especially comes in handy for making prediction of complexes. For example, in binder design, target protein structures are typically static and do not undergo structural changes. In such cases, one may just extract edges across the interface and within binders to assess the quality of target-binder complexes.

To summarize accuracy predictions along all edges and assign a single accuracy metric per structure, we calculated the average probability of conservation, *P(conserved)*, as follows:

$$P(conserved) = \frac{1}{4n}\sum_{e\in E}[P\left(|d_{true} - d_{decoy}| < 0.5\right) + P\left(|d_{true} - d_{decoy}| < 1.0\right) + P\left(|d_{true} - d_{decoy}| < 1.5\right) + P\left(|d_{true} - d_{decoy}| < 2.0\right)]$$

, where n is the number of edges, d_{decoy} is the distance of edge e in the input decoy structure, and d_{true} is the distance of edge e in its ground truth crystal structure. $P\left(|d_{true} - d_{decoy}| < x\right)$ can be obtained by summing the center mass of estograms predicted by the GAAP models. We refer to the true value of *P(conserved)* as "observed conservation."

7.4 RESULTS

7.4.1 *GAAP captures the accuracy of unseen macrocycle decoys.*

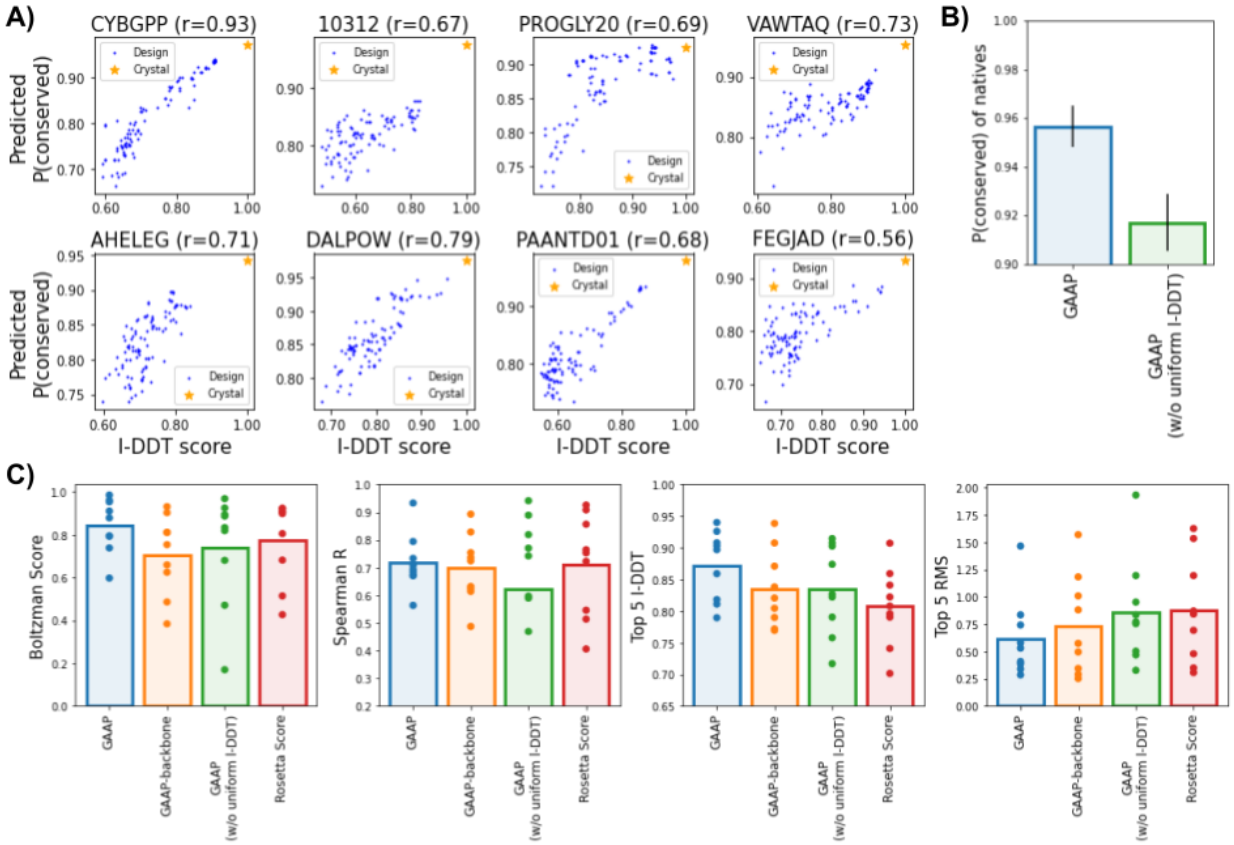

Figure 7.1. Performance of the GAAP models on unseen peptide macrocycles.
A) Scatter plots showing the correlation between predicted and observed average probability of conservation, *P(conserved)*, on unseen macrocycle datasets. The blue dots are decoy structures, and the orange stars are crystal structures. Spearman-r values are shown in parenthesis. All of them have p-value < 0.001. B) Average probability of conservation for native structures predicted by GAAP and GAAP trained without uniform decoy sampling in terms of l-DDT. Standard errors are shown in the error bars. C) Bar plot showing the average performance of GAAP, GAAP on poly-ala backbones, GAAP trained without uniform decoy sampling in terms of l-DDT, Rosetta energy score. All metrics are calculated for each native structure and its associated decoys presented in dots. Boltzmann score and Spearman-R values of GAAP variants are measured against true l-DDT, and RMS is used for Rosetta energy score.

The performance of the GAAP model on unseen macrocycle peptide structures is summarized in Figure 7.1. For all experimentally resolved crystal structures and their decoys, the predicted probability of conservation shows a clear funnel towards the crystal structures. In most cases, the crystal structures are selected as the best structures with scores higher than 0.90, indicating that the GAAP model can recognize energy minima. For PROGLY20, the crystal structure was not predicted to be the best structure, but it still ranks in the top 5 percentile (Figure 7.1A). Sampling decoys uniformly with respect to their global l-DDT values helps the GAAP model recognize the crystal structures with a higher predicted probability of conservation. This sampling method stops the network from focusing too much on decoys with average l-DDT scores (Figure 7.1B).

The GAAP model performs well on the variety of metrics used to evaluate the energy funnel of decoy structures. Figure 7.1C compares the performance of the GAAP model, the GAAP model predicting on backbones presented with poly-alanine sequences, the GAAP model trained without the uniform l-DDT decoy sampling, and the Rosetta energy function. The GAAP model shows superior performance to others for the Boltzmann score metric. For Spearman-R, the GAAP model performs as strongly as the Rosetta energy function. Note that the worst-case scenario for the GAAP model is better than that of the Rosetta energy function. We also selected the top 5 decoys for each native structure using each method and evaluated their true quality in terms of global l-DDT and RMS. Again, the GAAP model showed the best performance. Please note that we did not include the crystal structures in this analysis. This analysis was performed only among decoy structures. Hence, the GAAP models did not take any advantage of crystal artifacts or biases in the crystal structures that make it easy for the methods to tell them apart. Finally, it is particularly interesting that GAAP predictions on poly-

alanine backbones correlated with the global l-DDT of sidechain packed structures with their actual sequences. This suggests that GAAP may be used as a filter for peptide backbone design.

Finally, we applied the GAAP model on complexes that consist of macrocyclic peptides and proteins. Because the predictions are made per edges, we can focus on a subset of edges that corresponds to parts of interest (e.g., interface). While the GAAP model can assign high accuracy for decoys with high true accuracy (observed conservation), the network fails to predict low accuracy for decoys that are, in fact, less accurate. The GAAP predictions can still be used as a filter for selecting peptide binder design with enriched rate to find better structures (Figure 7.2)

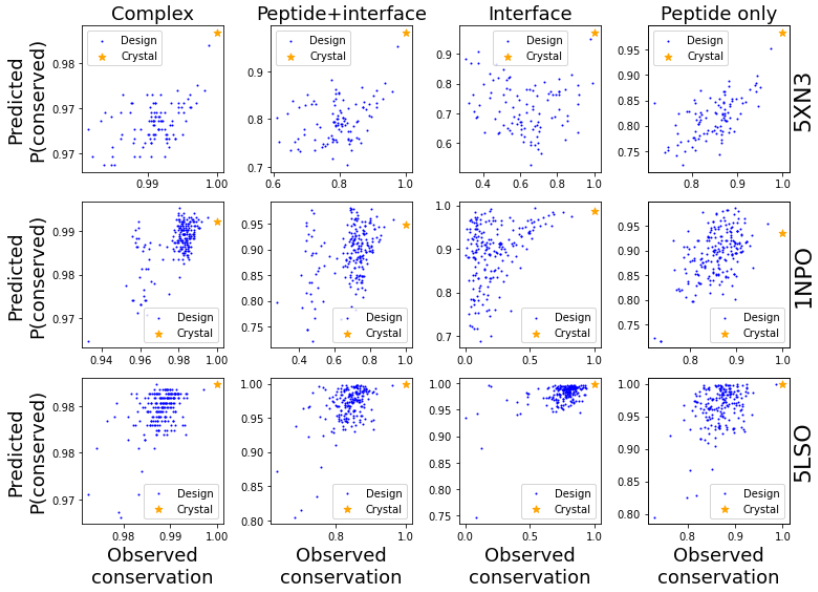

Figure 7.2. Performance of the GAAP models on peptide (macrocycle)-protein complexes. Scatter plots showing predicted P(conserved) and observed conservation for three different macrocycle-protein datasets (5XN3, 1MPO, 5LSO for each row). Each column shows P(conserved) averaged over a different set of edges: whole complex (1st column), peptide and interface (2nd column), interface (3rd column), and peptide only (4th column). Decoys are shown in blue dots, and experimentally resolved crystal structures are shown in orange dots.

7.4.2 *GAAP captures the accuracy of unseen protein-DNA complexes.*

We next assessed the performance of the GAAP model on unseen decoys and crystal structures of Protein-DNA complexes. Figure 7.3 shows GAAP can recognize crystal structures (orange) apart from their associated decoy structures (blue). We were concerned that GAAP may recognize some easily recognizable features in crystal structures. Hence, we added cartesian minimized crystal structures (cyan) and showed that GAAP still ranks them above the decoys structures (blue). The gap between the crystal structures (orange) and their cartesian minimized versions (cyan) suggests that our datasets' ground truth crystal structures may need to be all cartesian minimized in the future. Such a process ensures that native crystal structures are more concordant with Rosetta's force field and that it is harder for the EMA methods to tell them apart from their decoys by taking advantage of biases within crystal structures.

Figure 7.4 shows that decoys selected by the GAAP model outperform Rosetta's ddG in terms of RMSD and observed fraction of edges conserved. Table 7.12 shows that our training strategy, where we train the GAAP model with decoys sampled uniformly in terms of global l-DDT, helps us recognize both crystal and cartesian minimized crystal structures apart from their structures. This observation is consistent with the result we observed for macrocycle peptides. While the results on unseen test decoys are encouraging, we still have to test this method in more practical design settings and carefully monitor if the predictions positively impact the success rate of DNA-binding protein design.

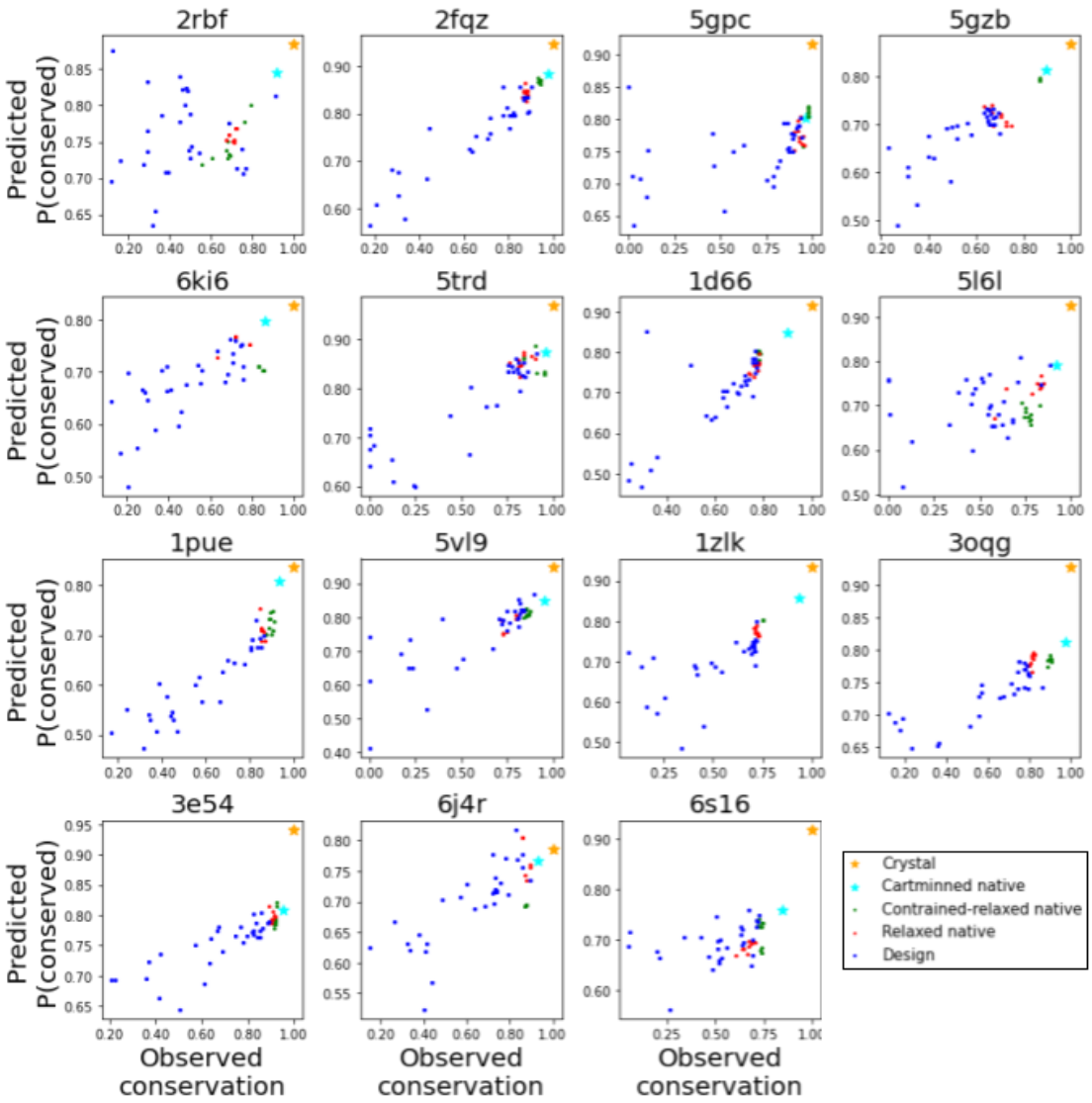

Figure 7.3. Performance of the GAAP models on protein-DNA complexes.

Scatter plots showing the correlation between the predicted and observed probability of conservation across interface on unseen protein-DNA complexes. Each dot represents either decoy or native crystal structures (orange; crystal structures, cyan; Rosetta minimized crystal structures with cartesian restraints, green; Rosetta relaxed crystal structures with 3D constraints; red; Rosetta relaxed crystal structures, blue: decoy (design) structures). The x-axis shows observed conservation, and the y-axis shows the associated prediction of P(conserved)

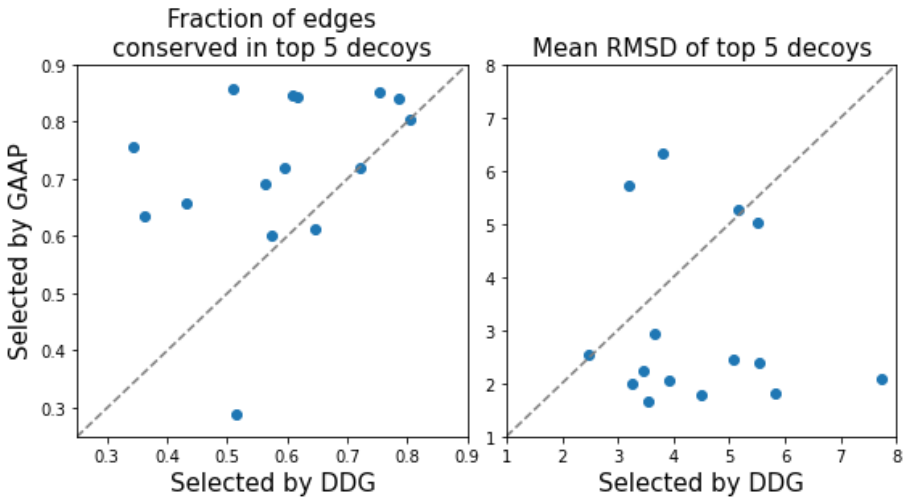

Figure 7.4. Performance of the GAAP models compared to Rosetta DDG.
The left panel shows the average fraction of edges conserved among the top 5 decoys selected by GAAP (y-axis) and Rosetta's ddG (x-axis). Each dot represents a set of decoys associated with the same crystal structure. The right panel shows the mean RMSD among the top 5 decoys.

Table 7.12. Fractions of crystal structures and cartesian minimized crystal structures ranked within the top 5 among their decoy structures.

	Crystal in top 5	Cartmin crystal in top 5
GAAP	100% (15 /15)	100% (15/15)
GAAP w/o uniform l-DDT sampling	93% (14/15)	60% (9/15)

7.5 SUMMARY

Understanding the rules that determine the structures of non-protein molecules and their interactions with proteins is an essential task in computational biology and protein chemistry. Although state-of-the-art methods such as AlphaFold2 [5] and RoseTTAFold [6] can predict protein structures with impressive accuracy, they do not generalize to chemistry outside of proteins. In this work, we successfully trained a graph neural network that can generalize to different types of chemistry outside of protein monomeric structures. This is possible because our GAAP framework abstracts the notion of amino acid residues and defines a graph over atoms instead. In addition, the GAAP framework evaluates interactions between a pair of atoms, making it possible for us to focus on specific subsets (e.g., interface) of an input structure at prediction time. The network can generate approximately 8 predictions per second on macrocycles on CPUs, which enables the screening of millions of structures for peptide design.

Although the network shows promising performance in predicting the structural accuracy of peptides and protein-DNA complexes, there is still considerable room for improvement. First, we would like to extend our training datasets to incorporate ligand crystal lattice structures. These datasets include interactions that are not often present in protein and peptide datasets, and we hypothesize that they may improve the generalizability of the network. Second, rather than predicting the accuracy of input structures, it may be more beneficial to predict the structures themselves explicitly. This is because sampling decoy structures without any bias is a challenging task. There are many methods to generate decoy structures, and our framework is always somewhat biased towards which methods were used to generate training samples. This is not so problematic if we know how test samples at real and practical

applications are generated. Alternatively, direct structure prediction may be a suitable option if we like to achieve true generalization regardless of sampling methods.

Chapter 8. CONCLUSION

In this series of work, we showcased the successful development (Chapter 2,3,5) and application (Chapter 4, 6) of deep learning models that estimate the accuracy of computationally modeled protein and non-protein structures (Chapter 7).

Many other methods for estimating the accuracy of computational modeled protein structures have previously been described [10]–[12], [36]. However, these projects often do not articulate their methods' use other than selecting geometrically more accurate structures out of the pool of candidate pool structures. We showed that our method, DeepAccNet [16], has state-of-the-art performance and is one of the top competitors in the CASP14 EMA category [35]. The method can be applied to practical downstream tasks, such as protein refinement and design. We believe that there are more opportunities for the EMA methods for protein monomeric structures. For example, the protein structure prediction task can be formulated as a reinforcement learning problem, the EMA methods may be used as a critic in the actor-critic setup [66]. We may also use the EMA methods as a discriminator for generative adversarial network [67] setup in the future.

The current state-of-the-art structure prediction methods (e.g., AlphaFold2 [5] and RoseTTAFold [6]) exhibit incredible performance for predicting the structures of protein monomers and the associated accuracy predictions. However, they also face some challenges. These methods are often reliant on coevolutionary information from multiple sequence

alignments. They perform incredibly smart ensembling of evolutionarily related protein structures. However, it is yet unclear if they truly understand the underlying physiochemical rules that govern the process of protein folding. EMA methods trained without MSA information (e.g., DeepAccNet-Standard, Pluto) may capture orthogonal information to MSA-reliant structure prediction methods. Our GAAP model exhibited in Chapter 7 takes another important step in this direction by abstracting the unit of amino acid residues and learning directly on atom clouds. These atom cloud datasets consist of protein structures and a wide variety of non-protein structures generally untouched by protein structure predictors. Although we do not yet have experimental validation data, our methods show that the accuracy of non-protein structures can be predicted on unseen test decoy sets. Saliency analysis on these networks may shed light on how and if deep learning models understand the underlying physiochemical rules in protein folding.

Lastly, one of the shortcomings of EMA methods is that, to some degree, they are always biased towards methods that generate training datasets. In our line of work, we attempted to alleviate this problem by generating decoys with several different methods and including datasets with different types of chemistry. Continuous re-training of the EMA methods is necessary to stay up-to-date for decoy structures generated by state-of-the-art methods for protein structure prediction and non-protein structure samplers.

BIBLIOGRAPHY

[1] Y. Song *et al.*, "High-resolution comparative modeling with RosettaCM," *Structure*, vol. 21, no. 10, pp. 1735–1742, 2013, doi: 10.1016/j.str.2013.08.005.

[2] J. Xu, "Distance-based protein folding powered by deep learning," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 116, no. 34, pp. 16856–16865, 2019, doi: 10.1073/pnas.1821309116.

[3] J. Yang, I. Anishchenko, H. Park, Z. Peng, S. Ovchinnikov, and D. Baker, "Improved protein structure prediction using predicted interresidue orientations," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 117, no. 3, pp. 1496–1503, 2020, doi: 10.1073/pnas.1914677117.

[4] A. W. Senior *et al.*, "Improved protein structure prediction using potentials from deep learning," *Nature*, vol. 577, no. 7792, pp. 706–710, 2020, doi: 10.1038/s41586-019-1923-7.

[5] J. Jumper *et al.*, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, no. 7873, pp. 583–589, Aug. 2021, doi: 10.1038/s41586-021-03819-2.

[6] M. Baek *et al.*, "Accurate prediction of protein structures and interactions using a three-track neural network," 2021. [Online]. Available: https://predictioncenter.org/casp14/

[7] A. Kryshtafovych, T. Schwede, M. Topf, K. Fidelis, and J. Moult, "Critical assessment of methods of protein structure prediction (CASP)—Round XIII," *Proteins*, vol. 87, 2019, doi: 10.1002/prot.25823.

[8] H. Park, "High-accuracy refinement using Rosetta in CASP13," *Proteins*, vol. 87, 2019, doi: 10.1002/prot.25784.

[9] M. Feig, "Computational protein structure refinement: almost there, yet still so far to go," *WIREs Comput Mol Sci*, vol. 7, no. 3, 2017, doi: 10.1002/wcms.1307.

[10] K. Uziela, D. Menéndez Hurtado, N. Shu, B. Wallner, and A. Elofsson, "ProQ3D: improved model quality assessments using deep learning," *Bioinformatics*, vol. 33, no. 10, pp. 1578–1580, 2017, doi: 10.1093/bioinformatics/btw819.

[11] G. Pagès, B. Charmettant, and S. Grudinin, "Protein model quality assessment using 3D oriented convolutional neural networks," *Bioinformatics*, vol. 35, no. 18, pp. 3313–3319, 2019, doi: 10.1093/bioinformatics/btz122.

[12] K. Olechnovič and Č. Venclovas, "VoroMQA: Assessment of protein structure quality using interatomic contact areas," *Proteins*, vol. 85, no. 6, pp. 1131–1145, 2017, doi: 10.1002/prot.25278.

[13] D. Bhattacharya, "refineD: improved protein structure refinement using machine learning based restrained relaxation," *Bioinformatics*, vol. 35, no. 18, pp. 3320–3328, 2019, doi: 10.1093/bioinformatics/btz101.

[14] L. Heo, C. F. Arbour, and M. Feig, "Driven to near-experimental accuracy by refinement via molecular dynamics simulations," *Proteins*, vol. 87, 2019, doi: 10.1002/prot.25759.

[15] L. Heo and M. Feig, "Experimental accuracy in protein structure refinement via molecular dynamics simulations," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 115, no. 52, pp. 13276–13281, 2018, doi: 10.1073/pnas.1811364115.

[16] N. Hiranuma, H. Park, M. Baek, I. Anishchenko, J. Dauparas, and D. Baker, "Improved protein structure refinement guided by deep learning based accuracy estimation," *Nature Communications*, vol. 12, no. 1, Dec. 2021, doi: 10.1038/s41467-021-21511-x.

[17] V. Mariani, M. Biasini, A. Barbato, and T. Schwede, "lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests," *Bioinformatics*, vol. 29, no. 21, pp. 2722–2728, 2013, doi: 10.1093/bioinformatics/btt473.

[18] H. M. Berman *et al.*, “The Protein Data Bank,” 2000. [Online]. Available: http://www.rcsb.org/pdb/status.html
[19] V. Mariani, F. Kiefer, T. Schmidt, J. Haas, and T. Schwede, “Assessment of template based protein structure predictions in CASP9,” *Proteins*, vol. 79, 2011, doi: 10.1002/prot.23177.
[20] A. Zemla, “LGA: A method for finding 3D similarities in protein structures,” *Nucleic Acids Res.*, vol. 31, 2003, doi: 10.1093/nar/gkg571.
[21] P. Conway, M. D. Tyka, F. DiMaio, D. E. Konerding, and D. Baker, “Relaxation of backbone bond geometry improves protein energy landscape modeling,” *Protein Sci.*, vol. 23, no. 1, pp. 47–55, 2014, doi: 10.1002/pro.2389.
[22] G. Pagès, B. Charmettant, and S. Grudinin, “Protein model quality assessment using 3D oriented convolutional neural networks,” *Bioinformatics*, vol. 35, 2019, doi: 10.1093/bioinformatics/btz122.
[23] S. Henikoff and J. G. Henikoff, “Amino acid substitution matrices from protein blocks,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 89, no. 22, pp. 10915–10919, 1992, doi: 10.1073/pnas.89.22.10915.
[24] J. Meiler, M. Müller, A. Zeidler, and F. Schmäschke, “Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks,” *Journal of Molecular Modeling*, vol. 7, no. 9, pp. 360–369, 2001, doi: 10.1007/s008940100038.
[25] A. Elnaggar *et al.*, “ProtTrans: Towards cracking the language of life’s code through self-supervised Deep learning and high performance computing,” *bioRxiv*, 2020, doi: 10.1101/2020.07.12.199554.
[26] D. P. Kingma and J. Lei Ba, “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION.”
[27] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic Attribution for Deep Networks,” Mar. 2017, [Online]. Available: http://arxiv.org/abs/1703.01365
[28] M. Mirdita, L. von Den Driesch, C. Galiez, M. J. Martin, J. Soding, and M. Steinegger, “Uniclust databases of clustered and deeply annotated protein sequences and alignments,” *Nucleic Acids Research*, vol. 45, no. D1, pp. D170–D176, Jan. 2017, doi: 10.1093/nar/gkw1081.
[29] G. Derevyanko, S. Grudinin, Y. Bengio, and G. Lamoureux, “Deep convolutional networks for quality assessment of protein folds,” *Bioinformatics*, vol. 34, 2018, doi: 10.1093/bioinformatics/bty494.
[30] A. H. A. Maghrabi and L. J. McGuffin, “Estimating the Quality of 3D Protein Models Using the ModFOLD7 Server,” *Methods Mol. Biol.*, vol. 2165, pp. 69–81, 2020, doi: 10.1007/978-1-0716-0708-4_4.
[31] J. Haas *et al.*, “Continuous Automated Model EvaluatiOn (CAMEO) complementing the critical assessment of structure prediction in CASP12,” *Proteins*, vol. 86 Suppl 1, pp. 387–398, 2018, doi: 10.1002/prot.25431.
[32] P. Benkert, S. C. E. Tosatto, and D. Schomburg, “QMEAN: a comprehensive scoring function for model quality assessment,” *Proteins*, vol. 71, 2008, doi: 10.1002/prot.21715.
[33] S. Bittrich, F. Heinke, and D. Labudde, “Central Bringing Excellence in Open Access eQuant: A Web Server for Energy Based Protein Structure Quality Assessment,” 2016. [Online]. Available: https://biosciences.hs-mittweida.de/equant

[34] J. Won, M. Baek, B. Monastyrskyy, A. Kryshtafovych, and C. Seok, “Assessment of protein model structure accuracy estimation in CASP13: challenges in the era of deep learning,” *Proteins*, vol. 87, 2019, doi: 10.1002/prot.25804.
[35] J. Won, S. Kwon, and C. Seok, “Assessment of EMA in CASP14 (Evaluation of Model Accuracy).” Accessed: Feb. 27, 2022. [Online]. Available: https://predictioncenter.org/casp14/doc/presentations/2020_12_03_EMA_Assessment_Seok.pdf
[36] K. Uziela, N. Shu, B. Wallner, and A. Elofsson, “ProQ3: Improved model quality assessments using Rosetta energy terms,” *Scientific Reports*, vol. 6, Oct. 2016, doi: 10.1038/srep33509.
[37] H. Park, S. Ovchinnikov, D. E. Kim, F. DiMaio, and D. Baker, “Protein homology model refinement by large-scale energy optimization,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 115, no. 12, pp. 3054–3059, Mar. 2018, doi: 10.1073/pnas.1719115115.
[38] R. J. Read, M. D. Sammito, A. Kryshtafovych, and T. I. Croll, “Evaluation of model refinement in CASP13,” *Proteins*, vol. 87, 2019, doi: 10.1002/prot.25794.
[39] H. Park *et al.*, “Simultaneous Optimization of Biomolecular Energy Functions on Features from Small Molecules and Macromolecules,” *Journal of Chemical Theory and Computation*, vol. 12, no. 12, pp. 6201–6212, 2016, doi: 10.1021/acs.jctc.6b00819.
[40] R. F. Alford *et al.*, “The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design,” *Journal of Chemical Theory and Computation*, vol. 13, no. 6, pp. 3031–3048, 2017, doi: 10.1021/acs.jctc.7b00125.
[41] A. J. McCoy, R. W. Grosse-Kunstleve, P. D. Adams, M. D. Winn, L. C. Storoni, and R. J. Read, “Phaser crystallographic software,” *J. Appl. Crystallogr.*, vol. 40, no. Pt 4, pp. 658–674, 2007, doi: 10.1107/S0021889807021206.
[42] W. Kabsch and C. Sander, “Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features,” *Biopolymers*, vol. 22, no. 12, pp. 2577–2637, 1983, doi: 10.1002/bip.360221211.
[43] Y. Zhang and J. Skolnick, “TM-align: A protein structure alignment algorithm based on the TM-score,” *Nucleic Acids Research*, vol. 33, no. 7, pp. 2302–2309, 2005, doi: 10.1093/nar/gki524.
[44] W. Kabsch and C. Sander, “Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features,” *Biopolymers*, vol. 22, 1983, doi: 10.1002/bip.360221211.
[45] V. Modi and R. L. Dunbrack, “Assessment of refinement of template-based models in CASP11,” *Proteins*, vol. 84, 2016, doi: 10.1002/prot.25048.
[46] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, “Recent Advances in Adversarial Training for Adversarial Robustness,” Feb. 2021, [Online]. Available: http://arxiv.org/abs/2102.01356
[47] D. Rigden, “CASP14 Refinement Assessment.”
[48] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Dec. 2015, [Online]. Available: http://arxiv.org/abs/1512.03385
[49] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” Sep. 2016, [Online]. Available: http://arxiv.org/abs/1609.02907

[50] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural Message Passing for Quantum Chemistry,” Apr. 2017, [Online]. Available: http://arxiv.org/abs/1704.01212
[51] S. Sanyal, I. Anishchenko, A. Dagar, D. Baker, and P. Talukdar, “ProteinGCN: Protein model quality assessment using Graph Convolutional Networks,” *bioRxiv*, 2020, doi: 10.1101/2020.04.06.028266.
[52] F. Baldassarre, D. Menéndez Hurtado, A. Elofsson, and H. Azizpour, “GraphQA: protein model quality assessment using graph convolutional networks,” *Bioinformatics (Oxford, England)*, vol. 37, no. 3, pp. 360–366, Apr. 2021, doi: 10.1093/bioinformatics/btaa714.
[53] F. B. Fuchs, D. E. Worrall, V. Fischer, and M. Welling, “SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks,” Jun. 2020, [Online]. Available: http://arxiv.org/abs/2006.10503
[54] A. Vaswani *et al.*, “Attention Is All You Need,” Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.03762
[55] A. Zemla, “LGA: A method for finding 3D similarities in protein structures,” *Nucleic Acids Res.*, vol. 31, no. 13, pp. 3370–3374, 2003, doi: 10.1093/nar/gkg571.
[56] B. Coventry *et al.*, “Robust de novo design of protein binding proteins from target structural information alone”, doi: 10.1101/2021.09.04.459002.
[57] L. A. Herzenberg, D. Parks, B. Sahaf, O. Perez, M. Roederer, and L. A. Herzenberg, “The History and Future of the Fluorescence Activated Cell Sorter and Flow Cytometry: A View from Stanford,” 2002. [Online]. Available: https://academic.oup.com/clinchem/article/48/10/1819/5642331
[58] T. E. P. Consortium *et al.*, “An integrated encyclopedia of DNA elements in the human genome.,” *Nature*, vol. 489, no. 7414, pp. 57–74, 2012, doi: 10.1038/nature11247.
[59] R. J. L Townshend *et al.*, “Geometric deep learning of RNA structure.” [Online]. Available: https://www.science.org
[60] R. Zubatyuk, J. S. Smith, J. Leszczynski, and O. Isayev, “Accurate and transferable multitask prediction of chemical properties with an atoms-in-molecules neural network,” 2019. [Online]. Available: https://www.science.org
[61] C. R. Groom, I. J. Bruno, M. P. Lightfoot, and S. C. Ward, “The Cambridge structural database,” *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials*, vol. 72, no. 2, pp. 171–179, Apr. 2016, doi: 10.1107/S2052520616003954.
[62] D. J. Mandell, E. A. Coutsias, and T. Kortemme, “Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling,” *Nature Methods*, vol. 6, no. 8. pp. 551–552, 2009. doi: 10.1038/nmeth0809-551.
[63] Z. Liu *et al.*, “PDB-wide collection of binding data: Current status of the PDBbind database,” *Bioinformatics*, vol. 31, no. 3, pp. 405–412, Jul. 2015, doi: 10.1093/bioinformatics/btu626.
[64] T. A. Halgren, “Merck molecular force field. II. MMFF94 van der Waals and electrostatic parameters for intermolecular interactions,” *Journal of Computational Chemistry*, vol. 17, 1996.
[65] H. Park, G. Zhou, M. Baek, D. Baker, and F. Dimaio, “Force Field Optimization Guided by Small Molecule Crystal Lattice Data Enables Consistent Sub-Angstrom Protein-Ligand Docking,” *Journal of Chemical Theory and Computation*, vol. 17, no. 3, pp. 2000–2010, Mar. 2021, doi: 10.1021/acs.jctc.0c01184.

[66] D. Panou and M. Reczko, “DeepFoldit-A Deep Reinforcement Learning Neural Network Folding Proteins.”
[67] I. J. Goodfellow *et al.*, “Generative Adversarial Networks,” Jun. 2014, [Online]. Available: http://arxiv.org/abs/1406.2661

Chapter 9. APPENDIX A

9.1 UNRELIABLE REGION PREDICTION

Accuracy values predicted from the network are used to identify unreliable regions. We noticed that the l-DDT metric prefers helical regions (as local contacts are almost always correct). To fix this systematic bias, we exclude short sequence separation contacts in the contact mask within sequence separation of 11 to get corrected residue l-DDT values. Then these values are smoothed through a 9-residue-window uniform weight kernel. The residues at the lowest accuracy are determined as unreliable regions. Two definitions of regions are made: in static definition, the accuracy threshold is varied until the fraction of unreliable regions lies between 10 to 20% of the entire structure. In dynamic definition, this range is defined as a function of predicted global accuracy (i.e., average residue-wise corrected accuracy): from f_{dyn} to f_{dyn} +10% with f_{dyn} = 20 + 20*(0.55 - Q)/30, where Q refers to predicted global accuracy. f_{dyn} is capped between 20 to 40%. One thousand models were generated for each definition of unreliable regions in the diversification stage. The static definition is applied throughout the iterative stage.

9.2 RESTRAINTS

We classified residue pairs in three confidence levels: high confidence, moderate confidence, and non-preserving. Highly or moderately confident residue pairs stand for those whose distance should be fixed from the reference structure (i.e., starting structure) at different strengths; non-preserving pairs refer to the rest which can freely deviate. Confident pairs are collected if C_β-C_β distance is not greater than 20Å and whose "probability with absolute estimated error ≤ 1Å", shortly P_{cen}, is above a certain threshold (e.g., 0.7). For those pairs, bounded functions are applied

at the coarse-grained modeling stage, and the sum of sigmoid functions at the all-atom modeling stage, minima centering at the original distance d_0 for both cases:

Bounded function:

$$f(d) = \frac{\left(d - (d_0 + tol + s)\right)}{s} + 1 \quad for\ d > d_0 + tol + s$$

$$f(d) = \frac{\left(d - (d_0 + tol)\right)^2}{s} \quad for\ d_0 + tol \le d \le d_0 + tol + s$$

$$f(d) = 0 \ \ for\ \ |d - d_0| < tol$$

$$f(d) = \frac{\left(d - (d_0 - tol)\right)^2}{s} \quad for\ d_0 - tol - s \le d \le d_0 - tol$$

$$f(d) = \frac{\left(d - (d_0 - tol - s)\right)}{s} + 1 \quad for\ d < d_0 - tol - s$$

Sum of sigmoid function:

$$f(d) = w_{fa} * [\frac{-1}{1 + \exp\left(-5.0 * \frac{d - d_0 + tol}{s}\right)} + \frac{1}{1 + \exp\left(-5.0 * \frac{d - d_i - tol}{s}\right)} + 1]$$

, where *s* and *tol* stand for width and tolerance of the functions. Thresholds in P_{cen} values for highly confident pairs, P_{high}, and moderately confident pairs, $P_{moderate}$, are set at 0.8 and 0.7, with (*s*, *tol*) = (1.0,1.0) and (2.0,2.0), respectively, by analyzing the network test results shown in Figure S11. Restraint weight at all-atom stage modeling, w_{fa}, is set as 1.0. We noticed iterative refinement with these empirically determined parameters (w_{fa}, {P_{high}, $P_{moderate}$}) brought too conservative changes. We, therefore, ran another iterative refinement with a more aggressive parameter set (0.2, {0.8,0.9}) and chose the trajectory from whichever sampled a higher predicted global l-DDT. For the non-preserving C_β-C_β pairs whose input distances are shorter than 40Å, error probability profiles (estograms) are converted into distance potentials by subtracting error bins from the original distances d0 and taking log odds to convert probability into energy units. Instead of

applying raw probabilities from the network, corrections are made against background probability collected from the statistics of the network's predictions over 20,000 decoy structures in training set conditioning on sequence separation, original distances d_0, and predicted global model quality. The potential was applied in full form interpolated by spline function at the initial diversification stage and was replaced by a more straightforward functional form in a subsequent iterative process for efficiency:

$$f(d) = (d - 9) + 1 \quad for\ d > 9\text{Å}$$

$$f(d) = (d - 8)^2 \quad for\ 8 \leq d \leq 9\text{Å}$$

$$f(d) = 0 \quad for\ d < 8\text{Å}$$

for those pairs predicted from estogram as contacting within 10Å. Contacts are predicted when $P_{contact} > 0.8$, with $P_{contact}$ = sum(P_i) over i whose $d_0 + e_i < 10$Å and P_i stands for probability in estogram at the i-th bin.

9.3 RECOMBINATION ITERATION

At the recombination iteration, instead of running RosettaCM as the sampling operator, model structures are directly generated by recombining the coordinates from two models according to the predicted residue l-DDT profiles by the network. For a “seed” member, 4 “partners” are identified among the remaining 49 members in the pool that have the most complementarity to the seed in the predicted residue l-DDT profiles. Next, all the members in the pool are recombined individually with their 4 partners, resulting in a total of 200 new structural models. First, for each seed-partner combination, “complementary regions” are identified where the seed is inferior to the partner in terms of predicted l-DDT. Then, coordinates at the regions are substituted for those from the partner. Multiple discontinuous regions are allowed, but the total coverage is restricted to the range between 20 to 50% of total residues. Next, Rosetta FastRelax is run by imposing residue-

pair restraints from estograms taken from either the partner or the seed interpolated into pair potentials (see above). Restraints from the partner are taken if any residue in the pair is included in complementary regions otherwise from the seed for the rest pairs. Recombination iterations are called at every 5 iterations to prevent over-convergence in the pool.

9.4 FINAL MODEL SELECTION

A model with the highest predicted global Cβ l-DDT is selected among 50 final pool members. Then a pool of structures similar to this structure (S-score > 0.8) are collected from the entire iterative refinement trajectory, structurally averaged, and regularized in model geometry by running dual-relax with strong backbone coordinate restraints with a harmonic constant of 10 kcal/mol Å2, which was the identical post-processing procedure in our previous work [8]. The final model refers to this structurally averaged and subsequently regularized structure. Structural averaging adds 1% all-atom l-DDT gain on average.

VITA

Bio:

- Naozumi Hiranuma was born in Tokyo, Japan, on July 7, 1990.

Education:

- Ph.D. in computer science, Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA. Research interest: deep learning, computational biology, protein machine learning. Thesis: Protein Structure Accuracy Prediction with Deep Learning and its Application to Structure Prediction and Design. Chair: Professor David Baker.

- B.A. in computer science and biology, Carleton College, Northfield, Minnesota. Research interest: genetic algorithms, multi-agent AI.

Publications:

- Hiranuma, Naozumi, et al. "Improved protein structure refinement guided by deep-learning-based accuracy estimation." Nature communications 12.1 (2021): 1-11.

- Hiranuma, Naozumi, Scott M. Lundberg, and Su-In Lee. "AIControl: replacing matched control experiments with machine learning improves ChIP-seq peak identification." Nucleic acids research 47.10 (2019): e58-e58.

- Koester, Julie A., et al. "Sexual ancestors generated an obligate asexual and globally dispersed clone within the model diatom species Thalassiosira pseudonana." Scientific reports 8.1 (2018): 1-9.