DEEP LEARNING FOR PROTEIN PROPERTY AND STRUCTURE PREDICTION

by

YUZHI GUO

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
August 2022

To my dear family, for all their endless trust, continuous support, and unconditional love.

ACKNOWLEDGEMENTS

There were many people who helped me during my Ph.D. studying career, and I would like to take this opportunity to thank them.

I would like to thank my supervising professor Dr. Junzhou Huang for constantly motivating and encouraging me, and also for his invaluable advice during the course of my doctoral studies. He held me to the highest of standards but also had the faith that I would be able to achieve them. None of the work in this thesis would have happened without him.

I wish to thank my thesis committee members Dr. Chengkai Li, Dr. Jia Rao, Dr. Dajiang Zhu for their interest in my research and for their valuable suggestions regarding my early proposal and this thesis. It is a privilege for me to have each of them serve in my committees.

My research and coding skill also benefited from my internships in industry. My special thanks go to Dr. Jiaxiang Wu. I have been learning a lot from him through the collaborations. Without the improvement of my general problem solving ability, some of the chapters in this thesis would not have been possible.

I want to thank all my colleagues from the Scalable Modeling and Imaging and Learning Lab (SMILE), the Computer Science and Engineering Department. It is my pleasure to meet such a concentration of creative and nice people here. I am grateful to all with whom I spent my time as a graduate student at UTA.

I especially want to thank my good friends for their help in my work and life. My special thanks go to Xinliang, Zheng, Sheng, and Hehuan.

Finally, my special thanks go to my family. I would like to express my earnest gratitude to my grandmother and my parents for their love and countless sacrifices to give me the best possible education. Without their patience and unreserved support, it would not have been possible to reach this stage in my career.

July, 27, 2022

ABSTRACT

DEEP LEARNING FOR PROTEIN PROPERTY AND STRUCTURE PREDICTION

YUZHI GUO, Ph.D.

The University of Texas at Arlington, 2022

Supervising Professor: Dr. Junzhou Huang

I present my work towards solving the fundamental, challenging, and valuable problem for protein property and structure prediction. Specifically, I focus on solving the problem from three critical aspects: (1) designing powerful deep learning networks for specific protein structure property prediction tasks; (2) proposing general methods that enhancing the protein sequence homologous feature, which is an important input feature of relevant tasks; (3) developing a self-supervised pre-training model for learning structure embeddings from protein tertiary structures. To evaluate the effectiveness of the developed methods, I apply several protein downstream tasks including protein secondary structure, solvent accessibility, backbone dihedral angles, protein structure quality assessment, and protein-protein interaction site prediction.

I accomplish my work step by step. Firstly, I start from the protein secondary structure prediction task, and constantly attempt and design different deep learning networks according to the characteristics of specific prediction tasks to learn the protein data representation. In order to learn the powerful representation of protein data and utilize the characteristics of protein secondary structure, I propose an En-

sembleASP method, which is protein ensemble learning with Atrous Spatial Pyramid networks for secondary structure prediction. Moreover, since the homologous information of some proteins is insufficient, I propose a Bagging method which targets at improving the performance of low-quality data in the prediction task. In addition, in order to further solve the problem of uneven distribution of the homologous information in the data, as well as facilitate scientists and researchers to quickly apply and experiment on existing models, I propose a plug-and-play method, WeightAln, which is developed based on the attention mechanism. WeightAln learns the weight of the homologous feature of a target protein, and applies it in the calculation process to obtain a stronger sequence homologous information of the target protein. Last but not least, in order to help protein structure-related downstream tasks, I propose a pre-training model for learning structure embeddings from protein tertiary structures. The model is optimized with a self-supervised loss function, which only relies on protein structures and does not require any additional supervision.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The three-dimensional structure of proteins is significant in the study of proteins, since the specific shape of a protein determines its function [2]. If the protein's three-dimensional structure is altered due to mutations in the amino acid structure, the protein becomes denatured and may not functions as expected. Proteins are chains of amino acids linked by peptide bonds. However, predicting the three-dimensional structure of proteins from amino acid sequence is a challenging task [3]. As a result, it is necessary to address simpler problems in both the prediction of one-dimensional structural properties, such as secondary structure, solvent accessibility, and backbone dihedral angles prediction. In recent years, deep learning techniques have been widely used in protein structure property prediction tasks, and achieved remarkable results compared with traditional machine learning methods [4, 5, 6]. However, there remains two major challenges that prevent us from achieving better performance in predicting protein structural properties: (1) most of the research efforts focus on looking for a more powerful amino acid sequence encoder, but the relationship among the structure property of proteins (label) is rarely studied; (2) homologous information is the most important input feature of protein prediction task, but the homologous feature of some proteins is low quality or redundant.

In this thesis, I introduce my works towards the effective protein representation learning and protein input feature enhancement based on the related knowledge from proteins and deep learning applications. The whole learning and research process involves many key issues and challenges. Here, I present my works and contributions

in the following three aspects: (1) I investigate the common grounds between protein secondary structure and semantic segmentation problem, then apply and modify image segmentation network to solve the prediction problem of protein secondary structure; (2) I develop an unsupervised framework to enhance the proteins with low-quality homologous feature; (3) to address the redundancy problem in homologous features, I propose a weighted homologous feature method based on the attention mechanism to enhance the homologous features for all target proteins; (4) I propose a self-supervised pre-training model for learning structure embeddings from protein tertiary structures to improve the performance on protein structure-related downstream tasks. Those works together make the thesis meaningful for both applications and methodology perspectives.

All of these works have been published in several research papers [7, 8, 9, 10, 11, 12, 13]. In this dissertation, I will present how those components contribute to the major goal as follows:

Chapter 2 presents the EnsembleASP network for protein secondary structure prediction. I find that in the secondary structure of proteins, usually, adjacent strings of amino acids have the same secondary structure, or that the characteristics of the protein secondary structure determine this well-regulated feature. Therefore, this problem is very similar to the image semantic segmentation [14]. As a result, while developing more efficient amino acids encoders, I also propose an ASP network (Atrous Spatial Pyramid Pooling (ASPP) based network) as the secondary structure generator in our proposed framework. Extensive experiments show that the proposed method can achieve higher performance on protein secondary structure prediction task than existing methods on protein CB513, Casp11 and CASP12 datasets. The method is expected to be useful for protein structure and further protein functions prediction.

In **Chapter 3**, I introduce the Bagging MSA model, which is the first attempt to enhance low quality PSSM (Position-Specific Scoring Matrix, which is a widely used homologous feature in structure property prediction) features of proteins. Through unsupervised learning, our model can generate more informative PSSM features for structure property prediction. Empirical evaluation of CB513, CASP11, and CASP12 datasets indicate the effectiveness of our method.

Inspired by Bagging MSA, **Chapter 4** gives a weighting mechanism to enhance the homologous features for all proteins, not just those with low quality PSSM. Specifically, I propose a novel protein sequence homologous feature weights learning framework, WeightAln, which generates learnable sequence homology weights for protein prediction tasks using attention-based deep learning techniques. Extensive experiments on three protein structure property prediction tasks, secondary structure, solvent accessibility, and backbone dihedral angles prediction, sufficiently demonstrate the effectiveness of our method.

In order to improve the performance of protein structure-related downstream tasks, I propose a self-supervised pre-training model for learning structure embeddings from protein tertiary structures in **Chapter 5**. Native protein structures are perturbed with random noise, and the pre-training model aims at estimating gradients over perturbed 3D structures. I demonstrate the effectiveness of our pre-training model on two downstream tasks, protein structure quality assessment (QA) and protein-protein interaction (PPI) site prediction. Hierarchical structure embeddings are extracted to enhance corresponding prediction models. Extensive experiments indicate that such structure embeddings consistently improve the prediction accuracy for both downstream tasks.

In **Chapter 6**, some brief future work is listed to be done for the completion of this thesis.

CHAPTER 2

PROTEIN ENSEMBLE LEARNING WITH ATROUS SPATIAL PYRAMID NETWORKS FOR SECONDARY STRUCTURE PREDICTION

This chapter investigates the problem of protein secondary structure prediction. A novel Conditionally Parameterized Convolutional network (CondGCNN) is proposed, which utilize the power of both CondConv and GCNN, and we leverage an ensemble encoder to combine the capabilities of both LSTM and CondGCNN to encode protein sequences to obtain better sequential features from proteins. In addition, due to the similarity between the image segmentation problem and the secondary structure prediction problem, I propose an ASP network (Atrous Spatial Pyramid Pooling (ASPP) based network) as the secondary structure generator in our proposed framework. Experimental results show that the proposed method can achieve higher performance than state-of-the-art methods on CB513, CASP11 and CASP12 datasets.

2.1 Introduction

The three-dimensional structure of proteins is significant in the study of proteins since the specific shape of a protein determines its function [2]. If the protein's three-dimensional structure is altered due to mutations in the amino acid structure, the protein becomes denatured and may not function as expected. Proteins are chains of amino acids linked by peptide bonds. However, predicting the three-dimensional structure of proteins from amino acid predictions is a challenging task [3]. Protein secondary structure prediction is an important part of this task [15].

The secondary structure is often evaluated by the Q3 accuracy: three-class classification, that is, two regular secondary structure states: helix (H) and strand (E), and one irregular type: coil (C) [16]. [17] developed a DSSP algorithm to extend the three general states into eight fine-grained states: 3_{10} helix (G), α-helix (H), π-helix (I), β-stand (E), β-bridge (B), β-turn (T), high curvature loop (s), and others (L). Recently, the focus of secondary structure prediction has been on the prediction of 8-state secondary structure (Q8) rather than on the prediction of Q3, since the fact that a chain of 8-state secondary structure contains more structural information for a variety of research and applications [6].

In recent years, deep learning based methods have been widely used in protein secondary structure prediction, and achieved much better results than traditional machine learning methods. For example, Recurrent Neural Network (RNN) based encoder method, which has been proved successful in natural language processing area, is used to predict protein secondary structures [4]; one-dimensional Convolutional Neural Network (1d-CNN) based encoder method has also made some achievements [18]. In addition, some methods combine the superiority of the two networks, such as DeepACLSTM [19]. They use CNN to catch local feature and bidirectional Long Short-term Memory Network (bLSTM) to obtain long-distance dependency information to obtain better amino acid sequence expression, which leads to better predicting

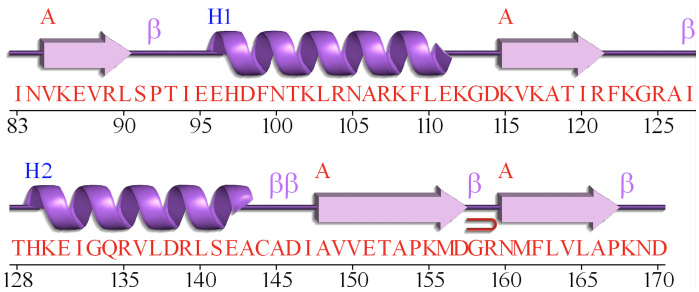


Figure 2.1. Protein 1TIG: Different purple symbols represent different secondary structures, and red characters represent amino acid codes [1].

performance. Meanwhile, if DeepCNN network [18] or ResNet [5] structure is used, CNN-based methods can also obtain long-distance dependency information from the sequence, e.g. CBRNN [6] combines the RNN-based and CNN-based networks.

Although these amino acid sequence encoders based deep learning methods have obtained great success, the relationship between the secondary structures of proteins is rarely studied. DeepCNF [18] method uses Conditional random field (CRF) as the output layer to learn the interdependency among adjacent secondary structure labels. However, the design of this method is not specific for the characteristic of protein secondary structure, and the Q8 accuracy does not improve much. Figure 2.1 shows the secondary structure and the amino acid sequence of protein 1TIG [20] in CB513 dataset, the figure is generated by PDBsum [21]. We find that in the secondary structure of proteins, usually, adjacent strings of amino acids have the same secondary structure, or that the characteristics of the protein secondary structure determine this well-regulated feature. Therefore, this problem is very similar to the image semantic segmentation [14], but with two differences: 1) The fact that the input data for our task is one-dimensional sequences rather than two-dimensional images. 2) For Image Semantic Segmentation, the pooling layer is widely used [22, 23, 14, 24], because the pooling of the adjacent pixels can effectively reduce the size of the input image, which will lead to fewer network parameters, without losing too much image information. However, for the protein sequence, the amino acid information at each position is crucial, so we cannot apply the pooling layer to the amino acid sequence.

Although methods in the field of image semantic segmentation, such as all versions after Deeplab v2 [14, 24], FastFCN [23], and GSCNN [22], have used different encoders, the Atrous Spatial Pyramid Pooling (ASPP) Network Structure [14, 25] followed by the encoder is an important step to identify the edges of objects in the image.

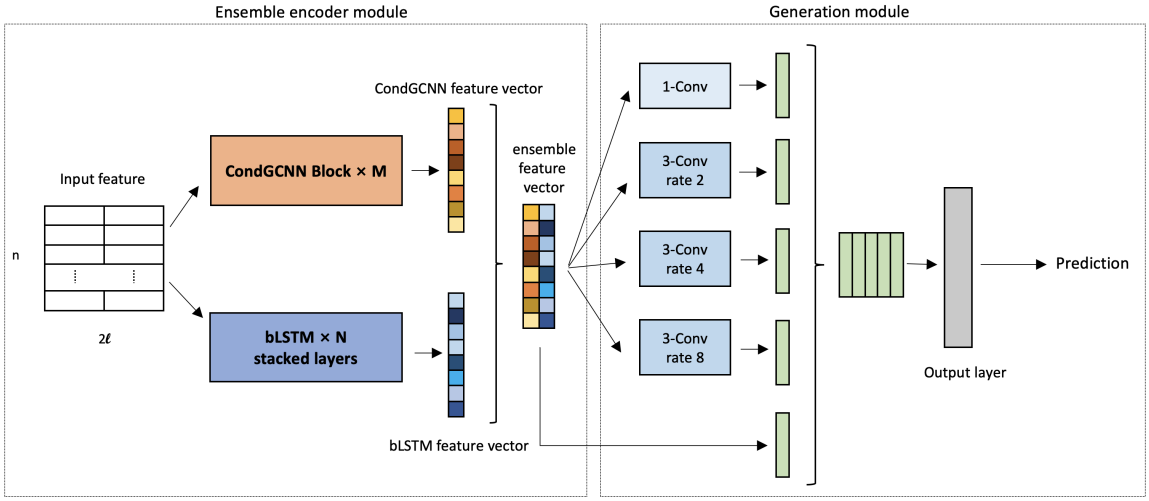


Figure 2.2. Our ensemble learning with ASP networks framework contains ensemble encoder module and generation module. For ensemble encoder, we use several CondGCNN blocks and bLSTM layers in the networks; for generation module, a modified ASPP is applied in the module.

Recently, CNN-based encoding networks have obtained great success on both language and image processing tasks. Gated Convolutional networks (GCNN) [26], which employs a CNN-based gating mechanism at the channel level, helps language modeling to achieve state-of-the-art results. Conditionally Parameterized Convolution (CondConv) [27], which uses extra sample-dependant modules to conditionally adjust the convolutional network, have obtained remarkable improvement in the image processing area. We introduce a novel Conditionally Parameterized Gated Convolutional network (CondGCNN) as a protein sequence encoder, which not only exploits a gating mechanism at the channel level, but also establishes a sample-dependent attention mechanism.

Inspired by the protein secondary structure prediction methods and the image semantic segmentation methods we have mentioned before, we propose a protein ensemble learning with ASP networks method, including an ensemble amino acid sequence encoder and the Atrous Spatial Pyramid Networks. Due to the remarkable

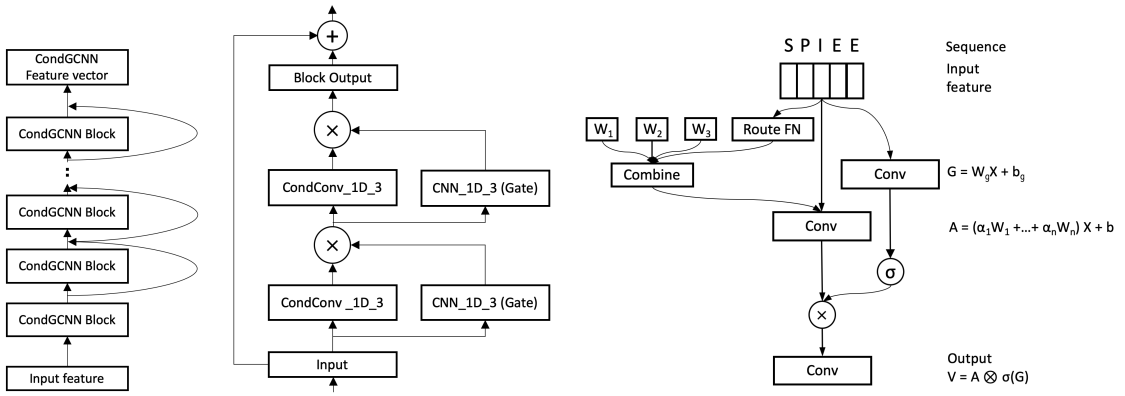
performance of CNN-based method in language model and image processing tasks, and the importance of lstm-based method in protein prediction [19, 4], our amino acid sequence encoder utilizes both CondGCNN model (a new encoder we proposed) and bLSTM model, and adds the ASP Network (optimized ASPP network for our problem) after the encoder.

The technical contributions of our method can be summarized as: 1) It is the first attempt to simulate protein secondary structure prediction as image segmentation tasks, and utilize the power of those models applied in the segmentation area to tackle secondary structure prediction problem, such as ASPP network (optimized as ASP network in our method). 2) We are the first to apply CondConv network on sequence processing problems, and embed it in the GCNN to form a novel amino acid sequence encoder, which equips a gating mechanism at the model channel level and a sample-dependent attention at the input level. 3) We construct an ensemble encoder with cnn-based and lstm-based networks to obtain more diverse information from amino acid sequences.

2.2 Method

2.2.1 Framework Overview

The framework of our method consists of the ensemble encoder module and the generation module. We will introduce each component later in detail. The overall workflow is shown in Figure 4.2. First, the input sequence features are input to the CondGCNN and the LSTM modules, respectively. Then, the two network outputs are concatenated as the feature vectors to feed into the generation module. Finally, the loss is calculated by the output prediction and secondary structure label, and back-propagated to the networks to adjust the parameters.



(a) (b) (c)

Figure 2.3. (a) 32 CondGCNN blocks are used to get the feature vectors of the CondGCNN encoder. (b) Each block contains two layers of Conditionally Parameterized Gated Convolutional network. The input vector of *Input* block is added to the output vector of the *Block Output*, the combination then input to the next block. (c) One layer of CondGCNN contains two parallel convolutional layers, one is the conditionally convolutional layer (A) and the other one is the gated layer (G). The output V is obtained by the element-wise production of A and σ(G).

2.2.2 Ensemble Encoder

The ensemble encoder module is composed of one CondGCNN module and one LSTM module. The CondGCNN module contains M $\times$ Conditionally Parameterized Gated Convolutional blocks, while the LSTM module is constituted by N stacked bLSTM. These two modules generate output feature vectors respectively.

2.2.2.1 CondGCNN module

Figure 2.3(a) and 2.3(b) demonstrate our CondGCNN blocks. We use 32 CondGCNN blocks to get the feature vectors in the CondGCNN encoder. Each CondGCNN block contains two layers of Conditionally Parameterized Gated Convolutional network. We build our CondGCNN layers according to [26, 27], Figure 2.3(c) illustrates the architecture of each CondGCNN layer. A protein sequence is represented by a $n \times 2l$ vector where n is the length of the protein sequence and l represents

the number of amino acid types. The detail about the input features is discussed in section 3.1. For each CondGCNN layer, we set up two CNN_1D_3 networks, one is for gating, and the other one is a one-dimensional Conditionally Parameterized Convolutional network. We calculate the output vector of the CondGCNN layer following:

$$V_h(X) = (X * W_{cond} + b) \otimes \sigma(X * W_g + b_g) \tag{2.1}$$

where, W_{cond}, b are the parameters of the CondConv network, W_g, b_g are the parameters of the gated convolutional network, σ is the Sigmoid function, and $\otimes$ is the element-wise product between vectors. Details of the GCNN network are described in [26]. Specifically, we parameterize the convolutional kernels in CondConv by:

$$W_{cond} = \alpha_1 \cdot W_1 + \alpha_2 \cdot W_2 + \cdots + \alpha_n \cdot W_n \tag{2.2}$$

where each $\alpha_i = r_i(X)$ is an example-dependent scalar weight computed using a routing function with learned parameters, and n is the number of experts. The routing function is able to meaningfully differentiate between the input examples. CondConv [27] computes the example-dependent rounting weights α_i from the layer input in three steps: global average pooling, fully-connected layer, and Sigmoid activation.

$$r(X) = Sigmoid(GlobalAveragePool(X)\, R) \tag{2.3}$$

where R is a matrix of the routing weights mapping the pooled inputs to n expert weights.

Overall, our CondGCNN encoding module utilizes the power of both CondConv and GCNN, which not only provides a gating mechanism at the channel level, but also implements an attention mechanism in a sample-dependant fashion.

2.2.2.2 LSTM module

Some studies about language modeling with the GCNN [26] claim that unlimited contextual information is unnecessary for language models, and GCNN is proved to be able to represent enough contextual information in practice. However, in the area of protein study, several works have proved that capturing the long contextual information (relation from the first atom to the last one) is necessary. Therefore, RNN-based approaches are crucial for protein studies [19, 6]. In this fashion, our proposed method implements two stacked bLSTM layers with hidden size equal to 512 within the LSTM module to capture more long-distance interdependencies of amino-acid residues. A bLSTM neural network consists of two LSTM neural networks in parallel, one of them runs on the input feature and the other one runs on the reverse of the input feature. The corresponding two output vectors are then concatenated as the LSTM module feature vector, see [19] for the stacked bLSTM network in detail.

2.2.3 Generation Module

As shown in Figure 4.2, we feed the concatenated feature vector from Ensemble encoder into the generation module to predict the protein secondary structure. The generation module contains the Atrous Spatial Pyramid Network and the output layer.

2.2.3.1 Atrous Spatial Pyramid Network

As we mentioned before, the secondary structure prediction task for proteins is similar to the semantic segmentation task for images. In image semantic segmentation, the model needs to classify each pixel with one of the predetermined classes. Similarly, in protein secondary structure prediction, we need to classify eight sec-

ondary structures of amino acids for each position. In addition, the labels of protein secondary structure behave consistently for adjacent positions too. Our generation model is inspired by the ASPP network, which is widely used in image segmentation []. In order to reduce the feature map size, ASPP network uses two ways for down-sampling: one is using convolution striding, and the other one is using regular pooling operations. ASPP sets the stride equal to 8 for each convolutional layer in ASPP, and incorporates the image-level features via Global Average Pooling (GAP) [28].

Considering protein sequences are short in length (usually formed by hundreds of amino acids) and each position in the sequence is important, we set the convolution stride to 1 and concatenate the ensemble feature vector with the outputs from four convolutional layers in the networks directly without a pooling layer. Since very high dilation rate is not needed for our scenario, we set the dilation rates = (2, 4, 8). Figure 2.4 demonstrates an example of one layer Atrous Spatial Pyramid network, as shown, the dilation rate of each Atrous (or dilated) convolutional layer is set as 2, the rate of the normal Convolutions is 1.

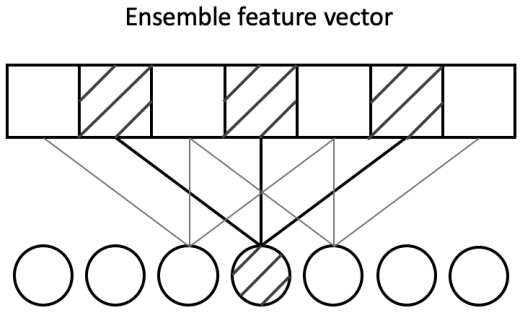


Figure 2.4. An example of one layer of Atrous one-dimensional convolutions with dilation rate equal to 2: A 3x1 kernel with a dilation rate of 2 has the same field of view as a 5x1 kernel, which delivers a wider field of view with same computational cost.

2.2.3.2 Output layer

As shown in Figure 2, after the Atrous Spatial Pyramid network, we feed the result to a 1d convolution with window size 1, to produce the final secondary structure logits. Fully connection layer (FC) is widely used in LSTM-based secondary structure prediction methods. However in our task, since the Atrous Spatial Pyramid networks apply multiplication on the channel of the feature vector, the network would contain too many parameters if a FC layer is implemented as the output layer, which makes the entire model hard to train. Thus, we replace the output layer with a one-dimensional convolutional layer. In section 3, we report the experimental results to prove the effectiveness of this change.

2.3 Experiments

2.3.1 Experiments set up

2.3.1.1 Datasets

We use four publicly available datasets: CullPDB [29] of 5926 proteins, CB513 [15] of 513 proteins, CASP11 of 85 proteins and CASP12 of 40 proteins (`http://predictioncenter.org`). 501 proteins in CullPDB are randomly sampled for validation, then the remaining proteins are used for training. We use CB513, CASP11 and CASP12 as test sets.

2.3.1.2 Input feature

Our input feature consists of two parts: sequence one-hot vectors and position-specific scoring matrix(PSSM). Each amino acid in the protein sequence is represented by a one-hot vector of length 21, which is 20 kinds of amino acids plus one unknown amino acid. PSSM represents the distribution of amino acid types on each position in the protein sequence [8]. Following the same procedure in [15], we get the PSSM

matrix by searching Uniref50 database [30], and concatenate it with the one-hot vectors. As shown in Figure 4.2, the input feature size is $n \times 2l$ where $l = 21$ and n is the length of the protein sequence.

2.3.1.3 Neural network structure and learning Hyper-paramets

In the CondGCNN module, we use 32 Conditionally Parameterized Gated Convolutional blocks. Each block contains two layers of CondGCNN with window size 3 and node size 64, the number of experts is 3. In the LSTM module, we use the two stacked layers bLSTM networks with hidden size equals to 512. In the ASP network, we use three parallel dilated convolutional layers with window size 3, node size 100, dilation rates = (2, 4, 8), and a parallel one-dimensional convolutional layers with window size equals to 1. We use a one-dimensional convolutional layer with window size 1 and node size is equal to 100 as the output layer.

2.3.1.4 Comparison methods

To evaluate our method, we compare it with five state-of-the-art methods: ICML2014, DeepCNF, MUFOLD-SS, CBRNN, and DeepACLSTM. ICML2014 [15] presents a method based on generative stochastic network (GSN) to globally train deep generative model. We use the public dataset they provided, the CullPDB dataset containing 5926 Program database (PDB) files, and CB513 contains 513 proteins. DeepCNF (Deep Convolutional Neural Fields) [18] utilizes the power of CNN and Conditional Random Fields (CRF): several CNN layers are used to extract the sequence feature of proteins, and CRF is used as the output layer to catch the relationship between labels. MUFOLD-SS [31] is a deep inception-inside-Inception (Deep3I) network architecture which extends deep inception networks through nested inception modules. Stacked inception modules could extract non-local residue interactions at

different ranges. CBRNN [6] extract the local context information of protein sequence by two-dimensional convolutional neural networks (2dCNNs), and long-distance information by bidirectional gated recurrent units (bGRUs) or bidirectional long short-term memory (bLSTM). DeepACLSTM [19] using 1dCNN and 2dCNN to extract the discriminative local interactions between amino-acid residues and bLSTM to capture long-range interactions between amino-acid residues.

2.3.2 Ablation study on each component

Table 2.1 shows the prediction results of Conv, CondConv, GCNN, and CondGCNN with different structures on CB513 dataset. First, we compare the results between the regular Convolutional network (Conv) and the Conditionally Parameterized Convolutional network (CondConv) on CB513 dataset to prove the effectiveness of the CondConv. We follow the settings of [18] to build a model with 5 layers of Convolutional networks, then apply the CondConv structure on the regular Convolutional networks directly. However, there is only 0.02 improvement on accuracy. The reason is that when CondConv is applied, that attention mechanism works in a sample-dependant manner, which means the CondConv will use more parameters to focus on distinguishing different samples compared with the traditional convolutional network. This will lead to the overfitting problem. To overcome this disadvantage, we adjust the dropout rate and conduct more experiments with different number of experts. Furthermore, we report the prediction results of GatedCNN (GCNN) with different numbers of blocks on the protein secondary structure prediction task. As observed, the best result is 0.698, which is obtained while using 32 GCNN blocks. Since we do not have quite a large training dataset, the overfitting problem would be severe if the network is too deep. Hence, with large number of blocks applied, the accuracy results of the validation and test set are reduced significantly. Last, we compare

our CondGCNN method with the above CNN-based methods, and the application of CondConv on the basis of GCNN can achieve 0.702 of Q8 accuracy on CB513 dataset.

Table 2.1. Q8 accuracy of CNN, CondConv, GCNN and CondGCNN on cb513 dataset with different structural settings.

Network	Experts num	Blocks num	Dropout rate	Q8 acc
Conv	-	-	0.0	0.678
CondConv	3	-	0.0	0.680
CondConv	3	-	0.2	**0.685**
CondConv	5	-	0.2	0.684
CondConv	8	-	0.2	0.681
GCNN	-	16	0.1	0.696
GCNN	-	32	0.1	**0.698**
GCNN	-	64	0.1	0.677
CondGCNN	3	32	0.2	**0.702**

In order to prove the effect of our Atrous Spatial Pyramid networks (ASP) module, we employ ASP module in LSTM method and DeepACLSTM method. We have noted that when directly insert the ASP module between the encoder and the output layer, the performance is not as expected. The reason is these two methods have used fully connected (FC) layer as the output layer, along with the augmented output of ASP network, leads to overfull parameters. The model is then too hard to train and become overfitting. In such a manner, we replace the output layer with a one-dimensional convolutional layer with a window size 1 and re-run the experiments. The corresponding results are shown in Table 2.2 and Table 2.3. bLSTM-FC represents the original two stacked layers bLSTM networks structure [4], and ACLSTM-FC represents the DeepACLSTM network structure with FC as the output layer [19]. We use bLSTM-ASP-FC and ACLSTM-ASP-FC to indicate that we have inserted our ASP networks between the original encoders (bLSTM and DeepACLSTM) and FC

layer. bLSTM-ASP-Conv1 and ACLSTM-ASP-Conv1 represent that FC layer is replaced by a 1d-cnn with window size 1 as the output layer after the ASP network. The results demonstrate that applying ASP directly to bLSTM and DeepACLSTM networks does not perform well for prediction. Nonetheless, after replacing the output layer with 1d-CNN, we improve the performance of LSTM method by 0.4% and DeepACLSTM method by 0.6%. The results prove that our ASP network can boost the existing state-of-the-art methods of protein secondary structure prediction. In addition, the two tables also show the hidden size (HS) of FC and the node size (NS) of ASP and Conv1.

Table 2.2. The results of before and after inserting the ASP network into the bLSTM network on CB513 datasets.

Network	FC-HS	ASP-NS	Conv1-NS	Q8 acc
bLSTM-FC	128	-	-	0.699
bLSTM-ASP-FC	128	64	-	0.625
bLSTM-ASP-Conv1	-	64	100	**0.703**

Table 2.3. The results of before and after inserting the ASP network into the ACLSTM network on CB513 datasets.

Network	FC-HS	ASP-NS	Conv1-NS	Q8 acc
ACLSTM-FC	128	-	-	0.705
ACLSTM-ASP-FC	128	64	-	0.706
ACLSTM-ASP-Conv1	-	64	100	**0.711**

2.3.3 The results of ensemble learning with ASP

After we conduct a series of experiments to prove the effectiveness of each module, we then combine them to build our network: Ensemble learning with Atrous Spatial Pyramid networks. To confirm the effectiveness of our model, we report the results of CB513, CASP11 and CASP12 datasets to compare with several state-of-the-art methods. As shown in Table 2.4, the 'Ensemble' represents our method without the ASP network, the 'Ensemble-ASP' shows the result of Q8 accuracy after inserting the ASP network. Our method achieve more than 1% accuracy improvement over other state-of-the-art methods on CB513 and CASP11 datasets, and get 0.7% higher on CASP12. The proposed model has not only utilized the predominance of CondGCNN and bLSTM, but also successfully applied the ASP network on protein secondary structure prediction task to obtain significant improvement.

Table 2.4. The comparison between the results of our method and the results of state-of-the-art methods.

Methods	CB513	CASP11	CASP12
ICML2014	0.664	-	-
DeepCNF*	0.683	0.707	0.681
BLSTM*	0.699	0.711	0.681
CBRNN	0.702	-	-
DeepACLSTM*	0.705	0.715	0.678
MUFOLD-SS*	0.704	0.717	0.684
Ensemble (ours)	0.717	0.721	0.686
Ensemble-ASP (ours)	**0.719**	**0.728**	**0.691**

* Data is generated by our experiment.

2.4 Conclusion

In this paper, we propose an ensemble learning encoder with ASP deep learning model (Ensemble-ASP) for protein secondary structure prediction. The framework contains ensemble learning encoder network and ASP network(modified ASPP network). In the ensemble learning network, we use 32-blocks CondGCNN and 2 stacked layers bLSTM networks to encode the rich contextual information. For CondGCNN, we utilize the power of both CondConv and GCNN, which can not only provide a gating mechanism to extract protein information but also implement a attention mechanism at sample-dependant. For ASP network, we use an modified ASPP network for our task to extract the encoder features at an arbitrary context distance. Extensive experiments illustrate that our method exceeds the state-of-the-art methods on 8-state prediction performance. Moreover, our proposed framework of connecting the ASP network after the encoder is considered generalizable, which is not only suitable for protein secondary structure tasks, but also capable for other sequence-labeling models.

CHAPTER 3

BAGGING MSA LEARNING: ENHANCING LOW-QUALITY PSSM WITH DEEP LEARNING FOR ACCURATE PROTEIN STRUCTURE PROPERTY PREDICTION

This chapter introduces a novel pipeline to enhance features for proteins with low-quality homologous features, named "Bagging MSA". The model adopt a convolutional network to capture local context features and bidirectional-LSTM for long-term dependencies, and integrate them under an unsupervised framework. Structure property prediction models are then built upon such enhanced features for more accurate predictions. Empirical evaluation of CB513, CASP11, and CASP12 datasets indicate that the unsupervised enhancing scheme indeed generates more informative features for structure property prediction.

3.1 Introduction

The function of a protein is closely related to its structure, which is largely determined by the amino-acid sequence [7, 32]. However, predicting one protein's structure based on its amino-acid sequence alone remains an open and challenging problem. An alternative approach is to firstly predict structure properties, including secondary structure, solvent accessibility, and backbone dihedral angles [33]. Those predictions are combined eventually to help the final prediction of protein structure.

PSSM (Position-Specific Scoring Matrix) features [34], which reflect per-residue evolution patterns in the sequence profile, are commonly used in the structure property prediction [35, 36]. The quality of PSSM features is basically determined by

the underlying multiple sequence alignments (MSA) [37]. MSA requires searching the query amino-acid sequence through a large-scale sequence database, *e.g.*UniRef [30] and UniClust [38]. The MSA quality of the protein can be evaluated by counting the number of homologous proteins, or the non-redundant sequence homologs(Meff [39]) retrieved from the database. However, for those proteins with a limited number of high-quality homologous sequences, the prediction quality is often limited due to less informative PSSM features [40]. One possible solution is to develop more efficient and accurate MSA search algorithm, such as SABERTOOTH [41], hhblits [42], jackhmmer [43], and HBLAST [44]. These algorithms have achieved certain performance improvement by speeding up the searching process, as well as find more accurate homologous protein sequences in the database. However, if the database did not contain enough homologous protein sequences for the target protein, it is still inaccessible to obtain sufficient quantity or high quality of the MSA, yet the corresponding high-quality PSSM features.

In this paper, we propose an unsupervised deep learning method to enhance the low-quality PSSM features of proteins. To be specific, during the training of our model, we randomly sample the MSA of each protein in a certain proportion in each learning iteration, which we called "Bagging MSA". Then, we use the "Weak PSSMs" calculated by these bags and the "Original PSSM" calculated by all MSA to train our network. In this way, our network can learn how to generate high-quality PSSM from a protein that has low-quality PSSM features.

The most commonly predicted one-dimensional structural property of a protein is the secondary structure. Therefore, in order to evaluate our method on different prediction networks, we use two widely used deep learning techniques in the protein secondary prediction area, which are CNN and bi-LSTM models [18, **?**, 45]. The

knowledge of the secondary structure of proteins and the network of validation of our method are described in section 2 and section 3.

The technical contributions of this paper are summarized as: 1) Our method is the first attempt to enhance low quality PSSMs of proteins. According to the experimental results, our method significantly improve the secondary structure prediction task of proteins with weak PSSM. 2) In the unsupervised module, our method calculate PSSM features by randomly sampling 10% to 20% MSA in each training iteration as the input data, and use the original PSSM features as unsupervised labels. This approach not only increases the diversity of the data, but also make the network more flexible to learn different PSSM quality differences so as to give full play to unsupervised learning. 3) Our method is generalizable since it is capable for any prediction model with PSSM as the input other than just secondary protein prediction task. 4) The unsupervised part of our method is independent, so the output could be used as the input directly for the inference phase of any prediction network, which is more flexible and efficient.

3.2 Related work

3.2.1 Position-Specific Scoring Matrix

3.2.1.1 MSA

A multiple sequence alignment (MSA) is a sequence alignment of multiple homologous protein sequences for the target protein[37]. See Fig. 3.1 for an example of MSA. MSA is an important step in comparative analyses and property predicting of biological sequences, since a lot of information *e.g.*evolution and co-evolution clusters, are displayed on the MSA and can be mapped to the target sequence of choice or on the protein structure [46]. Almost all existing approaches to studying proteins utilize

MSAs indirectly, that is, they convert MSAs into a position-specific scoring matrix (PSSM) that represents the distribution of amino acid types on each column [47].

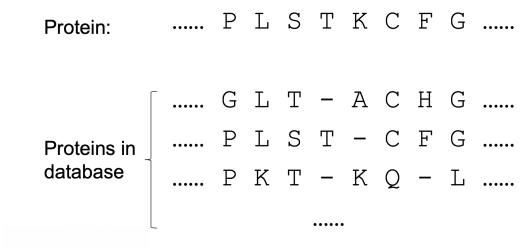


Figure 3.1. An example of MSA.

3.2.1.2 PSSMs calculation

PSSM scores are generally expressed as positive or negative integers. A positive score indicates that the frequency of substitutions in a given amino acid sequence is higher than expected, while a negative score indicates that the frequency of substitutions is lower than expected [48, 49].

We extract the PSSM features of size $n \times 21$ based on Eq.(3.1) and Eq.(3.2), where, n is the protein sequence length, 21 is the sum of twenty known amino acids appeared in the genetic code and one unknown amino acid marker. *Frequency* is the count of occurrences of residue j (j = 1,2,3, ..., 21) in column i (i = 1,2,3, ..., n), 20 represents the known amino acids. A simple procedure called pseudo-counts assigns minimal scores to residues which do not appear at a certain position of the alignment according to the following equation(3.1), where we set the *Pseudocount* equal to 1. N is the number of sequences in the multiple alignments. The *Background frequency* in Eq.(3.2) is the frequency of each residue appearing in the entire MSA of the protein.

$$score_{i,j} = \frac{Frequency + Pseudocount}{N + 20Pseudocount} \tag{3.1}$$

$$PSSM_{i,j} = \log(score/Backgroundfrequency) \tag{3.2}$$

3.2.2 Scoring criteria for PSSM

3.2.2.1 Count score

The number of sequence homologs is recorded as the Count score. As we mentioned before, PSSM is a matrix calculated from the MSA, and the quality of the MSA directly determines the quality of the PSSM. We can use the number of homologous proteins of the MSA to evaluate the quality of the PSSM, which is represented as Count score. The larger Count score leads to more reliable PSSM. Thus, the Count score is one important criteria to evaluate the quality of the PSSM features.

3.2.2.2 Meff score

We introduce the Meff score as the number of non-redundant sequence homologs. As in [40], homologous sequence in MSA of proteins have some redundancy, so we use Meff score as another criteria for PSSM to demonstrates the superiority and stability of our model under various evaluation standards.

The calculation formula of Meff score is shown in Eq.(3.3). where both i and j go over all the sequence homologs, $S_{i,j}$ is a binary number which describes the similarity of two proteins. We use the hamming distance to compute the similarity of two sequence homologs[39]: $S_{i,j}$ is 1 if the normalized hamming distance is less than 0.3; otherwise $S_{i,j}$ is set to 0.

$$Meff = \sum_i \frac{1}{\sum_j S_{i,j}} \tag{3.3}$$

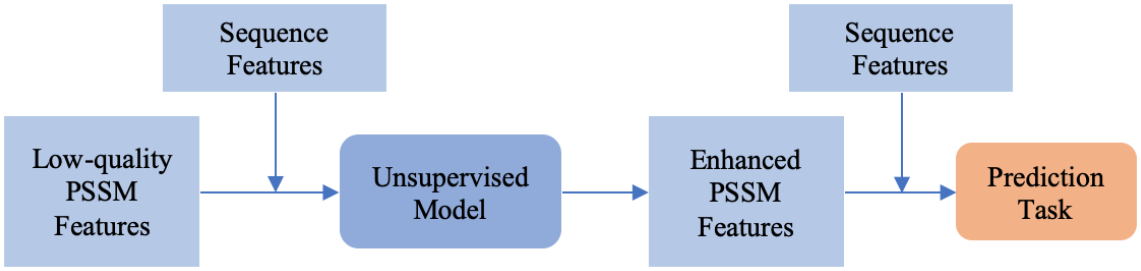


Figure 3.2. Framework Overview.

3.2.3 Protein secondary structure prediction

The sequence space of proteins is vast, with perhaps 20 residues at each position, and evolution has been sampling it over billions of years. One of the most important sub-problems in protein studies is the secondary structure prediction. Protein secondary structure refers to the local conformation of the polypeptide backbone of proteins. There are two regular SS states: alpha-helix (H) and beta-strand (E), and one irregular SS type: coil region (C) [16]. The other way is a DSSP algorithm [17] to classify SS into 8 fine-grained states. In particular, the algorithm assigns 3 types for helix (G, H and I), 2 types for strand (E and B), and 3 types for coil (T, S and L). Overall, many computational methods have been developed to predict both 3-state secondary structure and a few to predict 8-state secondary structure. Meanwhile, since a chain of 8-state secondary structures contains more precise structural information for a variety of applications [50, 15], the focus of secondary structure prediction has been shifted from 3-state secondary structure(Q3) prediction to the prediction of 8-state secondary structures(Q8). Because the Q8 problem is much more complicated than the Q3 problem, deep learning methods would be more suitable for addressing the Q8 problem.

3.3 Method

3.3.1 Framework overview

Our method consists of two stages: enhancing PSSM and secondary structure prediction. The workflow of the inference phase is shown in Fig. 4.2. We input the low-quality PSSM into the trained unsupervised model with the protein sequence features to generate enhanced PSSM features. Then the enhanced PSSM features with sequence features are concatenated as the input of the inference phase for the prediction network. Finally, the results of the enhanced PSSM and the original PSSM on the prediction model are compared for evaluation.

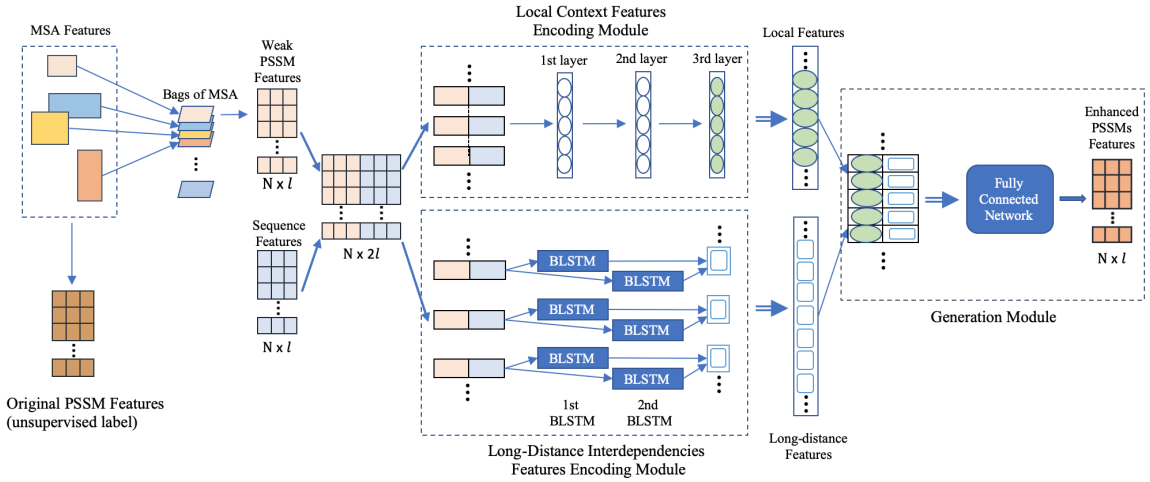


Figure 3.3. Unsupervised learning model. 1) Bagging MSA Module has two outputs:"Original PSSM" calculated by all MSA are used as the unsupervised labels; "Weak PSSM" calculated via the bags of MSA are fed into the two encoding networks. 2) The outputs of the two encoding networks are local features and long-distance features respectively. 3) The output of the generation module is the "Enhanced PSSM", which is used to calculate the loss from the"Original PSSM" to adjust the networks.

3.3.2 Unsupervised Learning to enhance PSSM

The architecture of our unsupervised learning method is shown in Fig. 3.3, which mainly contains four parts: Bagging MSA module, Local contexts feature encoding module, Long-distance interdependencies feature encoding module and Generation module. For each amino acid in a protein sequence, its input features are concatenated by its sequence features and PSSM features, which form a $2l$ (l=21) dimensional vector. We denote the size of the entire input features as $N \times 2l$, and the size of the output from unsupervised learning network is $N \times l$, where N is the length of the protein sequence. The details regarding input features are explained in the experiments section.

3.3.2.1 Bagging MSA

The main purpose of our enhancing PSSM module is to generate higher-quality PSSM features from low-quality PSSM features calculated from MSA with fewer rows or lower quality. Here we introduce the concept of 'Bagging MSA': As shown in Fig. 3.3, we randomly sample a small part of MSA for a protein and repeat this operation in each training iteration and for each protein. We bring in a hyper-parameter R to determine the proportion of selected homologous proteins in MSA randomly per training iteration, *e.g.*when R = [10%, 20%], a number greater than 10% and less than 20% would be randomly selected for each batch, and the homologous proteins in MSA would be randomly sampled according to this proportion. In this way, we are able to get many MSA bags, and each MSA bag would calculate a so-called 'Weak PSSM'. We used the weak PSSM calculated by these bags as a part of the input unsupervised data, and the original PSSM calculated by the complete MSA as the unsupervised labels. This module is ideal for unsupervised learning due to the

size of the PSSM matrix is always the same for the same protein, even though the MSA size of each bag and label is different.

3.3.2.2 Local contexts feature encoding module

We introduce a fully convolutional architecture as the local contexts feature encoding module. Recently, CNN has been successfully used in the seq2seq model [51] and machine translation [52], as well as applied in several protein studies, which achieved remarkable successes [50, 53]. This one-dimensional convolution operation is usually used to process sequence data, such as emotional analysis and sequence structure prediction [54, 18], so CNN would be a good fit for our prediction task.

In our method, the local contexts feature encoding module exploits the One-dimensional convolution to extract the local hidden patterns and features of adjacent amino-acid residues from the input matrix. This module contains three 1-d convolutional layers with the ReLU activation function, and the window size is equal to three for each layer, details are shown in Figure 3.4.

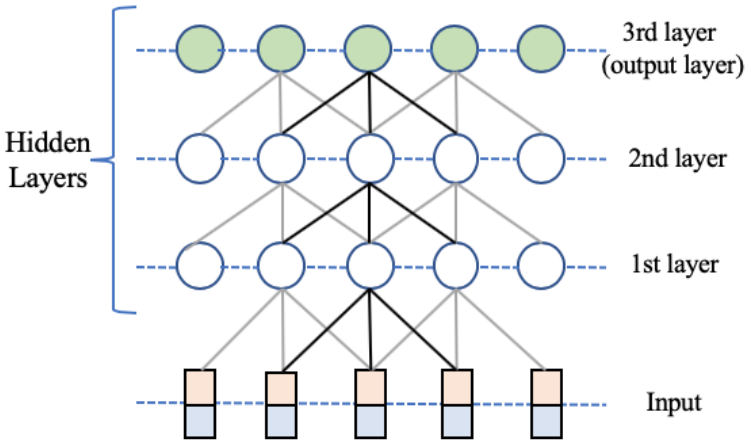


Figure 3.4. Local contexts feature encoding module includes three layers of 1d-CNN and the top layer(3rd layer) is the output layer.

3.3.2.3 Long-distance interdependencies feature encoding module

As we mentioned before, CNNs have the ability to capture local relationships of spatial or temporal structures, but we can not capture sufficient long-range sequence information by increasing the window size and network depth infinitely. However, long-distance interdependencies [45] of amino-acid residues are also critical for amino acid sequence information. Inspired by the success of some methods which use a combination of multiple neural networks, for example, coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks [55], ACLSTM[19] and CRRNNs [56], our method not only uses convolutional neural network with a few layers but also another network to catch Long-distance interdependencies feature.

RNN-based model has achieved remarkable results in sequence modeling; however, the gradient vector may grow or degrade exponentially over a long sequence during the training process. Thus LSTM neural networks are designed to avoid this problem by introducing the gate structures, which is good at capturing the long-range relations (from the first atom to the last one).

In our method, the long-distance interdependencies feature encoding module includes two stacked bidirectional LSTM neural networks. As shown in Figure 3.5, the input data are fed into the feature encoding model by its original order as well as the reverse order, and then the two outputs are concatenated together as the final features representation.

3.3.2.4 Generation module

Our method has one fully connected hidden layer in the generation module. Moreover, in order to get the complete information of protein sequence, as shown

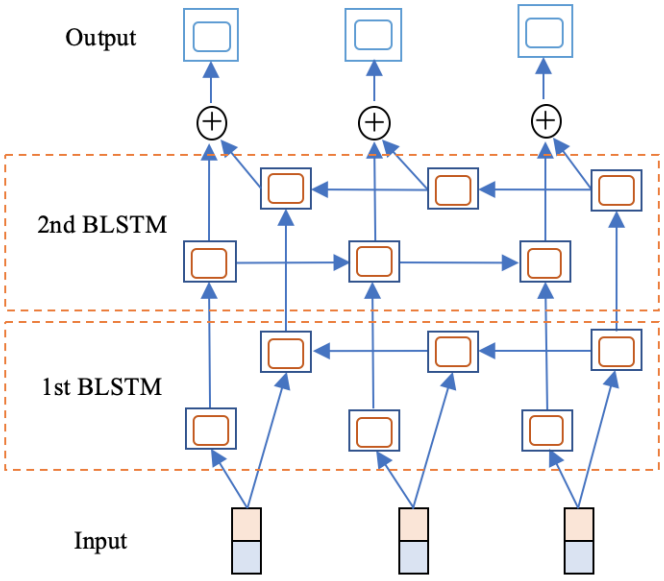


Figure 3.5. Long-distance interdependencies feature encoding module includes two stacked BLSTM neural networks.

in Fig. 3.3, we directly concatenate the outputs of the previous two modules and feed them into the fully connected(FC) layer with the ReLU activation function to generate the enhanced PSSMs. We use the MSE loss[57] to adjust our unsupervised network, as shown in Eq.3.4.

$$Loss_{unsup} = MSE(PSSM_{Enhanced}, PSSM_{Full}) \tag{3.4}$$

3.3.3 Prediction network

Since our unsupervised learning method is an independent enhancing PSSM network, we are able to use any deep learning network for the prediction module to verify the generalization of our method. In this study, we use two protein secondary structure prediction networks to evaluate our method: CNN-based network and LSTM-based network, which are two widely used deep learning prediction networks. For CNN-based method, we use five CNN layers [18], and fix the window

size to 11 since the average length of an alpha-helix is around eleven residues [58] and that of a beta-strand is around six [59]. For LSTM-based method, we use two stacked bidirectional LSTM neural networks [4] and a fully connected(FC) layer.

The input data for the prediction network is the same as the input for the unsupervised learning model, which is the concatenation of sequence information and PSSM features calculated by the complete MSA of the protein. The protein secondary structure is used as the label. Based on the validation results, we select the best model as the secondary structure predictor, then feed the enhanced PSSM features generated by our unsupervised network and the original PSSM into the predictor respectively. Last, the prediction performances of the two PSSM features are compared to evaluate the effectiveness of our enhanced PSSM model.

3.4 Experiments

3.4.1 Experiments set up

3.4.1.1 Dataset

We use four publicly available datasets: CullPDB [29] of 5926 proteins, CB513 [15] of 513 proteins, CASP11 of 85 proteins, and CASP12 of 40 proteins. CASP11 and CASP12 datasets are downloaded from the official CASP website (`http://predictioncenter.org`). 53 duplicated proteins observed in the CullPDB are removed and 591 proteins are randomly sampled for validation, then the remaining proteins are used for training. The other three datasets are used as the test dataset. We generate the position specific scoring matrix (PSSM) by searching the Uniref50 [60] database. And the labels used for the prediction network are 8-state protein secondary structures which are generated by DSSP [17, 61].

3.4.1.2 Input features

The input features for the encoding networks of our method are described in [15]. We extract the MSA from Uniref50 databases using Jackhmmer [43], and set the parameters refer to their guide [62], details are listed in 3.4.1.5. We randomly sample 10% to 20%(R = [10%, 20%]) of the MSA for each protein within each learning iteration(Bagging MSA), and then we calculate PSSM using Eq.(3.1) and Eq.(3.2). We transform those PSSMs by the Sigmoid function 1/(1+exp(-x)) where x is a PSSM entry to map each PSSM value in between 0 and 1. As shown in Fig. 3.3, the input features of the two encoding modules is a $N \times 2l$ matrix, where N is the length of the input sequence and $2l$ is the dimension of the concatenated vectors. In our method, the sequence feature vectors are sparse one-hot vectors of 21 elements(l=21) since there might be some unknown amino acids in a protein sequence. Therefore, there are 42 input features in total for each residue, 21 from PSSM features and the other 21 from sequence feature.

For the prediction part, there are 42 input features for each residue too, 21 of them are from weighted PSSM features and the others are from sequence feature. We compare the testing results of the enhanced input features with the original input features to evaluate the effectiveness of our unsupervised model.

3.4.1.3 Neural network structure and learning Hyper-parameters

The framework of our unsupervised learning method is very flexible in the network structure selection.

In the long-distance interdependencies feature encoding module, we can set different hidden layers and hidden dimensions (with different layers and layer hidden sizes). Moreover, different types of network can be chosen in addition to the bi-LSTM

network, such as LSTM [63]. Due to the space limitation of this paper, two stacked bi-LSTM with 512 hidden units are used for all experiments. Then, we use 1d-CNN of 3 hidden layers, and 100 neurons for each layer in the local contexts feature encoding module. The window size at each layer is set to 3.

For optimization, we use multi-step LR(learning rate) descent with [30,100,200] for epoch indices. The multiplicative factor of learning rate decay is 0.1. We use Adam [64] as the optimizer of our method. The initial learning rate for all training models is 0.0001.

For the protein secondary structure prediction task, we have two kinds of networks. For CNN network, we use five 1-dim CNN layers with window size 11, and neurons size 100 for each layer. For LSTM network, we use two stacked bi-LSTM with 512 hidden units and one fully connected(FC) layer.

3.4.1.4 Evaluation metric

For the unsupervised learning, we calculate the RMSE of the Enhanced PSSM and the Original PSSM in the input feature as the evaluation matrix. Q8 accuracy is the criterion of the prediction module.

3.4.1.5 Jackhmmer options for extracting MSA

In the per-target output, report target profiles with an E-value <=1.0; In the per-domain output, for target profiles that have already satisfied the per-profile reporting threshold, report individual domains with a conditional E-value of <=1.0; Use a conditional E-value of <=0.03 as the per-domain inclusion threshold, in targets that have already satisfied the overall per-target inclusion threshold; Obtain residue alignment probabilities from the built-in substitution matrix named BLOSUM62.

3.4.1.6 Infrastructure and software

Our model was implemented through Pytorch package. And our models was trained in a self-hosted 16-GPU cluster platform with Intel i7 6700K @ 4.00 GHz CPU, 64 Gigabytes RAM and four Nvidia GTX 1080Ti GPUs on each workstation.

Figure 3.6. The average accuracy of proteins within Count score ranges (a) CNN-based prediction model; (b) LSTM-based prediction model.

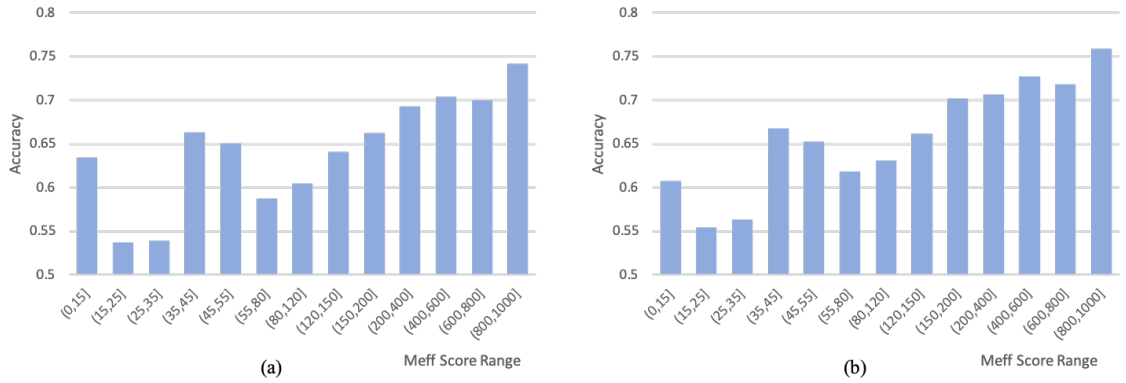


Figure 3.7. The average accuracy of proteins within Meff score ranges (a) CNN-based prediction model; (b) LSTM-based prediction model.

Table 3.1. Number of proteins in certain Count Score ranges.

range	(0,20]	(20,40]	(40,60]	(60,80]	(80,120]	(120,150]	(150,200]	(200,300]	(300,500]	(500,700]	(700,900]	(900,1000]
num	2	16	18	19	29	11	23	27	45	26	26	271

3.4.2 Results

3.4.2.1 Relationship between PSSM quality and performance

As we mentioned before, we use two methods to score the quality of the protein PSSM, higher score represents better quality. Fig. 3.6 and Fig. 3.7 show the relationship between the quality of PSSM and the corresponding performance on the prediction networks on CB513 dataset. Fig. 3.6 shows the average accuracy obtained by using Count score as the evaluation standard on the prediction network of CNN and LSTM respectively, and Fig. 3.7 for the Meff score. We can find that proteins with high-quality PSSM performs better than proteins with low-quality PSSM both CNN-based and LSTM-based prediction network, as well as under all evaluations including Count score or Meff score. Table 3.1 and table 3.2 show the data distribution within the ranges Count and Meff Scores. Thus, our method aims at improving the prediction performance for those proteins with original low-quality PSSM by enhancing their PSSM features. As shown in Fig. 3.8, which is a set of gray-scale images of the original pssm(a) and enhanced pssm(b) of a protein from cb513 dataset. Where, y-axis is the length N of the protein sequence, the sample protein contains 26 residues(N=26), x-axis is l, 20 plus an unknown amino acids marker(l=21). Lighter colors indicate larger values, while darker colors indicate smaller values. See `https://www.rcsb.org` for the structure information of the protein(6O4M) in the example.

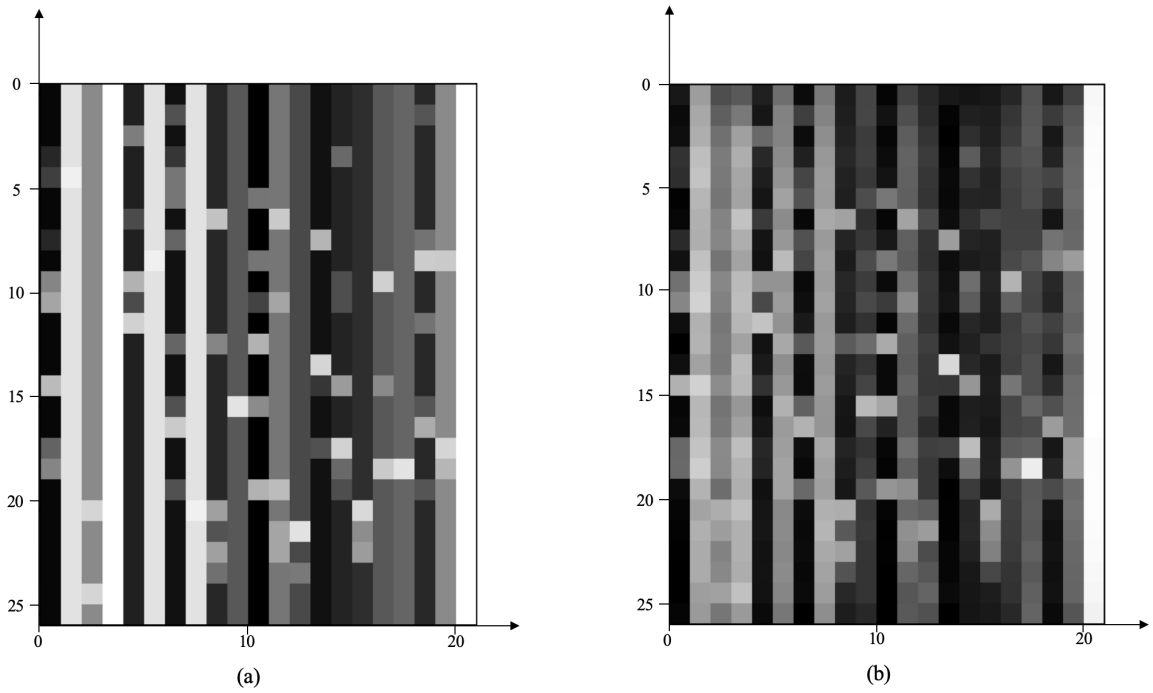


Figure 3.8. Gray-scale images of the PSSMs. (a) Original PSSM of 6O4M protein; (b) Enhanced PSSM of 6O4M protein.

Table 3.2. Number of proteins in certain Meff Score ranges.

range	(0,15]	(15,25]	(25,35]	(35,45]	(45,55]	(55,80]	(80,120]	(120,150]	(150,200]	(200,400]	(400,600]	(600,800]	(800,1000]
num	12	23	18	9	16	18	19	15	23	68	89	89	114

3.4.2.2 Enhancement on low-quality PSSM protein

Our method is used to enhance the performance of proteins with low-quality PSSM in secondary structure prediction task. However, while improving the low-quality PSSM, noise might have been added to the high-quality PSSM, which would end up with a lower accuracy score. Therefore, we need to find a standard to determine the definition of low-quality proteins for our method, which would be the thresholds of the Count score and the Meff score. As shown in Fig. 3.9, our method increase or decrease the accuracy of prediction tasks under certain ranges. Greater

than 0 means that the average accuracy of our method has improved under the threshold, while less than 0 means that it has decreased. Based on the accuracy results, we are able to find a consistent trend for both CNN-based and LSTM-based models: our method shows significant superiority for proteins with a Count score less than 60 and a Meff score less than 35.

In addition, in order to verify the threshold we selected is suitable for other datasets, we also report the results of casp11 and casp12, which are shown in table 3.3. The performances of extensive experiments demonstrate that our method has a significant effect on enhancing low-quality PSSM for different datasets.

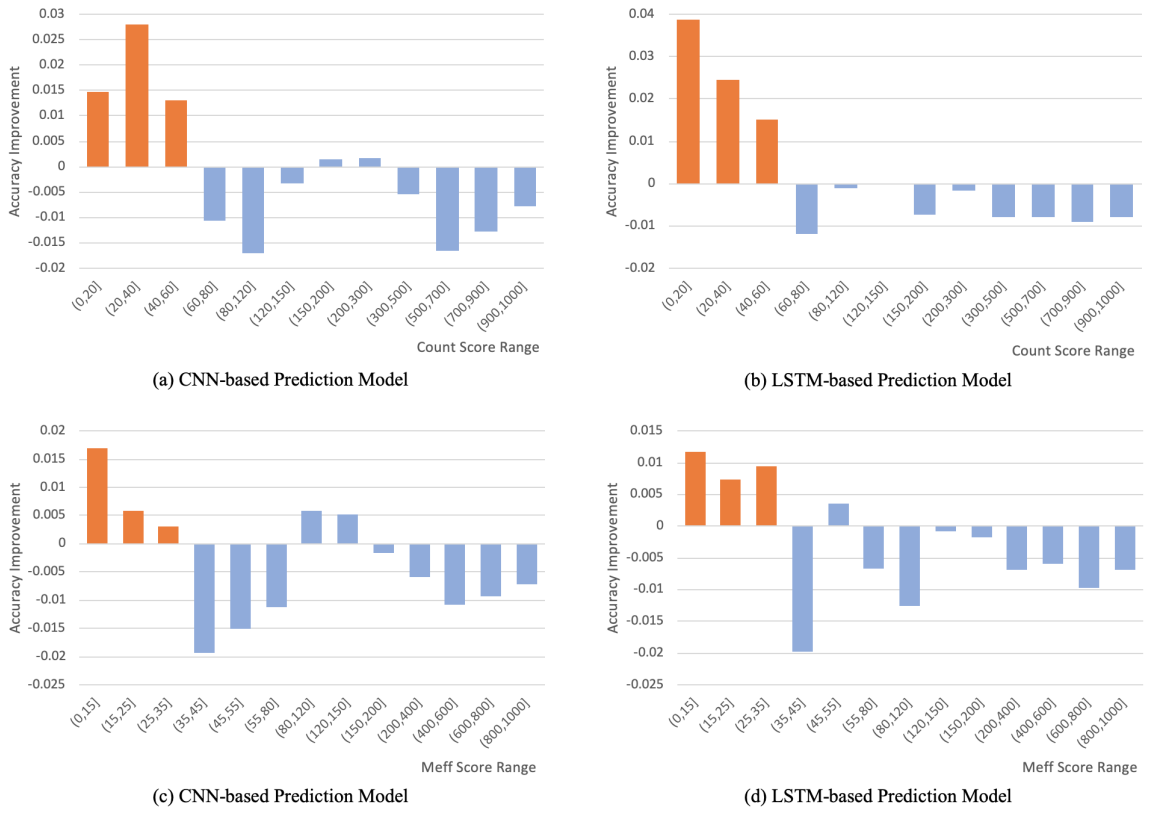


Figure 3.9. Our method has achieved significant improvement in all prediction tasks (CNN-based and LSTM-based) when the Count Score is less than 60 (a, b), and the Meff Score is less than 35 (c, d). These figures are the results on CB513 dataset.

Table 3.3. Comparison results (Q8 accuracy) of our Enhanced PSSM vs. Original PSSM. Enhancement experiments are conducted for low-quality proteins (Count score <= 60, Meff score <= 35) obtained from CB513, CASP11, and CASP12 datasets. Prediction experiments are conducted on CNN-based model and LSTM-based model.

Pred model	Score range	Datasets	Original PSSM	Enhanced PSSM	Num
CNN-based	Count ≤ 60	CB513	59.106%	61.093%	36
		CASP11	64.196%	67.781%	12
		CASP12	53.300%	56.519%	3
	Meff ≤ 35	CB513	55.973%	56.717%	53
		CASP11	62.846%	65.732%	17
		CASP12	52.353%	54.462%	7
LSTM-based	Count ≤ 60	CB513	60.982%	63.041%	36
		CASP11	64.037%	64.990%	12
		CASP12	54.335%	55.865%	3
	Meff ≤ 35	CB513	56.929%	57.831%	53
		CASP11	63.216%	63.504%	17
		CASP12	51.493%	53.921%	7

3.5 Conclusion

We propose an innovative Bagging MSA model to enhance low-quality PSSM features of proteins, which would help promote their performance in secondary structure prediction task. We employ an unsupervised learning network to enhance the PSSM features, and two conventional deep learning prediction models as the protein secondary structure prediction networks to prove the effectiveness of our method on various datasets. Our method is the first attempt to enhance PSSM features in the field of protein research. Moreover, the generalization of our Bagging MSA makes it suitable for numerous PSSM related protein prediction tasks. PSSM features are essential for studying proteins, our method pioneer another way to address the prediction limitation for low-quality proteins.

CHAPTER 4

WEIGHTALN: WEIGHTED HOMOLOGOUS ALIGNMENT FOR PROTEIN STRUCTURE PROPERTY

I introduce a novel Multiple Sequence Alignment (MSA) weights learning framework, WeightAln, which generates learnable MSA weights for protein prediction tasks using attention-based deep learning techniques in this chapter. Extensive experiments on three protein structure property prediction tasks, secondary structure, solvent accessibility, and backbone dihedral angles prediction, sufficiently demonstrate the effectiveness of the method.

4.1 Introduction

Recently, AlphaFold [65] and RosettaFold [66] have achieved great success in the task of predicting the three-dimensional structure of proteins. However, research on protein sub-problems is still important and in progress [67, 68], such as secondary structure, solvent accessibility, and backbone dihedral angles [50, 33]. These predicted properties can not only be combined eventually to help the final prediction of protein structure, but also help the biological scientists or researchers on specific tasks [69, 70, 71].

The sequence-based protein homologous alignment has been extensively explored and utilized to protein structure property prediction. Multiple sequence alignment (MSA) of sequence homologs in a protein family is now the most prevalent method for homology detection [9]. To facilitate the comparison and alignment, a MSA is usually represented as a position-specific scoring matrice (PSSM). Such imple-

mentation is commonly used in the protein structure property prediction [35, 72, 36]. In specific, PSSM is generated from MSA by simply calculating the frequency of various types of amino acids at each residue position [10]. However, the homologous sequences in the MSA usually contains some redundancy, and each sequence may also contribute differently while being used in different prediction tasks [40, 39]. Although some PSSM-based deep learning methods [19, 36, 73] have achieved remarkable performance in protein structure property prediction, the problem of redundancy in MSA has not been solved. [39] introduces a scoring method to measure the MSA quality, denoted as M_{eff}, represents the effective sequence number of non-redundant sequences after re-weighting. However, the M_{eff} weight is not flexible and learnable, which can not be adjusted by the prediction networks according to different tasks. Recent studies in natural language processing (NLP) and image processing have demonstrated that the attention mechanism is a powerful tool for extracting weighted information from words in the language sentence and pixels in the images [74, 75, 76], which inspires us to adapt attention to learning MSA weights.

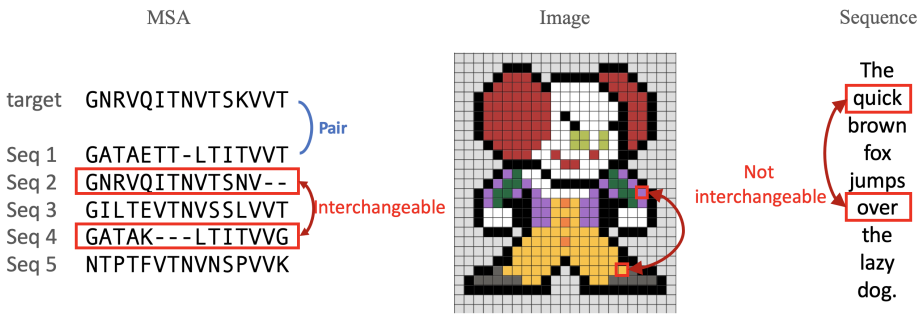


Figure 4.1. Comparison between MSA, image and sequence.

However, applying attention mechanism to our problem is not straightforward. There are two major discrepancies between previously mentioned areas and our prob-

lem, which makes the adaptation challenging. 1) The order of sequences is not important, while the position of pixels in an image and words in a sentence are critical, as indicated in Figure 4.1. This makes us design a permutation-invariant attention mechanism. 2) For each sequence and picture, the attentions are calculated separately. But in our MSA weighting problem, each sequence is derived from the target protein. Thus, such an individual is meaningless for the prediction task. Therefore, we believe a pair-wise attention mechanism should be exploited to learn the relationship between the target protein and its homologous sequences.

In this paper, we present WeightAln, an attention-based framework to obtaining weighted MSA for enhancing protein structure property prediction. To be more specific, each MSA sequence is paired with the target protein as the input, while the order of the pairs has no impacts. Then, we employ fully-connected neural networks as the Weighting MSA model for MSA weights generation. Next, weighted PSSM is calculated using MSA and its corresponding weights, then the weighted PSSM is input to the protein structure prediction task. We demonstrate the effectiveness of our method by conducting experiments on three supervised protein structure property prediction tasks: secondary structure, solvent accessibility, and backbone dihedral angles prediction.

The technical contributions of this paper are summarized as: 1) Our method proposes a fresh insight for protein structural prediction problem by training a weighted MSA. 2) Our WeightAln method is flexible, making it easily to plug-and-play with any downstream networks and any prediction tasks. Extensive experiments on three different protein structure property prediction tasks prove that our model outperforms other baseline methods on various prediction tasks.

The rest of the paper is organized as follows. Section 2 introduces the materials and methods, and the framework and modules used in our method are described in

detail. Section 3 illustrates the experimental results, which validates the superiority of our method. Last, we discuss the results and prospect of our method, and conclude our paper in Section 4 and 5.

4.2 Protein structure property prediction

We provide three protein structure property relevant downstream prediction tasks to serve as the benchmarks, which are secondary structure, solvent-accessibility, and backbone dihedral angles predictions. We use those three benchmarks to evaluate our method.

4.2.1 Eight-State Secondary Structure (SS) Prediction

Protein secondary structure refers to the local conformation of the polypeptide backbone of proteins [7]. DSSP algorithm [17] classifies SS into 8 fine-grained states: 3 types for helix (G, H and I), 2 types for strand (E and B), and 3 types for coil (T, S and L). Overall, secondary structure prediction is an eight-class classification problem at each amino acid position.

4.2.2 Three-State Relative Solvent Accessibility (RSA) Prediction

The solvent accessibility (ASA) is defined as the surface region of a residue that is accessible to a rounded solvent while probing the surface of that residue. The relative solvent accessibility (RSA) is the extent of the ASA of a given residue and is related to the residue spatial arrangement and packing [77]. Based on the RSA value, the prediction is a 3-state classification task where each input amino acid x_i is mapped to a label $y_i \in \{Buried(B), Intermediate(I), Exposed(E)\}$. According to the settings of [78], we use the threshold of 10% for B/I and 40% for I/E for the 3-state classification.

4.2.3 Backbone Dihedral Angles Predictions

Protein dihedral angles provide a detailed description of protein local conformation [79]. The dihedral angles – Φ value and Ψ value prediction [80, 36], are regression tasks.

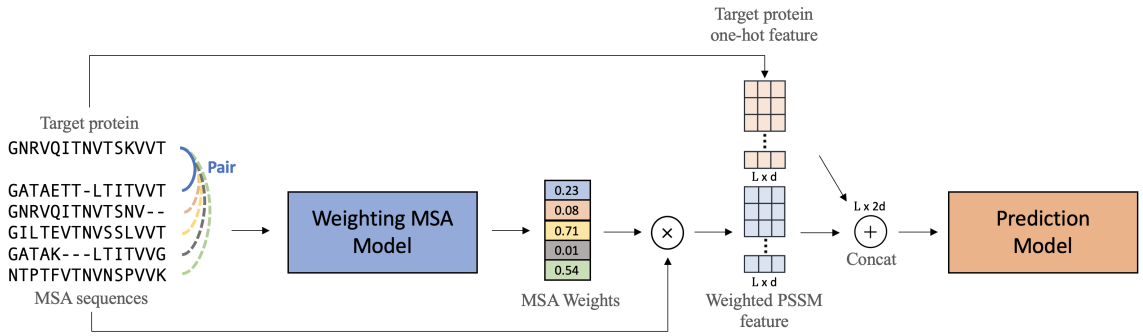


Figure 4.2. WeightAln framework is constituted by a Weighting MSA module and downstream prediction tasks. The Weighting MSA module contains three parts, Pairing MSA process, Weighting MSA model, and Weighted PSSM calculation. The prediction task can be any protein structure property prediction with any prediction model (network). Our Weighting MSA model takes a sequence pair as input, which is constructed by aligning a sequence in MSA against the target protein sequence. For the prediction model, there are $2d$ input features for each residue, where d of them are from weighted PSSM features and the others are from the sequence features, L is the length of the target protein sequence. $d = 21$ denotes the presence of 20 known amino acid types and one unknown type.

4.3 Method

As shown in Figure 4.2, the entire pipeline of our method consists of three parts: First, we input the pairs of the target protein and each sequence in MSA to the Weighting MSA model. Then we use the generated MSA Weights from the Weighting MSA model along with the MSA sequences to calculate the weighted PSSM. Finally,

we concatenate the target protein sequence features (one-hot vectors) and weighted PSSM features to feed into the prediction model.

4.3.1 Weighting MSA

To leverage the contribution of different MSA sequences while calculating the PSSM, we propose a novel weighting MSA network that assigns a weight to each MSA sequence. Specifically, we follow the same experimental protocol as previous studies [50, 8] to convert the target protein t and the corresponding MSA sequences to one-hot feature vectors, which are represented by $\mathbf{X}_t \in R^{L\times d}$ and $\mathbf{H}^{(t)} \in R^{N\times L\times d}$, respectively. Here, L is the sequence length, N is the number of MSA sequences, $d = 21$ indicates 20 known amino acid types and one unknown type. Since the order of MSA sequences is interchangeable, we simply apply those obtained feature vectors in the following study, rather than use the position embedding technique for feature pre-processing as previously reported in Fig. 4.1.

Given that a single sequence in MSA is meaningless for the prediction task, our weighting MSA network takes the $\left(\mathbf{X}_t, \mathbf{H}_k^{(t)}\right)$ pair as inputs to produce the weight $\alpha_{t,k}$ via three stacked fully-connected layers. $\mathbf{H}_k^{(t)}$ represents the k-th sequence in the MSA of the t-th target protein. Each fully-connected layer is defined as:

$$\mathbf{z}_{l+1} = \sigma\left(\mathbf{W}_l^T \mathbf{z}_l\right),\ l = 1, 2, 3 \tag{4.1}$$

where $\sigma(\cdot)$ is the ReLU function, and l is the layer index. The l-th layer's input is denoted by $\mathbf{z}_l$, where $\mathbf{z}_1$ is the concatenation of $\mathbf{X}_t$ and $\mathbf{H}_k^{(t)}$, and z_4 is the last layer's output, which is a scalar.

In order to ensure that the per-sequence weight $\alpha_{t,k}$ lies between 0 and 1, we apply the sigmoid function to the last layer's output z_4, *i.e.* $\alpha_{t,k} = 1/\left[1 + \exp\left(-z_4\right)\right]$.

Finally, for each target protein $\mathbf{X}_t$, we obtain a series of MSA weights $\{\alpha_1, \ldots, \alpha_N\}$, where the subscript t is dropped for simplicity.

Then we use the MSA and its Weights set to calculate the weighted PSSM as the input feature for prediction tasks.

The essence of PSSM is a matrix composed of the frequency of homologous amino acids corresponding to each residue position on the protein sequence. Given a protein sequence P with L amino acids, it can be formulated as:

$$P = A_1 A_2 A_3 A_4 A_5 \cdots A_L \tag{4.2}$$

where A_1 represents the 1st residue, A_2 the 2nd residue, and so forth. We count the number of each homologous amino acid on each residue, represented by Position Specific Count Matrix (PSCM):

$$PSCM = \begin{bmatrix} C_{1,1} & C_{1,2} & \cdots & C_{1,21} \\ C_{2,1} & C_{2,2} & \cdots & C_{2,21} \\ \vdots & \vdots & \ddots & \vdots \\ C_{L,1} & C_{L,2} & \cdots & C_{L,21} \end{bmatrix}, \; C_{i,j} = \sum_{k=1}^{n} F(M_{i,k}, j) \tag{4.3}$$

where 21 represents the 20 types of standard amino acids plus 1 unknown marker, the placeholder "–" for MSA in Fig. 4.1 is also regarded as the unknown amino acid type. The element $C_{i,j}$ represents the occurrence count of amino acid j ($j \in \{1, 2, 3, ..., 21\}$) at position i ($i \in \{1, 2, 3, ..., L\}$) of the protein sequences in MSA, the rows of the matrix represent the positions of the sequence, and the columns of the matrix represent the 21 types of amino acids. The return value of $F(M_{i,k}, j)$ is equal to 1 when $M_{i,k}$ is equal to j, otherwise it is equal to 0, where $M_{i,k}$ represents the amino acid type at the i-th position in the k-th MSA sequence.

The PSSM calculation formula is same as [48, 49, 8]:

$$score_{i,j} = \frac{C_{i,j} + Pseudocount}{N + 20Pseudocount} \tag{4.4}$$

$$PSSM_{i,j} = \log(score_{i,j}/Backgroundfrequency_j) \tag{4.5}$$

A simple procedure called pseudo-counts [8] assigns minimal scores to residues which do not appear at a certain position of the alignment according to the following equation(4.7), where we set the *Pseudocount* equal to 1. N is the number of sequences in the multiple alignments. The *Background frequency*$_j$ in Eq.(4.8) is the frequency of residue j appearing in the entire MSA of the protein.

The PSCM counting and PSSM calculation methods actually default the weight of each sequence in the MSA to 1. According to the MSA weights generated by our method, we can calculate weighted $C'_{i,j}$ by:

$$C'_{i,j} = \sum_{k=1}^{n} F(M_{i,k}, j) \times \alpha_k \tag{4.6}$$

and our weighted PSSM calculation can be represented by:

$$score'_{i,j} = \frac{C'_{i,j} + Pseudocount}{\sum_{k=1}^{N} \alpha_k + 20Pseudocount} \tag{4.7}$$

$$PSSM'_{i,j} = \log(score'_{i,j}/Backgroundfrequency'_j) \tag{4.8}$$

where α_k is the weight of the k'th MSA sequence, and the *Background frequency*$'$ is counted by the entire weighted MSA.

Finally, we input the weighted PSSM generated by the Weighting MSA module into the downstream network for protein structure property prediction.

4.3.2 Prediction Networks

Since our Weighting MSA network is an independent model, we can use any deep learning networks for any prediction tasks. In this study, we use two deep learning based networks as the downstream prediction networks to evaluate our method for all prediction tasks: CNN-based network and LSTM-based network, which are two

widely used deep learning based protein structure property prediction networks [50, 4, 18, 36]. For CNN-based method, we use five CNN (Convolutional neural network) layers [18], and set the window size as 11 since the average length of an alpha-helix is around eleven residues [58] and that of a beta-strand is around six [59]. For LSTM-based method, we use two stacked bidirectional LSTM (Long short-term memory) neural networks [4] and a fully connected(FC) layer.

In addition, in order to further prove the extensiveness and effectiveness of our method, we also apply two state-of-the-art prediction methods as the downstream networks to evaluate our method: 1) MUFOLD-SS [31] is a deep inception-inside-Inception (Deep3I) network architecture which extends deep inception networks through nested inception modules. Stacked inception modules could extract non-local residue interactions at different ranges. Overall, MUFOLD-SS is an efficient method for predicting 8-state SS and has the ability to extract more complex sequence-structure information between amino-acid residues. 2) The NetSurfP-2.0 [73] uses an architecture composed of one dimension convolution neural networks (1d-CNN) and long short-term memory (LSTM) neural networks. Due to the efficient performance of NetSurfP-2.0 method in dihedral angles prediction, we use this model as a downstream network to evaluate the performance of our method on dihedral angles (Phi-Φ and Psi-Ψ values) prediction task.

As shown in Figure 4.2, the input data for the prediction network is the concatenation of sequence features and weighted PSSM features calculated by MSA and its weights. We utilize Cross-Entropy loss [81] for secondary structure and relative solvent accessibility prediction tasks, and use MSE Loss [57] for dihedral angles prediction. The supervised loss is back-propagated to both Weighting MSA and the prediction networks.

4.4 Results

4.4.1 Experimental settings

4.4.1.1 Datasets

We use four publicly available datasets, CullPDB [29], CB513 [15], CASP11 and CASP12. They are obtained from GSN [15] and official CASP website. Cullpdb dataset is widely used in protein structure prediction [82, 18]. We use the CullPDB dataset of 5926 proteins for training and validation. 53 duplicated proteins with the other three datasets that we use for testing, are removed from the dataset, which means proteins in the dataset share no more than 25% sequence identity with our other datasets for testing [18]. And 591 proteins are randomly sampled for validation. The remaining proteins are used for training. The other three datasets: CB513 of 513 proteins, CASP11 of 85 proteins, and CASP12 of 40 proteins are used as the test dataset on secondary structure (SS) prediction task. We use the SS label of CB513 dataset provided by [15] and the label for CASP datasets are generated by DSSP [17, 61]. For relative solvent accessibility (RSA) and dihedral angels, we only use CASP11 and CASP12 as the test sets. The reason is that the PDB files of CB513 dataset are not released by [15]. Thus, the RSA, Phi, and Psi angles labels cannot be generated. We extract the MSA from Uniref50 databases [60] using Jackhmmer [43], and set the parameters refer to their guide [62]. The details are listed in Section 4.4.1.5. The labels used for the protein dihedral angles are generated by DSSP, and the RSA labels are generated based on [78].

4.4.1.2 Input features

The input for our Weighting MSA module is a sequence pair, which is constructed through aligning a sequence in MSA against the target protein sequence. In

our method, each sequence feature vector is a sparse $L \times 21$ one-hot vector where L is the length of the sequence and 21 is the sum of twenty known amino acids appeared in the genetic code and one unknown amino acid marker. Therefore, the dimension of each sequence pair input to the model should be $L \times 42$. For the prediction module, there are 42 input features for each residue too, where 21 of them are from weighted PSSM features and the others are from sequence one-hot feature. When running experiments for the baseline method, we use the original PSSM instead of the weighted PSSM generated by our method.

4.4.1.3 Evaluation metric

Secondary structure (SS) prediction task is a eight-state (Q8) classification problem and relative solvent accessibility (RSA) is a three-state (Q3) prediction task. We use Q8 (eight-state classification) accuracy and Q3 (three-state classification) accuracy as the criterion of those prediction tasks, respectively. For dihedral angles prediction, we evaluate the performance by Mean Absolute Error (MAE) as described by [83].

4.4.1.4 Neural network structure and learning Hyper-parameters

To learn a weight value for each sequence pair, we use 3 fully connected layers in the Weighting MSA network, and the activation function is ReLU. Sigmoid function is used for mapping each output weight value between 0 and 1. For optimization, we use multi-step LR (learning rate) descent with [30,100] for epoch indices (120 epochs totally). The multiplicative factor of learning rate decay is 0.1. We use Adam [64] as the optimizer of our method. The initial learning rate for all training models is 0.001. For the protein secondary structure prediction task, we have two kinds of networks. For CNN network, we use five 1-dim CNN layers with window size 11, and neurons

size 100 for each layer. For LSTM network, we use two stacked bi-LSTM with 512 hidden units and one fully connected(FC) layer. We utilize Cross Entropy loss to supervise all networks to predict SS and RSA, and use MSE Loss [57] for dihedral angles prediction [11].

4.4.1.5 Jackhmmer parameters

We use the following parameter to extract MSA using Jackhmmer software [62]: in the per-target output, report target profiles with an E-value <=1.0; in the per-domain output, for target profiles that have already satisfied the per-profile reporting threshold, report individual domains with a conditional E-value of <=1.0; use a conditional E-value of <=0.03 as the per-domain inclusion threshold, in targets that have already satisfied the overall per-target inclusion threshold; obtain residue alignment probabilities from the built-in substitution matrix named BLOSUM62. For each target protein, we use up to the first 1,000 sequences extracted from the Jackhmmer as the MSA features of the protein.

4.4.1.6 Infrastructure and software

Our model was implemented through Pytorch package. And our models was trained in a self-hosted 16-GPU cluster platform with Intel i7 6700K @ 4.00 GHz CPU, 64 Gigabytes RAM and four Nvidia GTX 1080Ti GPUs on each workstation.

4.4.2 Experimental results

As shown in Table 4.1 and Table 4.2, we compare the results of using our WeightAln method and not using our method (baseline) on the same prediction network on different prediction tasks. Because of the independence of our model, we can use any downstream network as the prediction network to evaluate our method. Table 4.1

Table 4.1. Comparison results on **CNN-based** prediction network

Datasets/Methods	SS [Q8]	RSA [Q3]	Phi [MAE]	Psi [MAE]
CB513				
Baseline	0.678	-	-	-
WeightAln	**0.683**	-	-	-
CASP11				
Baseline	0.701	0.637	22.09	38.28
WeightAln	**0.704**	**0.647**	**20.77**	**36.19**
CASP12				
Baseline	0.670	0.608	26.64	37.87
WeightAln	**0.674**	**0.611**	**22.99**	**35.79**

Table 4.2. Comparison results on **LSTM-based** prediction network

Datasets/Methods	SS [Q8]	RSA [Q3]	Phi [MAE]	Psi [MAE]
CB513				
Baseline	0.699	-	-	-
WeightAln	**0.706**	-	-	-
CASP11				
Baseline	0.704	0.652	21.40	34.98
WeightAln	**0.715**	**0.662**	**20.57**	**34.35**
CASP12				
Baseline	0.672	0.625	24.94	33.15
WeightAln	**0.683**	**0.634**	**22.61**	**32.85**

shows the results of CNN-based prediction network and Table 4.2 for LSTM-based prediction network. The different predicted structure properties are reported in the column header, together with the corresponding performance metric. For secondary structure (SS) and relative solvent accessibility (RSA), the evaluate metric is Q8 accuracy and Q3 accuracy (higher is better), respectively. For dihedral angles (Phi and Psi), the evaluate metric is MAE (lower is better). For each structure property and each dataset, the best score is reported in bold. Empty cells represent predictions that were not performed, because the structural property is not present in the corresponding dataset.

"Baseline" in the tables refers to the method using the original PSSM and sequence features as the input features of prediction networks (CNN-based/LSTM-based). "WeightAln" is our method which uses Weighting MSA network to generated weighted PSSM, and the weighted PSSM features are then used instead of the original PSSM to feed into the prediction networks. The following performance metrics are used for evaluation: Q8 accuracy for secondary structure; Q3 accuracy for relative solvent accessibility; and mean absolute error (MAE) in degrees for dihedral angles (Phi-Φ and Psi-Ψ angles) prediction. For classification tasks on CNN-based and LSTM-based prediction networks, our method achieve up to 1.1% improvement on SS prediction benchmark and 0.9% on RSA benchmark. For regression tasks, our method achieve up to 13.7% improvement on dihedral angles prediction benchmarks compared with using original PSSM features on prediction networks (baseline). The experimental results show that our method significantly improve the performance of all three structure property prediction tasks on both CNN-based and LSTM-based downstream networks.

In addition, in order to further prove the flexibility and effectiveness of our method, we conduct extensive experiments on two state-of-the-art prediction methods with the WeightAln method. As shown in Tab. 4.3 and Tab. 4.4, our WeightAln model further boosts the performance of the MUFOLD-SS method on secondary structure prediction and NetSurfP-2.0 method on dihedral angles (Phi and Psi angles) prediction tasks. The weighted PSSM generated by our WeightAln model is learnable and suitable for different prediction tasks. The results show that when we apply a powerful (state-of-the-art) prediction network, our Weighting MSA network can also generate a strong weighted PSSM to further improve the performance of the prediction network.

Table 4.3. Comparison results (Q8 acc) on **Mufold-ss-based** secondary structure prediction network*

Methods	CB513	CASP11	CASP12
MUFOLD-SS** [31]	0.704	0.717	0.684
WeightAln-MUFOLD-SS	**0.724**	**0.722**	**0.692**

* The different test sets are reported in the column header. The evaluate metric of the secondary structure prediction is Q8 accuracy (higher is better).

** To make a fair comparison, we use the data generated by our experiment to retrain the baseline.

Table 4.4. Comparison results (MAE) on **NetSurfP-2.0-based** dihedral angles prediction network*

Methods	CASP11 [Phi]	CASP11 [Psi]	CASP12 [Phi]	CASP12 [Psi]
NetSurfP-2.0** [73]	20.18	34.72	21.34	32.90
WeightAln-NetSurfP-2.0	**19.85**	**34.14**	**20.89**	**32.31**

* The different test sets are reported in the column header, together with the corresponding dihedral angle type (Phi / Psi). The evaluate metric of the dihedral angles prediction is mean absolute error (MAE) in degrees (lower is better).

** To make a fair comparison, we use the data generated by our experiment to retrain the baseline.

4.5 Discussion

Extensive experiments show that our WeightAln method has a stable effect on different prediction networks, that is, no matter whether the prediction network is complex or simple, powerful or not, our method can significantly improve the prediction performance. Our embedded weighting MSA module provides a new way of thinking for various downstream tasks of proteins. In the future, we will conduct experiments on higher dimensional protein prediction problems, such as contact map prediction and distance prediction.

4.6 Conclusion

We propose an innovative MSA weights learning framework, WeightAln, which learns from MSA and generate weighted PSSM. The weighted PSSM then can be utilized to improve the performance of protein structure property prediction tasks. First, a weighting MSA model is employed to generate different weights for the MSA of each target protein. Next, a weight PSSM is calculated according to the MSA and its corresponding weights. The weighted PSSM is then used in different prediction tasks instead of the original PSSM. Our proposed method explores the importance of each MSA sequence, and further generates learnable weighted PSSM for target protein. In addition, our weighting MSA model is independent, and can be easily embedded into various alignments related protein prediction tasks. Such implementation helps improve the quality of PSSM, which provides a fresh insight to improve the protein structure property prediction performance.

CHAPTER 5

SELF-SUPERVISED PRE-TRAINING FOR PROTEIN EMBEDDINGS USING TERTIARY STRUCTURES

I propose a self-supervised pre-training model for learning structure embeddings from protein tertiary structures. Native protein structures are perturbed with random noise, and the pre-training model aims at estimating gradients over perturbed 3D structures. I demonstrate the effectiveness of our pre-training model on two downstream tasks, protein structure quality assessment (QA) and protein-protein interaction (PPI) site prediction. Hierarchical structure embeddings are extracted to enhance corresponding prediction models. Extensive experiments indicate that such structure embeddings consistently improve the prediction accuracy for both downstream tasks.

5.1 Introduction

The biological functions of a protein, as well as its possible interaction with other molecules, are largely determined by its 3-dimensional structure [84]. For various protein-related applications, *e.g.* structure-based drug design (SBDD) [85, 86] and protein-protein interaction (PPI) prediction [87, 88], protein tertiary structures are one of the most critical features. However, it is time-consuming and costly to collect 3D structures for protein-ligand complex and multi-protein complex via experimental structure determination. As a result, the performance of SBDD and PPI models is often restrained by the limited structure data. On the other hand, computational methods for protein structure prediction have attracted increasing attention

for many decades. A large number of structure decoys can be generated via various prediction protocols, which raises the question on how to find out the most accurate prediction, *i.e.* protein structure quality assessment (QA) [89, 90]. As structure decoys produced by different protocols can be highly diverse and non-*i.i.d.*, it is critical to obtain universal embeddings for protein structures. To conclude, protein structure embeddings are crucial in many protein-related applications, but non-trivial to obtain due to limited data and/or potential bias of data distributions.

Recent advances in natural language processing (NLP) demonstrate that large-scale self-supervised pre-training models can be highly effective in various downstream tasks [74, 91]. Similar idea has been adopted to train large-scale language models for proteins, with either amino-acid sequences or multiple sequence alignments (MSAs). In [92, 93], LSTM and Transformer models are trained to predict randomly masked-out amino-acids in FASTA sequences, so as to formulate inter-residue interactions within proteins. Sturmfels *et al.* [94] propose to predict profiles derived from multiple sequence alignments, instead of randomly masked amino-acids. In [95], Transformer models are trained to predict masked-out position in multiple sequence alignments (rather than FASTA sequences), which better cooperates the co-evolution information embedded in MSAs. All these sequence-based pre-training models have been proved to be effective in learning meaningful embeddings for amino-acid types and providing critical features for secondary structure and contact predictions.

However, such sequence-based pre-training models do not utilize protein tertiary structures, which could be crucial to structure-related downstream tasks mentioned above. Additionally, the computational complexity of large-scale language models are often prohibitively high, and it usually takes weeks or even months to train such models on high-performance GPU clusters [95].

To address above issues, we propose a pre-training model for learning structure embeddings from protein tertiary structures. The model is optimized with a self-supervised loss function, which only relies on protein structures and does not require any additional supervision. Specifically, native protein structures are randomly perturbed with Gaussian noise, and the model aims at estimating the log probability's gradients over perturbed 3D coordinates. Due to intrinsic symmetries for 3D rotations and translation, the SE(3)-equivariance must be preserved in the gradient estimation. Standard SE(3)-equivariant models often involve complicated and time-consuming computation for spherical harmonics [96, 97] or regular representations [98]. In contrast, we construct SE(3)-invariant features as the pre-training model's inputs, and then reconstruct gradients over 3D coordinates with SE(3)-equivariance preserved. Such workflow, similar to [99], dramatically improves the computational efficiency without sacrificing the SE(3)-equivariance.

We demonstrate the effectiveness of our pre-training model with two downstream tasks: protein structure quality assessment and protein-protein interaction site prediction. Hierarchical structure embeddings (whole-protein, per-residue, and inter-residue) are extracted with the pre-training model, and then fed into corresponding models proposed for each downstream task as enhancement. Extensive experiments indicate that such structure embeddings consistently improve the prediction accuracy of downstream tasks.

The overall contributions of this paper are summarized as:

- We propose the first self-supervised pre-training model for protein tertiary structures, while existing models only utilize amino-acids sequences or multiple sequence alignments.
- Our pre-training model is computationally efficient, and is capable of generating informative structure embeddings at various hierarchical levels.

- We demonstrate that the prediction accuracy of downstream tasks can be consistently improved by cooperating structure embeddings provided by our pre-training model.

5.2 Related Work

5.2.1 Protein 3D structures dependent tasks

In this paper, we employ two downstream tasks which require protein three-dimensional (3D) structures to evaluate our pre-training model: protein model quality assessment (QA) and protein-protein interaction (PPI) site prediction.

Protein structure QA (estimation of model accuracy) estimates the quality of computational protein models in terms of the divergence from their native structure [100]. It aims at 1) finding the best model in a pool of protein structure prediction models, and 2) refining a model based on its estimated local quality. QA task utilizes two types of evaluation metrics: local score and global score. At the residue level, local score includes Local Distance Difference Test (LDDT) [101] and the Contact Area Difference (CAD) [102] scores. At the protein level, global score contains Global Distance Test Total Score (GDT_TS) [90], Global Distance Test High Accuracy (GDT_HA) [103], TM-score [104] and the global versions of LDDT and CAD.

Protein–protein interactions refer to the physical contacts between two or more proteins, which are crucial for the function of proteins [105, 88]. The identification of PPI Site is an efficient way to help understand the biological functions of a protein [106]. The PPI Site prediction is a residue level 2-state classification task.

5.2.2 Self-supervised Learning

The self-supervised learning method is well known for its good performance on NLP tasks by using substantial unlabeled data during the training. It does not

require explicit human guides, and also brings in flexibility [74]. An effective strategy of self-supervised training is to add certain noise to the data, then train the network to obtain the original data, which is considered as a self-recovery process. For example, masked-token prediction [91] replaces the value of tokens at multiple positions with alternate tokens and allows the network to predict back. Recently, a novel protein sequence self-supervised method called TAPE [92] uses this masked-token mechanism to train a pre-training model and achieves good performance on several sequence-based prediction tasks. However, due to the complexity of protein 3D structures, there is no structure-based pre-training method to adapt to the above 3D structure-dependent downstream tasks.

5.3 Methods

In this section, we describe how protein structures can be represented with SE(3)-invariance preserved, *i.e.* invariant to arbitrary 3D rotations and translations. Afterwards, we present our pre-training framework for protein structures, built upon energy-based models. Finally, we demonstrate how pre-trained models can be utilized in two downstream tasks: protein structure quality assessment (QA) and protein-protein interaction (PPI) site prediction.

5.3.1 SE(3)-invariant Representation of Protein Structures

Protein tertiary structures are largely determined by 3D coordinates of all the amino-acid residues' C_α atoms [107, 108]. Therefore, it is often sufficient to represent protein structures with 3D coordinates of C_α atoms. However, such coordinate-based representation depends on the overall configuration (location and orientation) of protein structures. Since rigid-body rotations and translations can be arbitrary and do

not affect protein structures, it is required that coordinated-based models must preserve the SE(3)-equivariance to capture such symmetries in the conformation space.

In this paper, we circumvent this SE(3)-equivariance restraint by introducing a SE(3)-invariant representation of protein structures. Specifically, we calculate the Euclidean distance between all the C_α atom pairs, and represent protein structures with the resulting pairwise distance matrix. Since the relative distance remains constant *w.r.t.* any 3D rotations and translations, such SE(3)-invariant representation allows much more flexible choices of subsequent models.

Formally, for a protein with amino-acid sequence of length L, we denote 3D coordinates of all the C_α atom as $\mathbf{X} \in R^{L\times 3}$, where $\mathbf{x}_i$ is the 3D coordinate of i-th residue's C_α atom. The pairwise distance matrix is denoted as $\mathbf{D} \in R^{L\times L}$, where each entry is determined by $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$. Our pre-training model is built upon pairwise distance matrices, thus the model itself does not need to be restrained to preserve the SE(3)-equivariance. Nevertheless, it is worth mentioning that it is feasible to propagate estimated gradients from the pairwise distance matrix to 3D coordinates via the chain rule, which is critical for training energy-based models, as we shall demonstrate later.

5.3.2 Self-supervised Pre-training

In order to extract informative protein and per-residue embeddings, we propose a pre-training model to approximate the data distribution of protein tertiary structures. The intrinsic motivation is that if the underlying data distribution is well approximated, then this pre-training model must have captured the critical information embedded in protein structures, which could be quite beneficial for various downstream tasks.

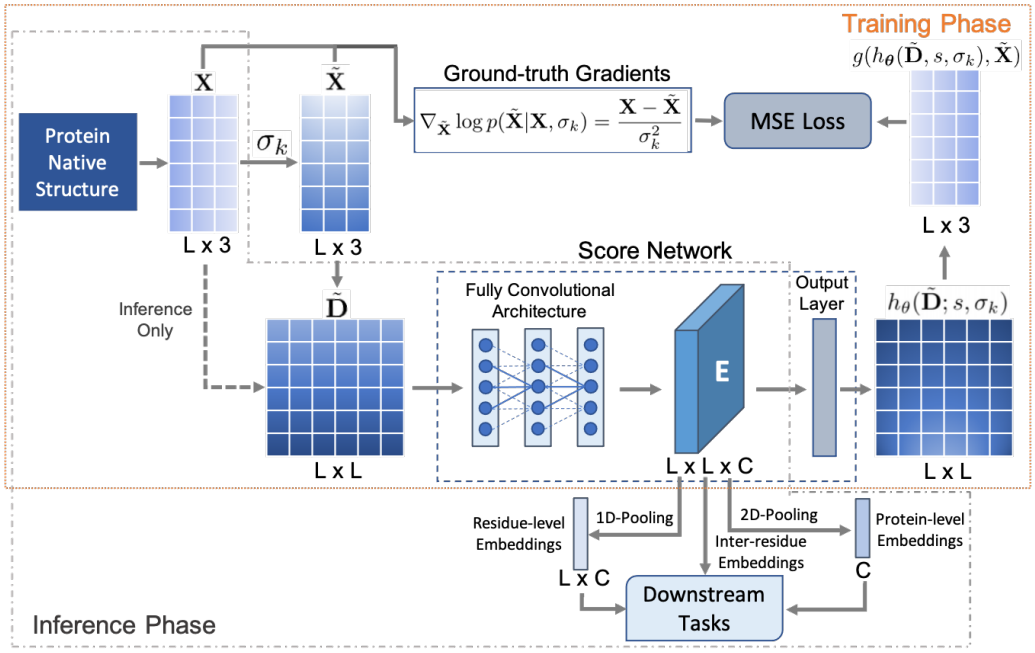


Figure 5.1. The workflow of the pre-training process. First, we extract C_α atoms' 3D coordinates, which are denoted as X, and perturb it with various levels of random noise to get perturbed 3D coordinates $\tilde{X}$. Then we compute the distance matrix $\tilde{D}$, which is further fed into the score network to predict the corresponding gradients. It is then transformed into the estimated gradients over perturbed 3D coordinates. We calculate the MSE loss between the estimated and ground-truth gradients as the pre-training signals to back-propagate to the score network. For the inference phase, we transfer the 3D coordinates X to the distance matrix D without perturbation, and extract the feature matrix E for the downstream tasks..

In [109], Song *et al.* propose to train an energy-based model via denoising score matching [110] for image generation. Original images are perturbed with Gaussian noise of different scales, and the network is trained to estimate the log probability's gradients over perturbed images. Although pairwise distance matrices, as SE(3)-invariant representations of protein structures, can also be viewed as 2D images, it is unreasonable to directly perturb distance matrices with random noise. The key difference lies in that for the image generation task, every randomly perturbed image is valid, so that the perturbed data distribution is still well defined. However, not all

$L \times L$ real-valued matrices are valid distance matrices, *i.e.* there may not exist a 3D structure satisfying the randomly perturbed distance matrix.

To tackle this issue, instead of applying random perturbation on distance matrices, we propose to firstly add Gaussian noise on 3D coordinates of all the C_α atoms, and derive the corresponding distance matrix as perturbed inputs. The score network is then trained to estimate gradients over perturbed distance matrices. Both inputs and outputs of the score network are invariant to 3D rotations and translations, so the score network can be instantiated by any convolutional neural networks. Since the random perturbation is performed over 3D coordinates, we only have closed-form ground-truth gradients over 3D coordinates. Therefore, we also need to propagate estimated gradients from distance matrices to 3D coordinates, which is made possible via the chain rule.

Formally, we choose a series of standard deviations for Gaussian noise, $\sigma_1 > \sigma_2 > \cdots > \sigma_K$, where K is the total number of random noise levels. We denote the native protein structure as $\mathbf{X}$, as represented by all the C_α atoms' 3D coordinates, and its perturbed counterpart as $\tilde{\mathbf{X}} \sim p(\tilde{\mathbf{X}}|\mathbf{X}, \sigma_k)$, which is given by:

$$\tilde{\mathbf{X}} := \mathbf{X} + \mathbf{Z},\ \mathbf{Z} \sim \mathcal{N}\left(\mathbf{0}, \sigma_k^2 \mathbf{I}\right) \tag{5.1}$$

where σ_k is selected as the random noise's standard deviation. The perturbed data distribution's log probability's gradients over perturbed 3D coordinates have a closed-form solution:

$$\nabla_{\tilde{\mathbf{X}}} \log p(\tilde{\mathbf{X}}|\mathbf{X}, \sigma_k) = \frac{\mathbf{X} - \tilde{\mathbf{X}}}{\sigma_k^2} \tag{5.2}$$

which can be easily derived from the multivariate Gaussian distribution's probability density function.

We denote the pairwise distance matrix corresponding to the perturbed 3D coordinates as $\tilde{\mathbf{D}}$, where $\tilde{d}_{ij} = \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|_2$. This perturbed distance matrix is then

fed into the score network, which consists of multiple residual convolutional blocks. Similar to [109], conditional batch normalization is employed to explicitly let the score network be aware of the random noise's standard deviation for generating the current perturbed input. The detailed network architecture is presented in Section 5.4.1.3. The score network is trained to estimate the log probability's gradients over the elementwise squared perturbed distance matrix:

$$\mathbf{H} := h_{\boldsymbol{\theta}}(\tilde{\mathbf{D}}, s, \sigma_k),\ h_{ij} \approx \nabla_{\tilde{d}_{ij}^2} \log p(\tilde{\mathbf{X}}|\mathbf{X}, \sigma_k) \tag{5.3}$$

where the amino-acid sequence s is also used as the inputs of the score network. As discussed above, it is non-trivial to derive closed-form ground-truth gradients for the distance matrices. Hence, we apply the chain rule to propagate the estimated gradients to perturbed 3D coordinates:

$$\begin{aligned} \mathbf{G} = \begin{bmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_L \end{bmatrix},\ \mathbf{g}_i &= \sum\nolimits_{j=1}^{L} 2\left(h_{ij} + h_{ji}\right)\left(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\right) \\ &\approx \nabla_{\tilde{\mathbf{x}}_i} \log p(\tilde{\mathbf{X}}|\mathbf{X}, \sigma_k) \end{aligned} \tag{5.4}$$

where the last term can be explicitly calculated by Eq. (5.2). For simplicity, we denote the above gradient propagation process as $\mathbf{G} = g(\mathbf{H}, \tilde{\mathbf{X}}) = g(h_{\boldsymbol{\theta}}(\tilde{\mathbf{D}}, s, \sigma_k), \tilde{\mathbf{X}})$.

So far, we have presented the log probability's ground-truth gradients over perturbed 3D coordinates, as well as the score network's estimation. The self-supervised loss function is given by:

$$\begin{aligned} Loss = &\frac{1}{2NK} \sum_{\mathbf{X}\in\mathcal{X}} \sum_{k=1}^{K} \sigma_k^2 \cdot E_{\tilde{\mathbf{X}}\sim\mathcal{N}(\mathbf{X}|\sigma_k^2\mathbf{I})} \\ &\left\| g(h_{\boldsymbol{\theta}}(\tilde{\mathbf{D}}, s, \sigma_k), \tilde{\mathbf{X}}) - \frac{\mathbf{X} - \tilde{\mathbf{X}}}{\sigma_k^2} \right\|_F^2 \end{aligned} \tag{5.5}$$

where $\mathcal{X}$ is the set of all the native protein structures, and $N = |\mathcal{X}|$ is its cardinality. The above loss function measures the difference between the ground-truth and the

estimated gradients for all the K random noise levels. Each level's loss is re-weighted by the corresponding standard deviation σ_k, so that each level approximately has an equal contribution to the overall loss function. By minimizing this loss function, the score network's estimated gradients approximately match ground-truth ones, thus the underlying data distribution of native protein structures is roughly parameterized by the score network. The overall training workflow is illustrated in Figure 5.1.

Once the pre-training model is sufficiently optimized, we may utilize it to extract structure embeddings for novel protein structures. Recall that the score network adopts the 2D convolutional network as the backbone architecture. For any specific protein structure, we calculate the pairwise distance matrix for all the C_α atoms, and feed it into the pre-training model. The final feature maps (next to estimated gradients) of size $L \times L \times C$ are then extracted, where C is the number of feature map channels. Such feature maps can be viewed as inter-residue structure embeddings, each of dimension C. Furthermore, by applying 1D and 2D global pooling, we obtain C-dimensional per-residue and whole-protein structure embeddings. To wrap up, during the inference phase (as depicted in Figure 5.1), we can extract whole-protein, per-residue, and inter-residue structure embeddings as additional inputs to downstream tasks.

5.3.3 Pre-training Model for Downstream Tasks

Here, we take two downstream tasks as examples, to demonstrate how structure embeddings produced by the pre-training model can be utilized to boost the prediction accuracy of downstream tasks.

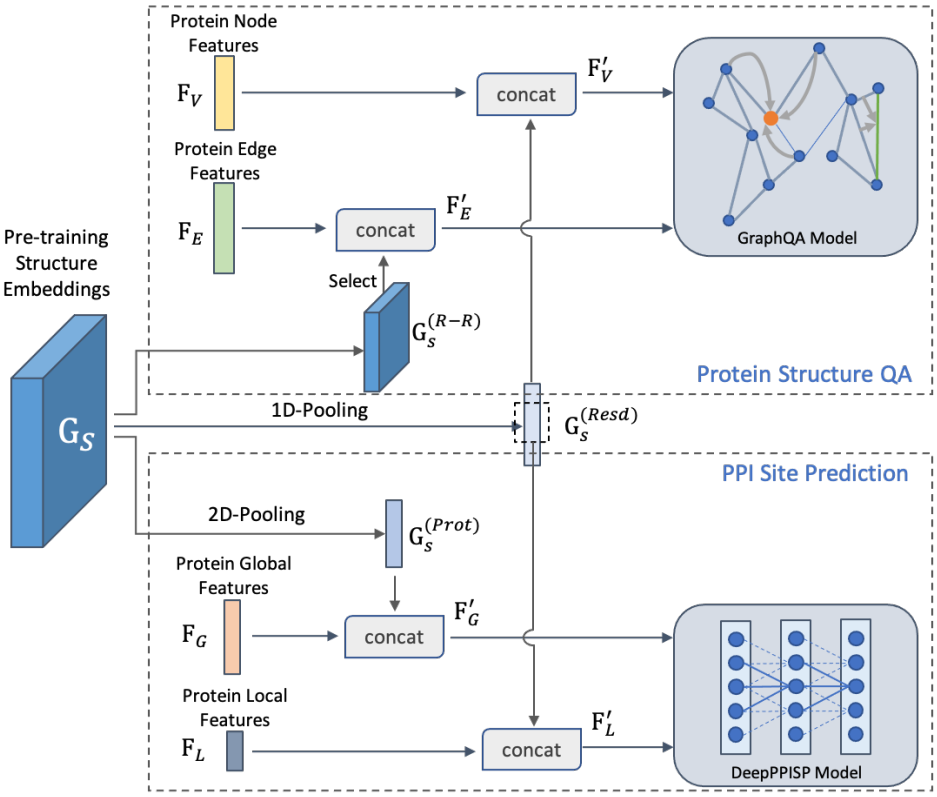


Figure 5.2. To align with the input feature vectors of the two downstream tasks, we conduct multiple operations on the embeddings generated by our pre-training model: 1) pre-trained edge embeddings is obtained by using the same selecting methods as GraphQA; 2) G_S^{Resd} is computed by 1D average pooling as the pre-trained node feature on QA task; 3) On the basis of G_S^{Resd}, we use the same window clipping operation as the DeepPPISP to obtain the enhanced local feature on i-th residue. 4) We perform 2D average pooling on G_S to get G_S^{Prot} as the pre-trained global feature for PPI Site prediction task..

5.3.3.1 Protein Structure Quality Assessment

Due to the randomness in the initialization and optimization process, multiple structure decoys are generated as the candidates for the same amino-acid sequence for most protein structure prediction methods [111, 112]. Protein structure quality assessment (QA) aims at identifying the best predicted structure among all the candidates, which is one of the indispensable modules in protein structure prediction.

In [90], the authors propose GraphQA to formulate the protein structure as a graph, where the nodes are amino-acid residues and the edges are inter-residue interactions. To simultaneously consider the sequential and geometric structure, GraphQA builds the edges for both sequential-adjacent and spatial-neighboring residue pairs. The model consists of multiple message passing operations [113] to gradually update the node embeddings and predict both local and global lDDT scores [101]. Empirical evaluation results indicate that GraphQA achieves similar prediction accuracy to state-of-art-methods for quality assessment, despites the simplicity of the node/edge features being used.

Here, we employ our pre-training model to extract structure embeddings to further enhance the node and edge features of GraphQA. Specifically, we feed the structure decoy which needs to be assessed into our pre-training model (without random perturbation), and obtain the resulting structure embeddings $\mathbf{G}_S$. Since each spatial location in the feature map corresponds to a pair of residues, we enhance the edge features by selecting feature vectors at the corresponding locations. Similarly, the node features can be enhanced by concatenating the 1D-Pooling results of structure embeddings $\mathbf{G}_S$. The GraphQA model then takes such enhanced node and edge features as input for local and global lDDT prediction. The overall workflow is depicted in the upper part of Figure 5.2.

5.3.3.2 Protein-protein Interaction Site Prediction

Protein-protein interaction models predict the physical contacts between two or more proteins, which play a vital role in various biological processes [114, 115]. To better understand how different proteins interact with each other, the first step is to identify which amino-acid residues in each protein are actually involved in the interaction. Formally, we follow [88] to define an amino-acid as a PPI site if its

absolute solvent accessibility before and after the protein binding is smaller than 1 $\mathring{A}^2$. Thus, the PPI site prediction task can be viewed as a pre-residue binary classification problem. In [88], the authors propose DeepPPISP as an end-to-end framework, which integrates both local contextual and global sequence features for PPI site prediction. Concretely, local features are extracted from a fixed-size sliding windows centered at each amino-acid residue to capture local patterns, while global features are extracted via an 1-dimensional convolutional network. After that, local and global features are concatenated and used by the subsequent classification sub-network for per-residue classification.

Similarly, our pre-training model can be used as a plug-n-play module to enhance both local and global features used in the DeepPPISP model. For each training sample used in PPI site prediction, we encode protein structures with our pre-training model to calculate the corresponding structure embeddings. The additional global features are obtained via applying the 2D-Pooling over structure embeddings. As for local features, per-residue structure embeddings can be computed as 1D-Pooling results of full-size structure embeddings. Such embeddings are then grouped by the same sliding window to generate additional local contextual features to describe each amino-acid residue. By concatenating all the original/additional local and global features, the DeepPPISP model can be trained with an enhanced feature set for PPI site prediction.

5.3.3.3 Summary

To wrap up, we have demonstrated how our pre-training model can be utilized to produce structure embeddings at various hierarchical levels. As long as the downstream task relies on structure-based features of proteins, it should always be

beneficial to include our structure embeddings to further enhance its feature representation. Potential application scenarios include protein fold classification [116] and structure-based drug design [86].

5.4 Experiments

5.4.1 Experiments setup

5.4.1.1 Datasets

For the pre-training model, we obtain native protein structures from the RCSB-PDB database (released on 01/05/2021) [117], which includes over 170 thousands unlabeled protein tertiary structures. The RCSB-PDB database is somewhat redundant, where identical or highly-similar amino-acid sequences may correspond to multiple protein structures. Therefore, we adopt the official sequence clustering results, BC-30 and BC-100, to filter-out the redundant sequences with at least 30% or 100% sequence identity, respectively. After removing overlap proteins with valid and test data in downstream tasks, the BC-100 dataset contains 73,585 proteins, among which 58,868 are used as the training set, 7,357 as the validation set, and the remaining ones are test set. The BC-30 dataset consists of 29,242 proteins. Within them, 23,394 proteins are used as training set, 2,923 as the validation set, and 2,925 proteins are used for testing.

For the protein QA prediction task, we use the dataset published by GraphQA [90]. CASP9-CASP12 datasets contain 85k decoys, which are randomly split into a training set (˜270 targets) and a validation set (˜50 targets). CASP13 dataset contains ˜14k decoys (˜72 targets) in the test set.

For the PPI site prediction task, we use the processed data from DeepPPISP [88], *i.e.* Dset_186 of 186 proteins, Dset_72 of 72 proteins [118] and PDBset_164

of 164 proteins [119]. DeepPPISP removes two proteins since they do not have the related protein DSSP files [17], which is one of the input features used in the method. DeepPPISP integrates three datasets to a fused dataset to ensure that the training and test set are from an identical distribution. We download the training, validation, and test data list from [88]. There are 300 proteins in the training set, 50 proteins for independent validation set, and 70 proteins in the test set.

5.4.1.2 Input features

In addition to the distance matrix described in Section 5.3.1, we also encode the protein-specific information as the input features of the score network, which include protein sequence one-hot feature and positional encoding [74]:

One-hot feature: Each amino acid in the protein sequence is represented by a one-hot vector with the length as 20, which refers to 20 kinds of amino acids. Thus, for a protein sequence of length L, we have a $L \times 20$ one-hot encoding feature vector. We repeat the process row-wisely and column-wisely to obtain a stacked $L \times L \times 40$ feature map.

Positional encoding: Following [74], we adopt the positional encoding scheme to encode each residue's relative position in the protein sequence. To obtain a positional encoding feature map of size $L \times L \times d_{model}$, we first encode the sequence into a $L \times \frac{1}{2}d_{model}$ matrix follows:

$$\begin{aligned} PE_{(pos,2i)} &= sin[i/pow(L_{max}, \frac{4i}{d_{model}})], \\ PE_{(pos,2i+1)} &= cos[i/pow(L_{max}, \frac{4i}{d_{model}})], \end{aligned} \tag{5.6}$$

where *pos* is the residue's position, and i denotes the dimension ($i \in \{0, 1 \cdots \lfloor \frac{1}{4} d_{model} \rfloor\}$); $L_{max} = 700$ is the maximal length of protein sequences. We then repeat the matrix PE row-wisely to form a 2D positional feature map.

5.4.1.3 Network architecture and learning hyper-parameters

Our score network for pre-training adopts the fully-convolutional neural networks architecture, which consists of 32 residual blocks with dilation convolution. To reduce the computational overhead, we apply the bottleneck mechanism [120] on each residual unit. We also use conditional batch normalization [109] to take random noise's standard deviation level into consideration. The number of hidden layers' channels k is set to 64. We use a batch size of 32 for training and validation, and randomly crop the input feature maps with size 32 for data augmentation. The positional encodings' dimension is set to $d_{model} = 24$. We construct random noise's standard deviations for $K = 32$ levels, which ranges from 0.01 to 10.0. When $\sigma_1 = 10.0$, the conformation space can be sufficiently explored, while $\sigma_K = 0.01$ indicates trivial perturbation is introduced to the native structures.

For the optimization, we apply a constant learning rate of 0.0001 and use Adam [64] as the optimizer for our pre-training model. After training 50 epochs, we select the optimal checkpoint based on the validation loss, and then use it for the upcoming structure embeddings (G_S) generation.

[1] To make a fair comparison, all the settings and data are the same with the original papers when we run baseline, sequence embeddings, and structural embeddings experiments. The evaluation metrics are originally used in GraphQA and DeepPPISP.

Method [1]	GDT_TS					CAD		LDDT	
	RMSE	R	R_{target}	z	FRL_5	ρ	ρ_{decoy}	ρ	ρ_{decoy}
w/o pre-trained embeddings	0.201	0.793	0.751	1.026	0.045	0.637	0.390	0.774	0.510
w/ sequence embeddings	0.158	0.799	0.772	1.101	0.037	0.624	0.387	0.754	0.502
w/ BC-30 embeddings (ours)	0.149	0.818	0.775	1.272	0.035	0.649	0.415	0.782	0.530
w/ BC-100 embeddings (ours)	**0.133**	**0.848**	**0.787**	**1.345**	**0.031**	**0.667**	**0.424**	**0.800**	**0.534**

Table 5.1. Results on global and local QA prediction task using GraphQA prediction model

Method [1]	ACC	Precision	Recall	F-measure	MCC
w/o pre-trained embeddings	0.589±0.012	0.270±0.006	0.623±0.018	0.377±0.004	0.163±0.006
w/ sequence embeddings	0.592±0.036	0.274±0.009	**0.635±0.058**	0.382±0.003	0.174±0.005
w/ BC-30 embeddings (ours)	0.614±0.016	0.280±0.005	0.604±0.026	0.382±0.001	0.177±0.002
w/ BC-100 embeddings (ours)	**0.621±0.029**	**0.285±0.010**	0.601±0.052	**0.386±0.003**	**0.185±0.004**

Table 5.2. Results on PPI Site prediction task using DeepPPISP prediction model

5.4.2 Results

Table 5.1 shows the comparison performance on protein QA downstream supervised task for CASP13 dataset. Other than evaluating the effectiveness of our method by running experiments with and without our pre-training model, we also compare the performance of protein sequence-based embedding. GraphQA is utilized as the baseline model, and we follow [92] to generate TAPE's sequence-based embeddings. Table 5.1-GDT_TS shows the results of various evaluation metrics for global quality predictions *w.r.t.* GDT_TS. For $RMSE$ and FRL_5, lower is better; for R, R_{target}, and z, higher is better. The results demonstrate that with the embeddings generated by our pre-training model, GraphQA is more capable than all other methods, including using the original features and adding sequence-based embeddings at ranking decoys on their overall quality.

The performance of local quality predictions *w.r.t.* the ground-truth CAD and LDDT scores are also reported in Table 5.1, higher is better. As observed, our pre-

training method further improves the performance at the local level, which indicates the high quality of our embeddings at the local (residue) level, as well as the ability of distinguishing the correctly predicted parts of the protein chain. In consequence, the embeddings extracted by our pre-training model can make the prediction network capture more complex information and long-range dependencies between residues compared with the original features. Please note that the results of adding sequence embedding on local scores are worse than the baseline. One possible reason is that local QA task is more dependent on inter-residue (edge) information, while TAPE does not contain such information. Moreover, adding a large number of dimensions' node features (768 dimensions of TAPE) makes the original network more difficult to train.

We implement the experiments precisely according to the experimental settings in GraphQA [90], including data-splitting, network hyper-parameters, and training strategy.

Table 5.2 shows the results of DeepPPISP model training with and without the embeddings generated by our pre-training model, and we introduce the TAPE embeddings for comparison as well. Since DeePPPISP does not provide a seed for data loading, we repeat the experiment five times to get the mean and standard deviation to eliminate the randomness and verify the robustness. Although the recall of our method is lower than the performance of baselines, the scores of all other assessment metrics are the highest. It is noteworthy that the PPI Site prediction training problem is imbalanced, thus the downstream task is usually more concentrated on the the performance of MCC and F-measure [121], and DeepPPISP uses F-measure to select the best validation model. Compared with QA task, PPI task has relatively balanced dependence on sequence information and structure information. Thus, it is reasonable that TAPE performs better than the baseline model which only utilizes the original

features. Moreover, our structure embeddings is able to achieve better performance by exploring the structure information.

In addition, we conduct experiments with pre-training on a smaller dataset, named the BC-30 filtered dataset, to confirm the effectiveness of proposed method. As shown in Table 5.1 and 5.2, although the data involved in pre-training is streamlined, it consistently performs well on downstream tasks. The results indicate that even pre-training on a smaller dataset, our model can still provide high-quality local and global embeddings for downstream tasks.

5.5 Conclusion

In this work, we propose a self-supervised pre-training model for protein structure. To the best of our knowledge, this is the first attempt to construct and evaluate self-supervised learning on protein 3D structures. In addition, our method can be easily applied to various downstream models. It is empirically demonstrated that our pre-training model can generate high quality structure embeddings for downstream tasks. Recent pre-training strategies mainly focus on the protein sequence dataset since it is easier to obtain and contains huge amount of data. However, even the dataset used for pre-training protein 3D structure is not as large as protein sequence dataset, we argue that the 3D structure contains more information than the sequence. In order to fully utilize the available protein data, our next move is to integrate the 3D structure pre-training strategy with a sequence-based pre-training method to acquire sufficient protein information.

CHAPTER 6

Conclusions

This thesis aims at developing effective deep learning techniques for protein property and structure prediction tasks. We investigate several typical type of protein prediction tasks including protein secondary structure, solvent accessibility, backbone dihedral angles, protein structure quality assessment, and protein-protein interaction site prediction.

We have demonstrated, both in theory and practice, our deep learning approaches and formed effective and efficient solutions with clear performance gains in extensive experiments on protein property and structure prediction tasks. Specifically, we have developed the following methods:

Protein Ensemble Learning With Atrous Spatial Pyramid Networks For Secondary Structure Prediction: We propose an efficient method to investigate the problem of protein secondary structure prediction. A novel Conditionally Parameterized Convolutional network (CondGCNN) is proposed, which utilize the power of both CondConv and GCNN, and we leverage an ensemble encoder to combine the capabilities of both LSTM and CondGCNN to encode protein sequences to obtain better sequential features from proteins. In addition, due to the similarity between the image segmentation problem and the secondary structure prediction problem, I propose an ASP network (Atrous Spatial Pyramid Pooling (ASPP) based network) as the secondary structure generator in our proposed framework. Experimental results show that the proposed method can achieve higher performance than state-of-the-art methods on CB513, CASP11 and CASP12 datasets.

Bagging MSA Learning: Enhancing Low-quality Pssm With Deep Learning For Accurate Protein Structure Property Prediction: A novel pipeline to enhance features for proteins with low-quality homologous features. The model adopt a convolutional network to capture local context features and bidirectional-LSTM for long-term dependencies, and integrate them under an unsupervised framework. Structure property prediction models are then built upon such enhanced features for more accurate predictions. Empirical evaluation of CB513, CASP11, and CASP12 datasets indicate that the unsupervised enhancing scheme indeed generates more informative features for structure property prediction. In the future, we shall attempt to combine the semi-supervised techniques [122] to further improve the accuracy.

WeightAln: Weighted Homologous Alignment For Protein Structure Property: We introduce a novel Multiple Sequence Alignment (MSA) weights learning framework, WeightAln, which generates learnable MSA weights for protein prediction tasks using attention-based deep learning techniques in this chapter. Extensive experiments on three protein structure property prediction tasks, secondary structure, solvent accessibility, and backbone dihedral angles prediction, sufficiently demonstrate the effectiveness of the method. In the future, we shall conduct experiments on higher dimensional protein prediction problems, such as contact map prediction and distance prediction.

Self-supervised Pre-training for Protein Embeddings Using Tertiary Structures: We propose a self-supervised pre-training model for learning structure embeddings from protein tertiary structures. Native protein structures are perturbed with random noise, and the pre-training model aims at estimating gradients over perturbed 3D structures. I demonstrate the effectiveness of our pre-training model on two downstream tasks, protein structure quality assessment (QA) and protein-protein

interaction (PPI) site prediction. Hierarchical structure embeddings are extracted to enhance corresponding prediction models. Extensive experiments indicate that such structure embeddings consistently improve the prediction accuracy for both downstream tasks.

In order to fully utilize the available protein data, our next move is to integrate the 3D structure pre-training strategy with a sequence-based pre-training method to acquire sufficient protein information. In addition, it might be interesting to further investigate bonds between protein data and graph neural network (GNN) model [123, 124], which might bring in many novel methods to further accelerate protein related research. The techniques described in this paper might also be extendable and beneficial to deep learning and machine learning research.

REFERENCES

[1] PDBsum, "Pdbsum website," http://www.ebi.ac.uk/thornton-srv/databases/cgi-bin/pdbsum/GetPage.pl?pdbcode=index.html, 2013.

[2] T. E. Creighton, *Proteins: structures and molecular properties*. Macmillan, 1993.

[3] K. A. Dill and J. L. MacCallum, "The protein-folding problem, 50 years on," *science*, vol. 338, no. 6110, pp. 1042–1046, 2012.

[4] S. K. Sønderby and O. Winther, "Protein secondary structure prediction with long short term memory networks," *arXiv preprint arXiv:1412.7828*, 2014.

[5] J. Singh, J. Hanson, R. Heffernan, K. Paliwal, Y. Yang, and Y. Zhou, "Detecting proline and non-proline cis isomers in protein structures from sequences using deep residual ensemble learning," *Journal of chemical information and modeling*, vol. 58, no. 9, pp. 2033–2042, 2018.

[6] Y. Guo, B. Wang, W. Li, and B. Yang, "Protein secondary structure prediction improved by recurrent neural networks integrated with two-dimensional convolutional neural networks," *Journal of bioinformatics and computational biology*, vol. 16, no. 05, p. 1850021, 2018.

[7] Y. Guo, J. Wu, H. Ma, S. Wang, and J. Huang, "Protein ensemble learning with atrous spatial pyramid networks for secondary structure prediction," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2020, pp. 17–22.

[8] ——, "Bagging msa learning: Enhancing low-quality pssm with deep learning for accurate protein structure property prediction," in *International Conference on Research in Computational Molecular Biology*. Springer, 2020, pp. 88–103.

[9] ——, "Comprehensive study on enhancing low-quality position-specific scoring matrix with deep learning for accurate protein structure property prediction: Using bagging multiple sequence alignment learning," *Journal of Computational Biology*, 2021.

[10] ——, "Eptool: A new enhancing pssm tool for protein secondary structure prediction," *Journal of Computational Biology*, vol. 28, no. 4, pp. 362–364, 2021.

[11] Y. Guo, J. Wu, H. Ma, J. Yang, X. Zhu, and J. Huang, "Weightaln: Weighted homologous alignment for protein structure property prediction," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2020, pp. 72–75.

[12] Y. Guo, J. Wu, H. Ma, S. Wang, and J. Huang, "Deep ensemble learning with atrous spatial pyramid networks for protein secondary structure prediction," *Biomolecules*, vol. 12, no. 6, p. 774, 2022.

[13] Y. Guo, J. Wu, H. Ma, and J. Huang, "Self-supervised pre-training for protein embeddings using tertiary structures," 2022.

[14] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[15] J. Zhou and O. G. Troyanskaya, "Deep supervised and convolutional generative stochastic network for protein secondary structure prediction," *arXiv preprint arXiv:1403.1347*, 2014.

[16] L. Pauling, R. B. Corey, and H. R. Branson, "The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain," *Proceedings of the National Academy of Sciences*, vol. 37, no. 4, pp. 205–211, 1951.

[17] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers: Original Research on Biomolecules*, vol. 22, no. 12, pp. 2577–2637, 1983.

[18] S. Wang, J. Peng, J. Ma, and J. Xu, "Protein secondary structure prediction using deep convolutional neural fields," *Scientific reports*, vol. 6, p. 18962, 2016.

[19] Y. Guo, W. Li, B. Wang, H. Liu, and D. Zhou, "Deepaclstm: deep asymmetric convolutional long short-term memory neural models for protein secondary structure prediction," *BMC bioinformatics*, vol. 20, no. 1, p. 341, 2019.

[20] H. M. Berman, T. Battistuz, T. N. Bhat, W. F. Bluhm, P. E. Bourne, K. Burkhardt, Z. Feng, G. L. Gilliland, L. Iype, S. Jain, *et al.*, "The protein data bank," *Acta Crystallographica Section D: Biological Crystallography*, vol. 58, no. 6, pp. 899–907, 2002.

[21] R. A. Laskowski, J. Jabłońska, L. Pravda, R. S. Vařeková, and J. M. Thornton, "Pdbsum: Structural summaries of pdb entries," *Protein Science*, vol. 27, no. 1, pp. 129–134, 2018.

[22] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-scnn: Gated shape cnns for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5229–5238.

[23] H. Wu, J. Zhang, K. Huang, K. Liang, and Y. Yu, "Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation," *arXiv preprint arXiv:1903.11816*, 2019.

[24] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation,"

in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[25] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3684–3692.

[26] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 933–941.

[27] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "Condconv: Conditionally parameterized convolutions for efficient inference," in *Advances in Neural Information Processing Systems*, 2019, pp. 1307–1318.

[28] M. Lin, Q. Chen, and S. Yan, "Network in network international conference on learning representations (iclr'14)," 2014.

[29] G. Wang and R. L. Dunbrack Jr, "Pisces: a protein sequence culling server," *Bioinformatics*, vol. 19, no. 12, pp. 1589–1591, 2003.

[30] B. E. Suzek, H. Huang, P. McGarvey, R. Mazumder, and C. H. Wu, "Uniref: comprehensive and non-redundant uniprot reference clusters," *Bioinformatics*, vol. 23, no. 10, pp. 1282–1288, 2007.

[31] C. Fang, Y. Shang, and D. Xu, "Mufold-ss: New deep inception-inside-inception networks for protein secondary structure prediction," *Proteins: Structure, Function, and Bioinformatics*, vol. 86, no. 5, pp. 592–598, 2018.

[32] W. An, Y. Guo, Y. Bian, H. Ma, J. Yang, C. Li, and J. Huang, "Modna: motif-oriented pre-training for dna language model," in *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 2022, pp. 1–5.

[33] R. Heffernan, K. Paliwal, J. Lyons, A. Dehzangi, A. Sharma, J. Wang, A. Sattar, Y. Yang, and Y. Zhou, "Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning," *Scientific reports*, vol. 5, p. 11476, 2015.

[34] G. D. Stormo, T. D. Schneider, L. Gold, and A. Ehrenfeucht, "Use of the 'perceptron'algorithm to distinguish translational initiation sites in e. coli," *Nucleic acids research*, vol. 10, no. 9, pp. 2997–3011, 1982.

[35] D. T. Jones, "Protein secondary structure prediction based on position-specific scoring matrices," *Journal of molecular biology*, vol. 292, no. 2, pp. 195–202, 1999.

[36] Y. Gao, S. Wang, M. Deng, and J. Xu, "Raptorx-angle: real-value prediction of protein backbone dihedral angles through a hybrid method of clustering and deep learning," *BMC bioinformatics*, vol. 19, no. 4, p. 100, 2018.

[37] L. Wang and T. Jiang, "On the complexity of multiple sequence alignment," *Journal of computational biology*, vol. 1, no. 4, pp. 337–348, 1994.

[38] M. Mirdita, L. von den Driesch, C. Galiez, M. J. Martin, J. Söding, and M. Steinegger, "Uniclust databases of clustered and deeply annotated protein sequences and alignments," *Nucleic acids research*, vol. 45, no. D1, pp. D170–D176, 2017.

[39] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa, and M. Weigt, "Direct-coupling analysis of residue coevolution captures native contacts across many protein families," *Proceedings of the National Academy of Sciences*, vol. 108, no. 49, pp. E1293–E1301, 2011.

[40] Z. Wang and J. Xu, "Predicting protein contact map using evolutionary and physical constraints by integer programming," *Bioinformatics*, vol. 29, no. 13, pp. i266–i273, 2013.

[41] F. Teichert, J. Minning, U. Bastolla, and M. Porto, "High quality protein sequence alignment by combining structural profile prediction and profile alignment using sabertooth," *BMC bioinformatics*, vol. 11, no. 1, p. 251, 2010.

[42] M. Remmert, A. Biegert, A. Hauser, and J. Söding, "Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment," *Nature methods*, vol. 9, no. 2, pp. 173–175, 2012.

[43] T. J. Wheeler and S. R. Eddy, "nhmmer: Dna homology search with profile hmms," *Bioinformatics*, vol. 29, no. 19, pp. 2487–2489, 2013.

[44] A. O'Driscoll, V. Belogrudov, J. Carroll, K. Kropp, P. Walsh, P. Ghazal, and R. D. Sleator, "Hblast: Parallelised sequence similarity–a hadoop mapreducable basic local alignment search tool," *Journal of Biomedical Informatics*, vol. 54, pp. 58–64, 2015.

[45] R. Heffernan, Y. Yang, K. Paliwal, and Y. Zhou, "Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility," *Bioinformatics*, vol. 33, no. 18, pp. 2842–2849, 2017.

[46] F. Oteri, F. Nadalin, R. Champeimont, and A. Carbone, "Bis2analyzer: a server for co-evolution analysis of conserved protein families," *Nucleic acids research*, vol. 45, no. W1, pp. W307–W314, 2017.

[47] F. Ju, J. Zhu, G. Wei, Q. Zhang, S. Sun, and D. Bu, "Seq-setnet: Exploring sequence sets for inferring structures," *arXiv preprint arXiv:1906.11196*, 2019.

[48] X. Ye, G. Wang, and S. F. Altschul, "An assessment of substitution scores for protein profile–profile comparison," *Bioinformatics*, vol. 27, no. 24, pp. 3356–3363, 2011.

[49] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped blast and psi-blast: a new generation of protein database search programs," *Nucleic acids research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[50] Z. Wang, F. Zhao, J. Peng, and J. Xu, "Protein 8-class secondary structure prediction using conditional neural fields," in *2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2010, pp. 109–114.

[51] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," *arXiv preprint arXiv:1705.03122*, 2017.

[52] J. Gehring, M. Auli, D. Grangier, and Y. N. Dauphin, "A convolutional encoder model for neural machine translation," *arXiv preprint arXiv:1611.02344*, 2016.

[53] J. Zhou, H. Wang, Z. Zhao, R. Xu, and Q. Lu, "Cnnh_pss: protein 8-class secondary structure prediction by convolutional neural network with highway," *BMC bioinformatics*, vol. 19, no. 4, pp. 99–109, 2018.

[54] C. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 69–78.

[55] J. Hanson, K. Paliwal, T. Litfin, Y. Yang, and Y. Zhou, "Accurate prediction of protein contact maps by coupling residual two-dimensional bidirectional long short-term memory with convolutional neural networks," *Bioinformatics*, vol. 34, no. 23, pp. 4039–4045, 2018.

[56] B. Zhang, J. Li, and Q. Lü, "Prediction of 8-state protein secondary structures by a novel deep learning architecture," *BMC bioinformatics*, vol. 19, no. 1, p. 293, 2018.

[57] D. M. Allen, "Mean square error of prediction as a criterion for selecting variables," *Technometrics*, vol. 13, no. 3, pp. 469–475, 1971.

[58] C. A. Andersen, H. Bohr, and S. Brunak, "Protein secondary structure: category assignment and predictability," *FEBS letters*, vol. 507, no. 1, pp. 6–10, 2001.

[59] S. Penel, R. G. Morrison, P. D. Dobson, R. J. Mortishire-Smith, and A. J. Doig, "Length preferences and periodicity in β-strands. antiparallel edge β-sheets are more likely to finish in non-hydrogen bonded rings," *Protein engineering*, vol. 16, no. 12, pp. 957–961, 2003.

[60] A. Bairoch, R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, *et al.*, "The universal protein resource (uniprot)," *Nucleic acids research*, vol. 33, no. suppl_1, pp. D154–D159, 2005.

[61] W. G. Touw, C. Baakman, J. Black, T. A. Te Beek, E. Krieger, R. P. Joosten, and G. Vriend, "A series of pdb-related databanks for everyday needs," *Nucleic acids research*, vol. 43, no. D1, pp. D364–D368, 2015.

[62] S. Eddy, "Hmmer user's guide," *Department of Genetics, Washington University School of Medicine*, vol. 2, no. 1, p. 13, 1992.

[63] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[65] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[66] M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, *et al.*, "Accurate prediction of protein structures and interactions using a three-track neural network," *Science*, vol. 373, no. 6557, pp. 871–876, 2021.

[67] K. Sato, M. Akiyama, and Y. Sakakibara, "Rna secondary structure prediction using deep learning with thermodynamic integration," *Nature communications*, vol. 12, no. 1, pp. 1–9, 2021.

[68] Q. Wang, B. Wang, Z. Xu, J. Wu, P. Zhao, Z. Li, S. Wang, J. Huang, and S. Cui, "Pssm-distil: Protein secondary structure prediction (pssp) on low-quality pssm by knowledge distillation with contrastive learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 617–625.

[69] J. J. Almagro Armenteros, C. K. Sønderby, S. K. Sønderby, H. Nielsen, and O. Winther, "Deeploc: prediction of protein subcellular localization using deep learning," *Bioinformatics*, vol. 33, no. 21, pp. 3387–3395, 2017.

[70] F. Gabler, S.-Z. Nam, S. Till, M. Mirdita, M. Steinegger, J. Söding, A. N. Lupas, and V. Alva, "Protein sequence analysis using the mpi bioinformatics toolkit," *Current Protocols in Bioinformatics*, vol. 72, no. 1, p. e108, 2020.

[71] H. Ma, F. Jiang, Y. Rong, Y. Guo, and J. Huang, "Robust self-training strategy for various molecular biology prediction tasks," in *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 2022, pp. 1–5.

[72] J. cheol Jeong, X. Lin, and X.-W. Chen, "On position-specific scoring matrix for protein function prediction," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 8, no. 2, pp. 308–315, 2010.

[73] M. S. Klausen, M. C. Jespersen, H. Nielsen, K. K. Jensen, V. I. Jurtz, C. K. Sønderby, M. O. A. Sommer, O. Winther, M. Nielsen, B. Petersen, *et al.*, "Netsurfp-2.0: Improved prediction of protein structural features by integrated deep learning," *Proteins: Structure, Function, and Bioinformatics*, vol. 87, no. 6, pp. 520–527, 2019.

[74] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[75] S. Wang, Y. Guo, Y. Wang, H. Sun, and J. Huang, "Smiles-bert: large scale unsupervised pre-training for molecular property prediction," in *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 2019, pp. 429–436.

[76] J. Yao, X. Zhu, J. Jonnagaddala, N. Hawkins, and J. Huang, "Whole slide images based cancer survival prediction using attention guided deep multiple instance learning networks," *Medical Image Analysis*, p. 101789, 2020.

[77] B. Lee and F. M. Richards, "The interpretation of protein structures: estimation of static accessibility," *Journal of molecular biology*, vol. 55, no. 3, pp. 379–IN4, 1971.

[78] M. Z. Tien, A. G. Meyer, D. K. Sydykova, S. J. Spielman, and C. O. Wilke, "Maximum allowed solvent accessibilites of residues in proteins," *PloS one*, vol. 8, no. 11, p. e80635, 2013.

[79] M.-S. Cheung, M. L. Maguire, T. J. Stevens, and R. W. Broadhurst, "Dangle: A bayesian inferential method for predicting protein backbone dihedral angles

and secondary structure," *Journal of magnetic resonance*, vol. 202, no. 2, pp. 223–233, 2010.

[80] H. Singh, S. Singh, and G. P. Raghava, "Evaluation of protein dihedral angle prediction methods," *PloS one*, vol. 9, no. 8, p. e105667, 2014.

[81] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[82] I. Drori, I. Dwivedi, P. Shrestha, J. Wan, Y. Wang, Y. He, A. Mazza, H. Krogh-Freeman, D. Leggas, K. Sandridge, *et al.*, "High quality prediction of protein q8 secondary structure by diverse neural network architectures," *arXiv preprint arXiv:1811.07143*, 2018.

[83] P. Kountouris and J. D. Hirst, "Prediction of backbone dihedral angles and protein secondary structure using support vector machines," *BMC bioinformatics*, vol. 10, no. 1, p. 437, 2009.

[84] J. M. Berg, J. L. Tymoczko, L. Stryer, *et al.*, *Biochemistry*. New York: WH Freeman, 2002.

[85] P. Śledź and A. Caflisch, "Protein structure-based drug design: from docking to molecular dynamics," *Current Opinion in Structural Biology*, vol. 48, pp. 93–102, 2018.

[86] M. Batool, B. Ahmad, and S. Choi, "A structure-based drug discovery paradigm," *International Journal of Molecular Sciences*, vol. 20, no. 11, 2019.

[87] T. Sun, B. Zhou, L. Lai, and J. Pei, "Sequence-based prediction of protein protein interaction using a deep-learning algorithm," *BMC Bioinformatics*, vol. 18, no. 1, p. 277, May 2017.

[88] M. Zeng, F. Zhang, F.-X. Wu, Y. Li, J. Wang, and M. Li, "Protein–protein interaction site prediction through combining local and global features with deep neural networks," *Bioinformatics*, vol. 36, no. 4, pp. 1114–1120, 2020.

[89] K. Olechnovič and Venclovas, "Voromqa: Assessment of protein structure quality using interatomic contact areas," *Proteins: Structure, Function, and Bioinformatics*, vol. 85, no. 6, pp. 1131–1145, 2017.

[90] F. Baldassarre, D. Menéndez Hurtado, A. Elofsson, and H. Azizpour, "Graphqa: protein model quality assessment using graph convolutional networks," *Bioinformatics*, vol. 37, no. 3, pp. 360–366, 2021.

[91] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv*, vol. 1810.04805, 2019.

[92] R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, X. Chen, J. Canny, P. Abbeel, and Y. S. Song, "Evaluating protein transfer learning with tape," *Advances in Neural Information Processing Systems*, vol. 32, p. 9689, 2019.

[93] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, and R. Fergus, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *Proceedings of the National Academy of Sciences*, vol. 118, no. 15, 2021.

[94] P. Sturmfels, J. Vig, A. Madani, and N. F. Rajani, "Profile prediction: An alignment-based pre-training task for protein sequence models," *arXiv preprint arXiv:2012.00195*, 2020.

[95] R. Rao, J. Liu, R. Verkuil, J. Meier, J. F. Canny, P. Abbeel, T. Sercu, and A. Rives, "Msa transformer," *bioRxiv*, 2021.

[96] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, "Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds," *arXiv Preprint*, 2018.

[97] F. Fuchs, D. Worrall, V. Fischer, and M. Welling, "Se(3)-transformers: 3d roto-translation equivariant attention networks," in *Advances in Neural Information Processing Systems*, 2020.

[98] M. Hutchinson, C. L. Lan, S. Zaidi, E. Dupont, Y. W. Teh, and H. Kim, "Lietransformer: Equivariant self-attention for lie groups," *arXiv Preprint*, vol. 2012.10885, 2020.

[99] C. Shi, S. Luo, M. Xu, and J. Tang, "Learning gradient fields for molecular conformation generation," *arXiv Preprint*, vol. 2105.03902, 2021.

[100] 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: Structure, Function, and Bioinformatics*, vol. 87, no. 12, pp. 1351–1360, 2019.

[101] 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.

[102] K. Olechnovič, E. Kulberkytė, and Č. Venclovas, "Cad-score: a new contact area difference-based function for evaluation of protein structural models," *Proteins: Structure, Function, and Bioinformatics*, vol. 81, no. 1, pp. 149–162, 2013.

[103] A. Zemla, "Lga: a method for finding 3d similarities in protein structures," *Nucleic acids research*, vol. 31, no. 13, pp. 3370–3374, 2003.

[104] Y. Zhang and J. Skolnick, "Scoring function for automated assessment of protein structure template quality," *Proteins: Structure, Function, and Bioinformatics*, vol. 57, no. 4, pp. 702–710, 2004.

[105] J. De Las Rivas and C. Fontanillo, "Protein–protein interactions essentials: key concepts to building and analyzing interactome networks," *PLoS Comput Biol*, vol. 6, no. 6, p. e1000807, 2010.

[106] M. Li, Z. Fei, M. Zeng, F.-X. Wu, Y. Li, Y. Pan, and J. Wang, "Automated icd-9 coding via a deep learning approach," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 16, no. 4, pp. 1193–1202, 2018.

[107] D. Gront, S. Kmiecik, and A. Kolinski, "Backbone building from quadrilaterals: A fast and accurate algorithm for protein backbone reconstruction from alpha carbon coordinates," *Journal of Computational Chemistry*, vol. 28, no. 9, pp. 1593–1597, 2007.

[108] G. G. Krivov, M. V. Shapovalov, and R. L. Dunbrack Jr, "Improved prediction of protein side-chain conformations with scwrl4," *Proteins: Structure, Function, and Bioinformatics*, vol. 77, no. 4, pp. 778–795, 2009.

[109] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *arXiv preprint arXiv:1907.05600*, 2019.

[110] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[111] J. Yang, I. Anishchenko, H. Park, Z. Peng, S. Ovchinnikov, and D. Baker, "Improved protein structure prediction using predicted interresidue orientations," *Proceedings of the National Academy of Sciences*, vol. 117, no. 3, pp. 1496–1503, 2020.

[112] F. Ju, J. Zhu, B. Shao, L. Kong, T.-Y. Liu, W.-M. Zheng, and D. Bu, "Copulanet: Learning residue co-evolution directly from multiple sequence alignment for protein structure prediction," *Nature Communications*, vol. 12, no. 1, p. 2535, May 2021.

[113] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1263–1272.

[114] J. De Las Rivas and C. Fontanillo, "Protein-protein interactions essentials: key concepts to building and analyzing interactome networks," *PLoS computational biology*, vol. 6, no. 6, pp. e1 000 807–e1 000 807, Jun 2010.

[115] X. Li, W. Li, M. Zeng, R. Zheng, and M. Li, "Network-based methods for predicting essential genes or proteins: a survey," *Briefings in Bioinformatics*, vol. 21, no. 2, pp. 566–583, 02 2019.

[116] D. Chen, X. Tian, B. Zhou, and J. Gao, "Profold: Protein fold classification with additional structural features and a novel ensemble classifier," *BioMed Research International*, vol. 2016, p. 6802832, Aug 2016.

[117] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic acids research*, vol. 28, no. 1, pp. 235–242, 2000.

[118] Y. Murakami and K. Mizuguchi, "Applying the naïve bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites," *Bioinformatics*, vol. 26, no. 15, pp. 1841–1848, 2010.

[119] G. Singh, K. Dhole, P. P. Pai, and S. Mondal, "Springs: prediction of protein-protein interaction sites using artificial neural networks," PeerJ PrePrints, Tech. Rep., 2014.

[120] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[121] M. Zeng, B. Zou, F. Wei, X. Liu, and L. Wang, "Effective prediction of three common diseases by combining smote with tomek links technique for imbalanced medical data," in *2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS)*. IEEE, 2016, pp. 225–228.

[122] X. Zhang, S. Wang, F. Zhu, Z. Xu, Y. Wang, and J. Huang, "Seq3seq fingerprint: towards end-to-end semi-supervised deep drug discovery," in *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. ACM, 2018, pp. 404–413.

[123] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[124] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.

BIOGRAPHICAL STATEMENT

Yuzhi Guo received his Ph.D. in Computer Science and Engineering from the University of Texas at Arlington at 2022. Prior to beginning the Ph.D. program, Yuzhi obtained his M.S. degree from Stevens Institute of Technology, USA in 2018 and B.S. degree from Beijing University of Technology, China in 2016. His main research interests are deep learning, machine learning, and bio-informatics. During his Ph.D. program, he has published several papers in the top tier conferences and journal in the literature such as the, International Conference on Research in Computational Molecular Biology (RECOMB), IEEE International Conference on Bioinformatics and Biomedicine (BIBM), AAAI Conference on Artificial Intelligence (AAAI), ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM BCB), International Conference on Computer Vision (ICCV), Journal of Computational Biology (JCB), Biomolecules.