Protein Function Prediction with Primary-Tertiary Hierarchical Learning

Arnold Kazadi[†]
Rice University & Baidu Research
arnoldnyamabo@rice.edu

Ming Zhang
Baidu Research
zhangming20@baidu.com

Jingbo Zhou[*]
Baidu Research
zhoujingbo@baidu.com

***Abstract*—Computational methods can automate the task of protein function prediction, which is expensive and time-consuming if conducted in traditional laboratory settings. However, most of the current methods overlook the inherent primary-tertiary hierarchy that exists between the different representation phases a protein goes through. That is sequence and three-dimensional (3D) structure conformation. They either work with one form of representation or the other. In this work, we propose a deep learning model that successively leverages the 3D representation and sequence representation for protein function prediction. We conduct an extensive experimental evaluation on two public datasets to show that our method outperforms state-of-the-art approaches on protein function prediction. Source code and data are available at https://github.com/PaddlePaddle/PaddleHelix/tree/dev/apps/protein_function_prediction/PTHL. A web server to represent the proposed method for protein function prediction is available at https://paddlehelix.baidu.com/technique#PTHL.**

***Index Terms*—Protein function prediction, Graph neural networks, Protein 3D geometry, Protein sequence**

I. Introduction

Proteins display a wide variety of functions ranging from the growth to the maintenance of the body [1]. Being able to determine these function(s) can be instrumental in solving tasks such as the development of new drug therapies [2–4]. Manually determining these functions through wet-lab experiments can be expensive and time-consuming. Many computational methods are therefore being employed, among which learning-based methods are showing exciting results.

A protein is a sequence of amio acids/residues (at the primary structure level) which determines how the protein will fold into a three-dimensional (3D) structural conformation (in the tertiary structure level) in an Euclidean space. As shown in Fig. 1, this structural conformation determines the function(s) of the protein [5, 6]. There is, therefore, an inherent primary-tertiary hierarchy entailed in a protein representation. Due to the great success of deep learning on sequential data such as texts, natural language processing (NLP) models have been used for the protein function prediction, where the protein sequence is considered as a sentence, that is, a sequence of characters [7–11]. Other methods have only focused on exploiting the structural representation of proteins instead. Here as well, following the success of 3D Convolutional Neural Networks (CNN) in Computer Vision, some proposed methods have considered representing the 3D spatial representations as 3D grids [12]. However, this approach is not memory efficient, as the decomposition of a protein into grids is not trivial and most of the space is unoccupied. An alternate structural representation treats a protein as a 2D graph generated from its contact-map [13, 14]. Yet, representing a 3D object in 2D loses some critical 3D geometry features. In both cases, that is, sequence representation only and structure representation only, important information is ignored.

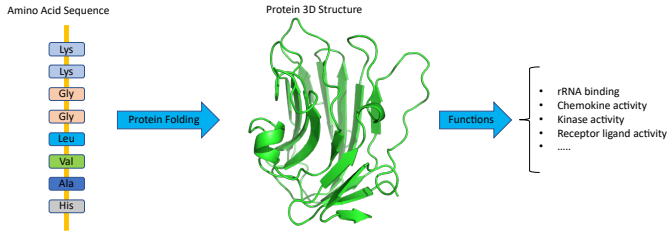

Fig. 1. A protein starts as a sequence of amino acid residues (in the primary level) and later folds into a 3D structure (in the tertiary level) which, in turn, determines functions of the protein.

We propose Primary-Tertiary Hierarchical Learning (PTHL) for protein function prediction that represents proteins by considering both primary (sequential) and tertiary (3D structure) levels. The 3D structure is utilized to learn fine-grained representation with a specially designed Protein Geometry-aware GNN (PG-GNN). This representation is then used in a sequence learning model, Primary-Tertiary encoder (PT-Encoder), to get the final representation of a protein. This learns protein representation hierarchically with information flowing from the fine-grained 3D structural conformation to the sequential representation.

PG-GNN aims to learn the protein geometry information at the tertiary structure level. The novelty of PG-GNN relies on a Residue-wise Reflection Vector Perceptron (RR-VP), as well as an $\alpha \rightarrow \beta$ oriented message passing tailored to learn protein tertiary structure. Taking Geometric Vector Perceptron (GVP) [15] as the basis model to process the residue features, RR-VP is devised to encode the geometry information of each residue associated with its local reference frame. Compared with the common message passing of GNNs,

[†]This work was done when the first author was an intern at Baidu Research under the supervision of Jingbo Zhou. [*]Corresponding author.

the novel part of the $\alpha \rightarrow \beta$ oriented message passing is to take the relative positions/directions between two residues into consideration. The primary-tertiary encoder (PT-Encoder) uses a transformer encoder to process the sequence of residues whose representations are output by PG-GNN.

In summary, the main contributions of this work are:

- We propose a method, PTHL, for protein function prediction that processes a protein in a hierarchical way by first learning the fine-grained 3D structural conformation of residues. These are later leveraged by a sequence model to predict the final protein function.
- We propose a protein geometry-aware graph neural network which uses RR-VP to preserves not only the geometry of the 3D structural conformation of a protein, but also the geometry of each residue. We further improve this method by incorporating the relative position of residues when doing message passing.
- Experimental evaluation performed on several public datasets shows improvement in performance compared to state-of-the-art methods.

II. Related Work

Proteins are mainly represented as either sequence or structure data. Sequence-based methods have borrowed methods developed in NLP by considering the protein sequence as a sentence [16]. The majority of methods for protein/enzyme function prediction in the sequence-based category have employed 1D CNNs [8, 9, 17]. Similar trends have been also observed in other protein-related tasks such as protein fold classification [18], and protein-compound interaction [19]. The increasing availability of public protein databases has prompted an advancement in protein representation learning [20–23], however, these methods usually require a very high computation cost to train protein language models on a large amount of protein database. Since proteins' functions are strongly dependent on their spatial representations, 3D CNNs have been adopted where a protein is treated as a volume that can be divided into a 3D grid [12].

Other approaches consider proteins as graphs, where residues are represented as nodes and the links between them are generated based on their proximity in the Euclidean space. GNNs are the go-to deep learning architectures when taking the graph representation approach [13, 14, 24–26]. Jing et al. [15] proposed geometric vector perceptrons/neurons which extends 1D scalar operations to 3D vectors for protein 3D representation learning. However, these methods overlook the inherent hierarchy between different structure levels of a protein. That is, the sequential and structural representations. In this work, we present a method that combines them by flowing information from the fine-grained 3D representation of a protein to the sequential one, from which the final protein representation is derived.

III. Problem Formulation

Proteins are composed of one or more long chains of residues (amino acids) called protein chains. Each protein chain plays a set of roles that define its functions in the organism. We are, therefore, predicting the function(s) of the individual protein chains (of a protein). This task follows the same setting of previous studies, like [13]. Furthermore, protein functions, depending on biological activities they are involved in, environments where they occur, are divided into three categories: *molecular function* (MF), *cellular component* (CC), and *biological process* (BP). A protein chain can have functions belonging to each one of these categories.

We now formulate the task of protein function as follows. Given a set of protein chains $\mathcal{P} = \{p_i\}$ and the sets of protein functions $\mathcal{F}_o$ (where $o =$\{MF, CC, BP\}) , we are proposing a method that can approximate the underlining (complex) mapping functions $f_o : \mathcal{P} \rightarrow \mathcal{F}_o$ that exists between a protein chain and its function.

IV. Method

The overall architecture of our methodPTHL is in Fig. 2.

A. Protein Primary and Tertiary Structures

Our method works successively with proteins' 3D conformations (tertiary level) and sequences of residues (primary level). A protein is first represented as a graph $\mathcal{G} = (V, E)$, where each node $v_i \in V$ represents a residue R_i, and $E \subset V \times V$ are the edges. Each node v_i is associated with a 3D vector $\mathbf{c}_i (\in \mathbb{R}^3)$ to account for the protein's 3D conformation. This is chosen to be the coordinates of the C-α of the residue R_i. There is an edge between v_i and v_j if their Euclidean distance is less than a given threshold δ, that is, $\|\mathbf{c}_i - \mathbf{c}_j\|_2 \leq \delta$. After learning residues' 3D representations, they are rearranged in a sequence (as they naturally occur in the protein), processed with a sequence-based model to output the protein's final representation.

B. Protein Geometry-aware GNN

The basic component of our Protein Geometry-aware GNN (PG-GNN) is the geometric vector perceptron (GVP)[15]. We improve the GPV as a Residue-wise Reflection Vector Perpection (RR-VP) to consider the geometry information of each residue. Finally, we introduce how to use the proposed $\alpha - \beta$ oriented message passing method to take the relative position between residues into consideration of the PG-GNN.

Using a graph to represent a protein loses important 3D geometry features because graphs only show topological structures of objects,. To remediate to that, some recent works have proposed to redefine the notion of features, by distinguishing between scalar features and vector features [15]. The latter are derived from vectors and should be treated differently. Equipping a graph with vector features can preserve and leverage the information contained in the 3D conformation of a protein, central to protein functions. Each node is associated with scalar features $\mathbf{s} \in \mathbb{R}^n$ and vector features $\mathbf{V} \in \mathbb{R}^{\nu \times 3}$, represented as a tuple $(\mathbf{s}, \mathbf{V}) \in \mathbb{R}^n \times \mathbb{R}^{\nu \times 3}$. These features are processed by a geometric vector perceptron (GVP) instead of the ordinary multi-layer perceptron (MLP) [15] to obtain the updated features $(\mathbf{s}', \mathbf{V}') \in \mathbb{R}^m \times \mathbb{R}^{\mu \times 3}$. That is, $(\mathbf{s}', \mathbf{V}') = \text{GVP}(\mathbf{s}, \mathbf{V})$ [15].

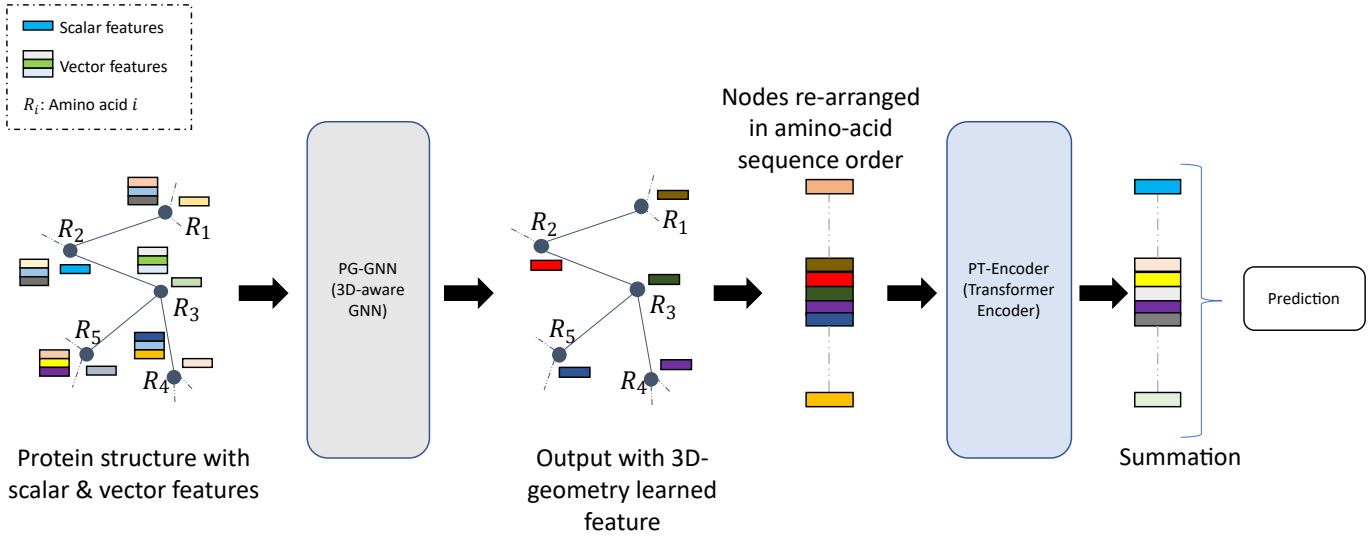

Fig. 2. Overall architecture of PTHL. A protein is first transformed into a graph with scalar & vector features. This is fed into our geometry-aware GNN, PG-GNN, which preserves and learns 3D information from graph. Nodes with ouput features learned from PG-GNN are re-arranged according to the amino acid sequence order, which is fed to PT-Encoder, a transformer encoder, for sequential learning. The output from the encoder is used for the final prediction of the protein functions.

1) Residue-wise Reflection Vector Perceptron (RR-VP): Residues are 3D objects with a conformation of their own. We propose RR-VP to include this additional geometry information into GVP. Each residue R_i is associated with a local reference frame defined by a base $\mathbf{B}_i \in \mathbb{R}^{3\times 3}$, which is used for message passing and node update in the GNN. Given a residue R_i (node v_i) with $\mathbf{B}_i$, $(\mathbf{s}_i, \mathbf{V}_i)$, and $N_{(v_i)} = \{v_j \mid (v_j, v_i) \in E\}$, the message passing operation is :

$$\mathbf{m}_{j\to i} = [\mathbf{s}_j \parallel \mathbf{s}_{j,i}] \quad \forall j \mid v_j \in \mathcal{N}_{(v_i)} \tag{1}$$

$$\mathbf{M}_{j\to i} = [\mathbf{V}_j \parallel \mathbf{V}_{j,i}]\mathbf{B}_i \quad \forall j \mid v_j \in \mathcal{N}_{(v_i)} \tag{2}$$

$$(\mathbf{m}'_{j\to i}, \mathbf{M}'_{j\to i}) = \text{GVP}(\mathbf{m}_{j\to i}, \mathbf{M}_{j\to i}) \tag{3}$$

where $\parallel$ denotes concatenation, $(\mathbf{s}_{j,i}, \mathbf{V}_{j,i})$ are the features of the edge from the node v_j to v_i. Note that in Eq. (2), vector features are multiplied by $\mathbf{B}_i$ to convert them from the global reference to the local reference of node v_i. Updated node features $(\mathbf{s}'_i, \mathbf{V}'_i)$ of v_i are obtained as follows:

$$\mathbf{a}_i = \sum_{j:v_j\in\mathcal{N}_{(v_i)}} \mathbf{m}'_{j\to i} \qquad \mathbf{A}_i = \sum_{j:v_j\in\mathcal{N}_{(v_i)}} \mathbf{M}'_{j\to i}\mathbf{B}_i^T \tag{4}$$

$$(\mathbf{s}_i^{self}, \mathbf{V}_i^{self}) = \text{GVP}(\mathbf{s}_i, \mathbf{V}_i) \tag{5}$$

$$\mathbf{s}'_i = \mathbf{s}_i^{self} + \mathbf{a}_i \qquad \mathbf{V}'_i = \mathbf{V}_i^{self} + \mathbf{A}_i \tag{6}$$

In Eq. (4), the vector features are multiplied by the transpose of $\mathbf{B}_i$ of node v_i to bring them back to global reference system.

2) $\alpha \to \beta$ Oriented Message Passing: Local reference frame and RR-VP improve the awareness of the 3D geometry in a graph representation of a protein. However, the 3D conformations of residues, which are reduced to single points (nodes) in the graph, are not very much exploited. Therefore, we improve on these 3D geometry aware operations by considering the relative positions/directions between 2 residues when doing message passing.

We define the direction of a residue R_i as the one of the unit vector pointing from its C_α to its C_β. C_α is the central point in the backbone of a protein chain where the different substituents attach to an amino acid. C_β is the first carbon atom of an amino acid side chain. Considering the elative position between residues is important because the side chains of residues are instrumental to overall protein 3D conformation and to determine the functions of the protein.

When updating the features of R_i, its neighborhood is divided into 2 regions: *front* (for nodes with direction vectors within 90° from it), and *back* (for nodes beyond 90°) (Fig. 3). These two regions are treated separately during message passing and update. That is, Eq. (1-4) are applied independently to both regions to obtain $\mathbf{a}_i^{(front)}$, $\mathbf{A}_i^{(front)}$; and $\mathbf{a}_i^{(back)}$, $\mathbf{A}_i^{(back)}$.

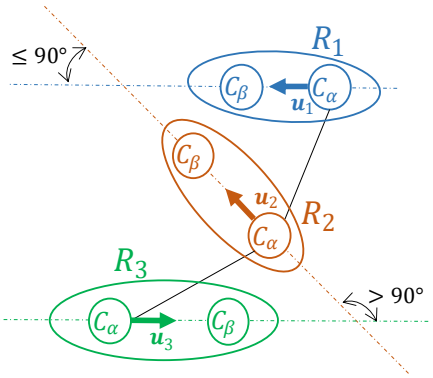

Fig. 3. Defining the front and the back regions of residue R_2 with its adjacent residues R_1 and R_3. Its direction is shown with unit vector $\mathbf{u}_2$ from its C_α and C_β (The same applies for R_1 with $\mathbf{u}_1$ and R_3 with $\mathbf{u}_3$). R_1 is front of R_2 because the angle between $\mathbf{u}_2$ and $\mathbf{u}_1$ is $\leq 90^o$ and R_3 is at the back.

Update for v_i is then reformulated from Eq. (6) to obtain:

$$\mathbf{s}'_i = \mathbf{s}_i^{self} + \mathbf{W}_a[\mathbf{a}_i^{(front)} \parallel \mathbf{a}_i^{(back)}] \tag{7}$$

$$\mathbf{V}'_i = \mathbf{V}_i^{self} + \mathbf{W}_A[\mathbf{A}_i^{(front)} \parallel \mathbf{A}_i^{(back)}] \tag{8}$$

The rationale behind dividing the neighborhood of a residue into two regions is supported by the fact that side-chains of residues (where C_β is found) point into different directions. Thus, adjacent residues would likely face into different directions (which forms the direction $C_\alpha \rightarrow C_\beta$).

C. Primary-Tertiary Encoder

After applying the PG-GNN on the tertiary structure, we propose to use a primary-tertiary Encoder (PT-Encoder) to learn the protein representation upon the primary structure. After L layers of PG-GNN, we get $(\mathbf{s}_i^L, \mathbf{V}_i^L)$ for residue R_i. However, we use only the scalar features $\mathbf{s}_i^L$ for the sequential learning part. $\mathbf{s}_i^L$ contain the fine-grained 3D geometry which will be exploited in the primary structure level sequence representation. To process this sequence we use a Transformer encoder [27]. The final representation of the protein $\mathbf{p} \in \mathbb{R}^F$ is therefore computed from its sequence features $\mathbf{s}_1^L, \mathbf{s}_2^L, \ldots, \mathbf{s}_n^L$.

$$\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_n = \text{Encoder}(\mathbf{s}_1^L, \mathbf{s}_2^L, \ldots, \mathbf{s}_n^L) \tag{9}$$

$$\mathbf{p} = \sum_i^n o_i \tag{10}$$

D. Prediction and Loss Function

Functions of a protein represented by $\mathbf{p}_i$ are predicted as:

$$\hat{\mathbf{y}}_i = \text{sigmoid}(\text{MLP}(\mathbf{p}_i)) \quad \in [0,1]^C \tag{11}$$

where $C = \{|\mathcal{F}_{MF}|, |\mathcal{F}_{CC}|, |\mathcal{F}_{BP}|\}$ (Section III). It is a multi-label prediction problem on which we apply the binary cross-entropy loss function $\mathcal{L}_{BCE}$ as the objective function.

$$\mathcal{L}_{BCE} = \sum_{i=1}^{|\mathcal{P}|} \sum_{j=1}^{C} -\mathbf{y}_{ij} \log(\hat{\mathbf{y}}_{ij}) - (1 - \mathbf{y}_{ij}) \log(1 - \hat{\mathbf{y}}_{ij})$$

where $\hat{\mathbf{y}}_{ij}$ is the j^{th} component of the predicted function vector of $\mathbf{p}_i$ and $\mathbf{y}_{ij}$ is the j^{th} component of its ground truth.

V. Experiments

We conducted experiments on two datasets, one for protein function prediction and the other for enzyme classification.

A. Datasets

We downloaded the PDB dataset [1] and followed the processing and splits proposed by [13] with a threshold $\delta = 10$ Å for graph generation. Only protein chains with sequences less than 1000 were kept because of the right-skewed distribution of protein lengths in the PDB dataset. The training, validation and test sets are split with ratios 8:1:1, that is, 80% of data for training, and 10% for validation and testing, respectively. The test set and the remaining sets (that is, training and validation) are split in a way that there is at most only 40% of sequence identity between them. Thus, generalization is accessed on protein chains (from the test set) with little similarity to those seen during training and validation. We also consider the enzyme prediction dataset also proposed in [13]. The dataset can be downloaded from here[2].

B. Settings

1) Features: For residue R_i, $\mathbf{s}_i \in \mathbb{R}^{22}$ is a one-hot vector representing the amino acid type, $\mathbf{V} \in \mathbb{R}^{2\times 3}$ is composed of the units vectors $\mathbf{v}_p$, $\mathbf{v}_s \in \mathbb{R}^3$ from itself to its predecessor and successor residue R_p, R_s, respectively. That is,

$$\mathbf{v}_{p/s} = \text{Normalize}(\mathbf{c}_{p/s} - \mathbf{c}_i) \tag{12}$$

$\mathbf{B}_i$ of the local reference frame of the residue R_i is defined as in [28] by considering the x-axis as the vector from R_i's N atom to C-α, the y-axis as the cross product between the x-axis and the vector between its C atom and C-α. z-axis is the cross-product between the x-axis and y-axis.

2) Metrics: We use the protein-centric metrics F_{max} and term-centric area under the Precision-Recall curve (AUPRC) for the evaluation of our method same as in [13]

C. Baselines

- **DeepFRI** [13]: represents proteins as graphs and uses a pre-trained language model for node feature initialization.
- **3D-CNN**: we implimented a 3D-CNN where a protein is considered as a box which is voxelized following [12]
- **1D-CNN**: 1D-CNN for protein representation like in[8]
- **CNN_RNN** follows from [29]. Combination of 1D-CNN and GRU.
- **GCN+PT-Encoder**: This method is similar to our method, but GeomGNN is replaced with GCN [30].
- **GAT+PT-Encoder**: This method is similar to our method, but GeomGNN is replaced with GAT [31].
- We also ablate some components of PTHL. **PG-GNN**, no transformer encoder; **PT-Encoder** transformer encoder only; **PTHL(w/o b & f)** ignores the division of a residue's neighborhood into front and back regions; **PTHL(w/o RR-VP)** does not apply the frame reference.

D. Performance Evaluation

TABLE I shows the experiment results. Overall, we can see that our method performs the best for protein function prediction in most of these categories. This proves how important it is to consider both states (i.e., sequential and structural) of proteins for better representation. GAT/GNN+PT-Encoder are similar to our method but perform poorly, this is because they fail to capture the 3D geometry of protein structure which is paramount in determining their functions. However, failing to model this 3D geometry correctly may also produce poor performance. For instance, 3D-CNN models proteins as boxes, which usually results in empty/unoccupied spaces in the model, and loss of rotation-invariant property of proteins. Sequential approaches (such as 1D-CNN, CNN_RNN) perform even better than 3D-CNN due to the lack of proper

[1] https://www.wwpdb.org/ftp/pdb-ftp-sites

[2] https://github.com/flatironinstitute/DeepFRI

TABLE I
COMPARATIVE PERFORMANCE OF DIFFERENT APPROACHES WITH OUR PROPOSED METHOD. BEST RESULTS ARE HIGHLIGHTED IN **BOLDFACE**.

Method	Molecular Function		Biological Process		Cellular Component		Enzyme	
	F_{max}	$AUPRC$	F_{max}	$AUPRC$	F_{max}	$AUPRC$	F_{max}	$AUPRC$
DeepFRI	45.65	37.81	**37.11**	17.52	**45.01**	20.02	67.40	69.32
3D-CNN	29.59	22.60	24.78	10.80	30.20	12.54	45.04	47.80
1D-CNN	28.89	21.23	27.47	12.07	31.82	13.91	47.26	50.61
CNN_RNN	38.44	32.88	26.91	14.85	36.30	18.36	52.77	52.28
GCN+PT-Encoder	38.80	32.08	24.58	14.77	25.60	18.85	57.83	52.61
GAT+PT-Encoder	43.08	36.37	28.25	16.19	28.35	19.67	65.24	60.21
PG-GNN	48.35	39.78	32.91	13.13	40.84	17.58	70.11	68.17
PT-Encoder	26.68	15.03	27.92	08.77	37.60	15.16	37.73	36.95
PTHL (Ours)	**55.30**	**50.07**	34.89	**20.74**	34.35	**23.01**	**75.44**	**75.51**

3D geometry handling. Even though PG-GNN is capable of handling 3D geometry of proteins, it lacks the sequential representation of proteins, and therefore, resulting in poor performance compared to PTHL.

E. Ablation Analysis

We further discuss about PTHL and its variants (lacking one of the components) to showcase its expressiveness (Fig. 4) on both protein function prediction dataset and enzyme classification dataset. We can see that where both the primary and tertiary structures are used; that is, PTHL, PTHL(w/o RR-VP) and PTHL(w/o b&f); we have recorded better results than where this combination is missing (PG-GNN and PT-Encoder). With the tertiary structure or primary structure taken alone, there is a decrease in performance, with the worse performance results registered from PT-Encoder. Even though these two sources of information are used together in PTHL(w/o b &f), we can, however, notice a boost in performance when considering (in PTHL) the *back* and *front* regions of a residue during the message passing, which is also crucial because the side chains of residues are instrumental to overall protein 3D conformation. We can also see the importance of performing the transformation from local system to global system and vice versa during message passing with RR-VP as PTHL performs better than PTHL(w/o RR-VP) in most cases.

VI. CONCLUSION

In this paper, we propose a method that models proteins by leveraging both their primary structure and tertiary structure representations for protein function prediction. A specially designed protein geometry-aware GNN processes the 3D structure of proteins for fine-grained feature learning to be used in a sequential modeling method for function prediction. We show through experiments how this combination (primary + tertiary) is so crucial by quantitatively comparing our proposed method to the state-of-the-art approaches which usually consider only one type of representation (i.e., either sequence or 3D structure). For future work, we want to improve on the protein geometric learning more appropriate for 3D representation. How to utilize the primary structure and tertiary structure learning in other ways also deserves a future investigation.

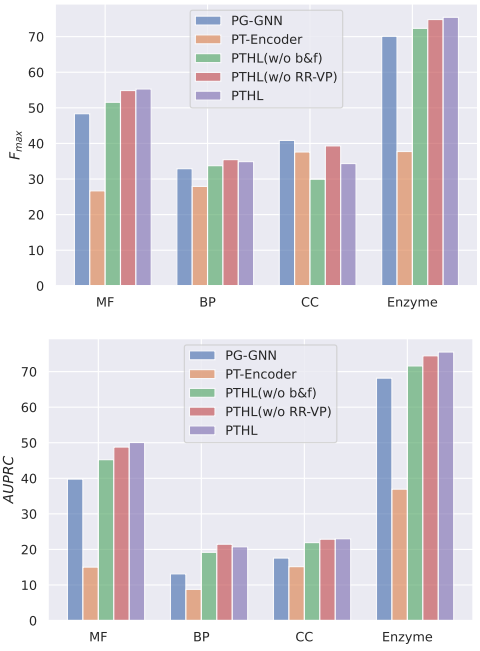

Fig. 4. Comparison of performance between our method PTHL and its variant for ablation evaluation.

REFERENCES

[1] Huan-Xiang Zhou and Sanbo Qin. Interaction-site prediction for protein complexes: a critical assessment. *Bioinformatics*, 23, 2007.

[2] Freyr Sverrisson, Jean Feydy, Bruno E. Correia, and Michael M. Bronstein. Fast end-to-end learning on protein surfaces. In *CVPR*, 2021.

[3] Chengxin Zhang, Peter L Freddolino, and Yang Zhang. Cofactor: improved protein function prediction by combining structure, sequence and protein–protein interaction information. *Nucleic acids research*, 45(W1):W291–W299, 2017.

[4] Petri Törönen, Alan Medlar, and Liisa Holm. Pannzer2: a rapid functional annotation web server. *Nucleic acids research*, 46(W1):W84–W88, 2018.

[5] InterPro in 2019: improving coverage, classification and access to protein sequence annotations. *Nucleic Acids*

Research, 47(D1), 2018.

[6] Natalie L. Dawson, Tony E. Lewis, Sayoni Das, Jonathan G. Lees, David Lee, Paul Ashford, Christine A. Orengo, and Ian Sillitoe. CATH: an expanded resource to predict protein function through structure and sequence. *Nucleic Acids Research*, 45, 2016.

[7] Rui Fa, Domenico Cozzetto, Cen Wan, and David T. Jones. Predicting human protein function with multi-task deep neural networks. *PLOS ONE*, 13(6):1–16, 2018.

[8] Maxat Kulmanov and Robert Hoehndorf. DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics*, 36(2):422–429, 2019.

[9] Maxat Kulmanov, Mohammed Asif Khan, and Robert Hoehndorf. DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics*, 34:660–668, 2017.

[10] Ronghui You, Xiaodi Huang, and Shanfeng Zhu. Deeptext2go: Improving large-scale protein function prediction with deep semantic text representation. *Methods*, 145:82–90, 2018.

[11] Ronghui You, Zihan Zhang, Yi Xiong, Fengzhu Sun, Hiroshi Mamitsuka, and Shanfeng Zhu. GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics*, 34(14): 2465–2473, 2018.

[12] Afshine Amidi, Shervine Amidi, Dimitrios Vlachakis, Vasileios Megalooikonomou, Nikos Paragios, and Evangelia I. Zacharaki. Enzynet: enzyme classification using 3d convolutional neural networks on spatial representation. *PeerJ*, 6:e4750, May 2018.

[13] Vladimir Gligorijević, P. Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C. Taylor, Ian M. Fisk, Hera Vlamakis, Ramnik J. Xavier, Rob Knight, Kyunghyun Cho, and Richard Bonneau. Structure-based protein function prediction using graph convolutional networks. *Nature Communications*, 12, 2021.

[14] Boqiao Lai and Jinbo Xu. Accurate protein function prediction via graph attention networks with predicted structure information. *Briefings in Bioinformatics*, 23(1), 2021.

[15] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *ICLR*, 2021.

[16] Mai Ha Vu, Rahmad Akbar, Philippe A Robert, Bartlomiej Swiatczak, Geir Kjetil Sandve, Victor Greiff, and Dag Trygve Truslew Haug. Linguistically inspired roadmap for building biologically reliable protein language models. *Nature Machine Intelligence*, 5(5):485–496, 2023.

[17] Alperen Dalkıran, Ahmet Süreyya Rifaioğlu, Tunca Dogan, and Mehmet Volkan Atalay. Ecpred: a tool for the prediction of the enzymatic functions of protein sequences based on the ec nomenclature. *BMC BIOINFORMATICS*, 2018.

[18] Jie Hou, Badri Adhikari, and Jianlin Cheng. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 12 2017.

[19] Masashi Tsubaki, Kentaro Tomii, and Jun Sese. Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, 35:309–318, 2018.

[20] Ehsaneddin Asgari and Mohammad R. K. Mofrad. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLOS ONE*, 10:1–15, 2015.

[21] Kevin K Yang, Zachary Wu, Claire N Bedbrook, and Frances H Arnold. Learned protein embeddings for machine learning. *Bioinformatics*, 34(15):2642–2648, 2018.

[22] Zuobai Zhang, Minghao Xu, Arian Rokkum Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. In *ICLR*, 2023.

[23] Can Chen, Jingbo Zhou, Fan Wang, Xue Liu, and Dejing Dou. Structure-aware protein self-supervised learning. *Bioinformatics*, 39(4):btad189, 2023.

[24] Jingbo Zhou, Shuangli Li, Liang Huang, Haoyi Xiong, Fan Wang, Tong Xu, Hui Xiong, and Dejing Dou. Distance-aware molecule graph attention network for drug-target binding affinity prediction. *arXiv preprint arXiv:2012.09624*, 2020.

[25] Shuangli Li, Jingbo Zhou, Tong Xu, Liang Huang, Fan Wang, Haoyi Xiong, Weili Huang, Dejing Dou, and Hui Xiong. Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. In *KDD*, pages 975–985, 2021.

[26] Shuangli Li, Jingbo Zhou, Tong Xu, Liang Huang, Fan Wang, Haoyi Xiong, Weili Huang, Dejing Dou, and Hui Xiong. Giant: Protein-ligand binding affinity prediction via geometry-aware interactive graph neural network. *TKDE*, 2023.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, 2017.

[28] Ilia Igashov, Nikita Pavlichenko, and Sergei Grudinin. Spherical convolutions on molecular graphs for protein model quality assessment. *Machine Learning: Science and Technology*, 2, jul 2021.

[29] Mohamed E.M. Elhaj-Abdou, Hassan El-Dib, Amr El-Helw, and Mohamed El-Habrouk. Deep_cnn_lstm_go: Protein function prediction from amino-acid sequences. *Comput. Biol. Chem.*, 95(C), dec 2021.

[30] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.