GNNGO3D: Protein Function Prediction Based on 3D Structure and Functional Hierarchy Learning

Liyuan Zhang, Yongquan Jiang*, Yan Yang, Member, IEEE

Abstract—Protein sequences accumulate in large quantities, and the traditional method of annotating protein function by experiment has been unable to bridge the gap between annotated proteins and unannotated proteins. Machine learning-based protein function prediction is an effective approach to solve this problem. Most of the existing methods only use the protein sequence but ignore the three-dimensional structure which is closely related to the protein function. And the hierarchy of protein functions is not adequately considered. To solve this problem, we propose a graph neural network (GNNGO3D) that combines the three-dimensional structure and functional hierarchy learning. GNNGO3D simultaneously uses three kinds of information: protein sequence, tertiary structure, and hierarchical relationship of protein function to predict protein function. The novelty of GNNGO3D lies in that it integrates the learning of functional level information into the method of predicting protein function by using tertiary structure information, fully learning the relationship between protein functions, and helping to better predict protein function. Experimental results show that our method is superior to existing methods for predicting protein function based on sequence and structure.

Index Terms—Graph Neural Networks, Gene Ontology, Language Model, Machine Learning, Protein Function Prediction

1 Introduction

PROTEINS are biomacromolecules responsible for a wide range of activities in our cells, tissues, organs, and bodies, playing a central role in the structure and function of cells [1]. However, proteins with well-characterized functions represent only a small fraction of all known proteins and are limited to a few species. Therefore, accurate prediction of protein function is helpful to accelerate research in the fields of animal and plant breeding, biotechnology, and human health [2]. High-throughput and low-cost sequencing techniques have produced an explosive number of sequences, but only a small number of sequences have been experimentally annotated [3]. The Uniprot database currently includes over 100 million sequences, but only 0.5% of them are manually annotated. Several biological and computational challenges make it difficult to predict protein function, which is determined in the context of an organism and rarely by any single experiment or publication [1]. Understanding the function and mechanism of newly discovered proteins is one of the key biological issues in the post-genome era [4], [5].

Protein functions are defined by GeneOntology(GO), which is composed of directed acyclic graphs. GO contains many terms that describe the biological function of genes and their products and is widely used in the field of protein function [6]. GO represents protein function as three functional ontologies with hierarchical structure: molecular function (MF), biological process (BP), and cellular component (CC) to describe different aspects of these functions. Different ontologies respectively describe the function of different levels of proteins. MF describes the activity of gene products at the molecular level. BP describes the biological processes completed through various molecular activities. CC describes the cellular structure position of gene products when performing their functions. Each ontology is a directed acyclic graph, and each node in the graph represents a function called GO term. The edge between nodes indicates that there is a hierarchical relationship between two GO terms. The root node in the graph represents the parent term, while the leaf node is a further refinement of the parent term [6]. If a protein is annotated with a GO term, it means that the protein has the function indicated by the term. Since gene ontology is a directed acyclic graph, there is a hierarchical relationship between GO terms, and when a protein is annotated by a term, it automatically inherits the functionality of all ancestor terms of this term as well. A protein is usually annotated by multiple GO terms, so protein function prediction can be regarded as a multi-label classification task [7], [8].

Protein sequences contain a variety of biological characteristics related to structure and function. Traditional protein biometric features (e.g., motif sequence profile and secondary structure) are calculated by a set of programs and then combined as sequence feature vectors [9]. Although these methods directly exploit the direct relationship between protein sequence features and biological functions, this requires in-depth knowledge of proteomics

- *L. Zhang is with the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China. E-mail: Zly121tay@my.swjtu.edu.cn.*
- *Y. Jiang and Y. Yang are with the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, Sichuan Province 611756, China, and also with Artificial Intelligence Research Institute, Southwest Jiaotong University, Chengdu, Sichuan 611756, China. E-mail: yqjiang@swjtu.edu.cn, yyang@swjtu.edu.cn.*

*Corresponding author: Yongquan Jiang

as well as higher costs [10].

To solve the large gap between the number of sequences and functions, many computational methods have been developed to automatically predict protein functions [4]. A common type of data used for automatic functional prediction is the amino acid sequence because conserved sequence implies conserved function. Traditional protein function prediction methods used BLAST [11] and the hidden Markov model [12] for sequence similarity comparison. The proteins that need to be measured are compared by sequence similarity in a large database, and functional annotations are transferred from the most similar sequences. This method does not have a good prediction effect on those proteins that are relatively isolated and do not have many similarities in the database.

In addition, with the development of machine learning, machine learning has made great progress in protein function prediction, using neural networks to learn features related to protein function. Conventional machine learning methods, such as support vector machine, random forest, logistic regression, and other algorithms are used for classification problems. Experiments have also determined that machine methods are superior to those based on sequence similarity alignment [13].

It has been shown in the literature that deep learning technology is suitable for complex computing problems with high-dimensional features and complex or non-linear relationships [14]. These techniques can effectively learn task-relevant representations from noise and high-dimensional input data. Convolutional neural network (CNN) [14] in the field of computer vision research has achieved success in protein function prediction. It can extract features of specific tasks from protein sequences, search repeated spatial patterns within a given sequence, and use multiple convolutional layers to stratify them into complex features [4]. And CNN is often used in the architecture of sequence encoders to learn sequence patterns or motifs related to function [6], [15].

Due to a large number of unknown protein sequences (UniprotKB>175M) [2]. Without functional annotation, these protein sequences cannot be directly used to train models for protein function prediction. However, these sequences can be used in unsupervised models to learn amino acid and protein features. Recently, combining methods in the field of NLP, pre-trained protein language models have achieved better performance than other methods in protein function prediction, and are more promising in extracting complex sequence-structure-function relationships [16]. The parameters of the pre-trained model are fixed, and the representations learned using the unsupervised language model can be fine-tuned through supervised training and applied to downstream tasks related to proteins. Studies have also demonstrated that the use of pretraining in bioinformatics is beneficial to protein functions [2], [4], [10], [17].

Protein function is encoded as an amino acid sequence, but the sequence can be diversified during evolution while retaining the same function [2]. Methods based on sequence prediction function use sequence similarity to convey functional information and are not suitable for novel sequences that are not similar to annotated sequences. Proteins fold into three-dimensional structures in living organisms to perform their functions [18]. Protein structure determines its function, and structure is in principle more conserved than sequence [19]. Even if the protein sequence is different, two proteins with similar spatial structures may have the same function [20]. In other words, purely sequence-based methods may not be good at transferring functions between structural homologs [17]. Learning the tertiary structure information of a protein can better predict the function of a protein than the sequence information. Critical Evaluation of Functional Annotation (CAFA), a community-driven benchmark for automated protein functional annotation, has shown that integrated approaches combining multiple protein information are generally superior to sequence-based approaches [5], [22]. Experimental and computational advances in structural biology have made the three-dimensional structure of many proteins available. The Protein Data Bank(PDB) [21] is a database for storing proteins and their complexes, with 170,000 entries [4]. On the other hand, unsupervised protein sequence models are used to capture contact relationships between protein residues. It is also widely used in many protein structure prediction methods [17].

Some studies have used 3D CNN to extract function-related features from protein tertiary structure information [22], [23]. Extract 3D structure from protein data base (PDB). The 3D structure is then converted into a 3D voxelized representation and further fed to ResNet-50 (which is a CNN model) to extract relevant features from the protein structure [22]. Since most of the 3D space is not occupied by proteins, storing and processing 3D representations of protein structures at high-resolution results in low storage efficiency.

In contrast, geometric deep learning methods, as well as some specific graph neural networks GCN [24] and GAT [25], overcome these limitations more effectively in graph-like molecular representations. The purpose of the graph neural network (GNN) is to learn the vector representation of entities and relations in the network and the rules that constitute them, to save the topological relationship between structured input data, and to track the graph structure in the node through the node processing of input data. Two recent studies, DeepFRI [4] and GAT-GO [17], explored the use of graph neural networks for functional prediction in combination with protein sequence and structure information.

DeepFRI [4] used a pre-trained LSTM-based protein language model to extract residue level features of protein sequences, and three-layer GCN [24] learned complex structure-function relationships. GAT-GO [17] uses features extracted from the pre-trained protein language model as the contact map between input and predicted residues to learn the structure-function relationship. The above two methods outperform the current leading methods and sequence-based convolutional neural networks in maximum F-score (Fmax) and area under the precision-recall curve (AUPRC). As we mentioned above, protein function prediction is a multi-label classification task, and a protein is usually annotated by multiple GO terms, which

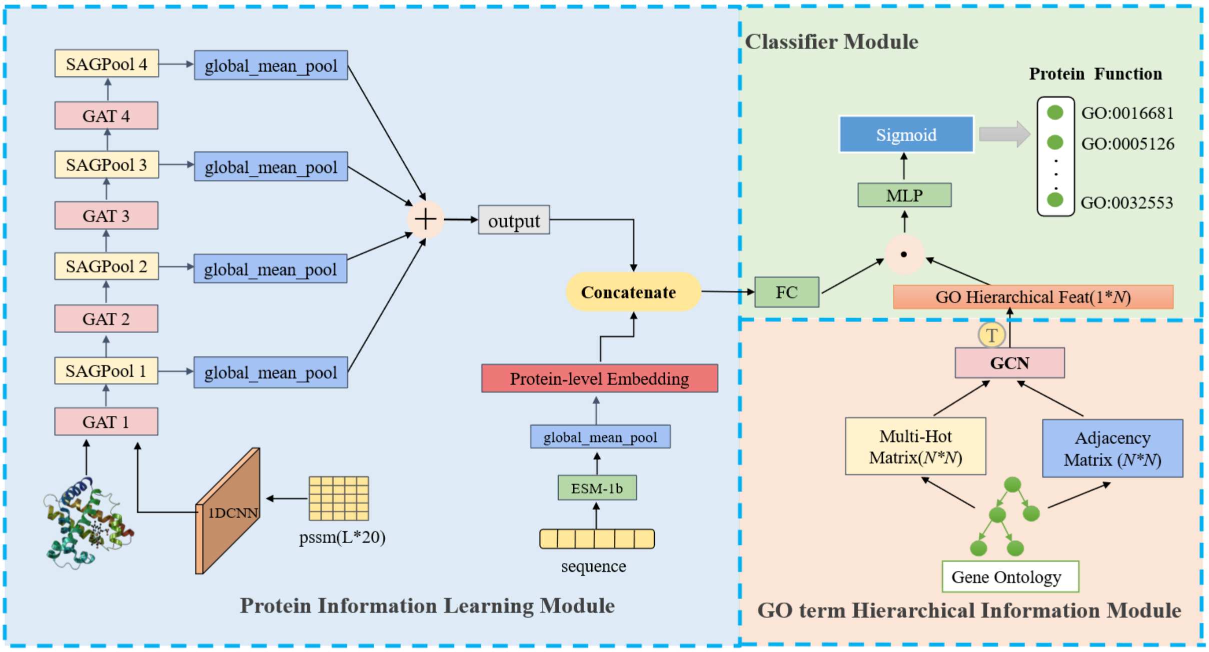

Fig. 1. Schematic method overview. a) Protein Information Learning Module, with four graph convolutional layers for learning complex sequence-structure–function relationships. b) GO term hierarchical information module, used for learning the semantic representation and potential inter-relation-ship of gene ontology. c) MLP classifier to predict GO term probability.

have a hierarchical relationship between parent and son terms.

Existing functional prediction methods based on tertiary structure [2], [4], [17] considered that all protein functional annotations (GO terms) were isolated and regarded protein functional prediction as a planar multi-label classification without considering the hierarchical relationship between GO terms in the directed acyclic graph of gene ontology. It's just that each GO term is uncorrelated. However, DeepFRI[4] and GAT-GO[17] 's function pre-diction tasks have hundreds of functionally related labels, which has certain limitations in the process of function prediction.

In this paper, we propose GNNGO3D, a novel method that utilizes protein sequence and tertiary structure information combined with GO term hierarchy information in gene ontology for function prediction. The overall architecture of the method is shown in Fig. 1. GNNGO3D consists of three modules: a) Protein information learning module. The position-specific score matrix (pssm) generated by Hhblits[26], i.e. protein sequence alignment tool is used as the node features. And the contact of RaptorX[27] predicted protein sequence in space is used as the adjacency matrix of the graph. Both are input to GAT[25] network to learn the relationship between sequence, structure, and function, and learn the protein feature (i.e., *GAT-feat*). Using a pre-trained protein language model to generate sequence-level embeddings, called *sequence-embeddings*. Concatenating the sequence-level embeddings and the output of the graph attention network to obtain the learned protein-level feature is called *protein-feat*. b) GO term hierarchical information module, using GCN[24] to learn the semantic representation and potential interrelationships of gene ontology, and optimize protein representation at the same time, which helps to improve the accuracy of protein function prediction tasks. c) Classifier module, the protein-level feature named protein-feat, and gene ontology semantic representation named *term-vector* are carried out vector dot product to learn the mapping from feature representation to semantic representation in an end-to-end way. The dot product results are fed into the MLP, and the MLP output (the probability of each GO term of the protein sequence) is mapped between 0 and 1 by the Sigmoid function. Meanwhile, protein functional annotation and back-propagation are used to improve the mapping coefficients and obtain consistent representation.

In conclusion, the work contribution of this paper can be summarized in the following ways:

1) The tertiary structure information of protein is combined with the hierarchical information of gene ontology, and the interrelationship of functional annotation is considered comprehensively, which improved the accuracy of functional prediction.
2) Using the multi-stage feature fusion strategy, we read out and add the node features of the graph convolution layer of each layer in the GAT module, which can prevent forgetting some important information and retain the features more relevant to protein function.
3) Our approach uses protein sequence, tertiary structure, homologous evolutionary information, and GO term level information, using more comprehensive features to better characterize proteins.

The rest of this paper is organized as follows. Section 2 describes the data sources, dataset information, and evaluation indicators we used. Section 3 describes the work on our method GNNGO3D. Section 4 concludes our results with a large number of experiments. Finally, Section 5 summarizes this paper and the prospects for future work.

2 Dataset and Evaluation Metrics

2.1 Dataset Formation

The protein sequence database UniprotKB[1] [28] currently contains over 100 million protein sequences available. Protein Data Bank(PDB[2])[21] contains the tertiary structure data of nucleic acid and protein. We downloaded the dataset PDB-cdhit of GAT-GO[17], which was originally constructed by DeepFRI[4]. PDB-cdhit uses CD-HIT [29] to split the dataset into a training set and a test set with 40% homology. The ground truth of PDB-cdhit dataset uses GO term annotation from gene ontology[3], which is organized into three ontologies, MF, BP, and CC, according to different functional categories.

GO term annotations are used for 2752 cross-ontologies from the Gene Ontology database, including 489 MF, 1943 BPO, and 320 CCO. The corresponding GO term of each protein was retrieved from the SIFTS [30] and UniProtKB [28] databases. SIFTS transfers GO annotation to the protein chain of PDB through the protein mapping relationship between Uniprot-ID and PDB-ID [4]. Each protein sample in the PDB-cdhit dataset can be viewed as a separate graph structure, with amino acids as nodes and amino acid contacts in space as edges. The average number of nodes in sample avg_nodes=277, and the average number of edges avg_edges= 2522. The summary of the dataset is shown in Table 1.

TABLE 1 Summary of the PDB-cdhit Data Set

Data set	Protein_nums	Avg_nodes	Avg_edges
Train set	29902	277	2522
Valid set	3323	274	2483
Test set	3416	347	3082

2.2 Evaluation Metrics

We use two main types of assessment in the CAFA challenge: i) the protein-centric indicator Fmax(maximum F-score), which measures the accuracy of assigning GO term to proteins; ii) the area under the precision-recall curve (AUPRC) centered on GO term, which measures the accuracy of protein allocation to different GO terms. In the protein function prediction task, the output of the predictor is the score for each term in the ontology, with the score placed between 0 and 1. A higher score indicates more confidence in the predictor's prediction results.

Fmax is the maximum harmonic average of the accuracy and recall rate of all possible thresholds predicted on the protein term correlation matrix, as shown in Eq. (1).

$$F_{\max} = \max_{\tau}\left\{\frac{2\cdot pr(\tau)\cdot rc(\tau)}{pr(\tau)+rc(\tau)}\right\} \tag{1}$$

First, we use Eq. (2) and Eq. (3) to calculate the average accuracy and recall rates. τ is the threshold range between 0 and 1, with each step of 0.01. v is a single GO term. $P_i(\tau)$ is the binary prediction of GO term at τ . T_i is the truth value GO term set of protein. $I(\cdot)$ is the indicator function.

$$pr_i(\tau) = \frac{\sum_{v\in O} I(v \in P_i(\tau) \wedge v \in T_i)}{\sum_{v\in O} I(v \in P_i(\tau))} \tag{2}$$

$$rc_i(\tau) = \frac{\sum_{v\in O} I(v \in P_i(\tau) \wedge v \in T_i)}{\sum_{v\in O} I(v \in T_i)} \tag{3}$$

$pr(\tau)$ and $rc(\tau)$ are respectively the average accuracy and average recall rates at the threshold τ , calculated as Eq. (4) and Eq. (5). Where $m(\tau)$ is the number of proteins that have at least one GO term in the test set and N is the number of all proteins in the test set.

$$pr(\tau) = \frac{1}{m(\tau)} \cdot \sum_{i=1}^{m(\tau)} pr_i(\tau) \tag{4}$$

$$rc(\tau) = \frac{1}{N} \cdot \sum_{i=1}^{N} rc_i(\tau) \tag{5}$$

We use Eq. (6) to calculate the AUPRC for each term and then take the average AUPRC for all terms. Where Rn and Pn are the accuracy and recall rate when the threshold is n, and N is the total number of thresholds, which in this paper is 100. AUPRC is primarily used to evaluate highly unbalanced label classification problems.

$$AUPRC = \sum_{n=1}^{N}(R_n - R_{n-1})P_n \tag{6}$$

3 Methods

3.1 Sequential Features

a) Hhblits [31] is used to search for similar proteins in the database, and E-value=0.001 is set for generating the position specific scoring matrix (pssm). With a given sequence, proteins with similar sequences in the database can be identified more quickly and accurately, which can pave the way for protein functions to be analyzed.

b) The pre-trained unsupervised protein language model ESM-1b[32] is used as a feature extractor to extract the input protein sequence to the residue level embedding, and then global pooling is carried out to obtain the protein-level feature. ESM-1b is a high-model capacity Transformer trained with hyperparameter optimization, pre-trained with the protein sequence of Uniref 50. Protein sequence language models can learn the physical and chemical properties, secondary and tertiary structures, and internal rules of functions hidden in the input sequences, so as to complete the task of protein function prediction.

3.2 Construction of contact maps

PDB database stores protein structure information in the form of three-dimensional atomic coordinates. The spatial structure information of proteins is usually based on the spatial coordinate information of amino acids to calculate the distance map between amino acids, and the distance matrix of this protein satisfies translation and rotation invariance. t. A certain distance threshold is set to obtain the link relationship between amino acids which is named the contact map. The contact map of a protein can be viewed as a binary adjacency matrix, where each amino acid is represented as a node, and the edge of the adjacency matrix indicates whether two amino acids are in contacThe spatial structure of proteins can be expressed as the topological relationship of amino acids in space.

[1] UniprotKB: https://www.uniprot.org/

[2] Protein Data Bank: https://www.rcsb.org/

[3] Gene Ontology: https://geneontology.org/

When using the PDB database to collect protein structure information, we encountered some problems such as missing amino acids or atoms in protein structure files, and some amino acids in protein sequence did not have 3D atomic coordinate information determined by experiment. This greatly affected the accuracy of predicting protein function.

To solve this problem, we use the protein *Cb-Cb* distance predicted by RaptorX [27] mentioned in GAT-GO [17] and set the distance threshold of 10Å to construct the contact map. The principle of RaptorX [27] is to carry out the convolution transformation of protein sequences through ResNet and also carry out the convolution transformation of interaction between protein residues. Through these two different convolution transformations, the interaction relationship between protein amino acids can be predicted very accurately.

3.3 Graph networks

The traditional deep learning model is designed for a grid or simple sequence, and can not get good performance for graph structure learning.

3.3.1 GCN

Graph Convolutional Networks (GCN) [24] can use the structure of the graph to learn the node representation of the graph. GCN uses the neighborhood information of GO terms for message propagation between GO terms to generate semantic representations and potential interrelationships between GO terms. The multi-hot encoding ($H^0 \in \mathbb{R}^{|N|\times|N|}$) of GO terms and the corresponding adjacency matrix ($A \in \mathbb{R}^{|N|\times|N|}$) between GO terms are taken as input, and the feature representation between GO terms is updated. GCN layer is calculated as follows:

$$H^{l+1} = \text{ReLU}(D^{-\frac{1}{2}}(A+I)D^{\frac{1}{2}}H^l W^l) \tag{7}$$

where *A* is the adjacency matrix, *I* is the identity matrix, *D* is the diagonal matrix of *A*, and *W* is the trainable weight matrix.

3.3.2 GAT

Graph attention network (GAT) [25] uses a mask self-attention layer to solve the shortcomings of graph convolution and other methods, and assigns different weights to different nodes in the neighborhood. GCN has a greater advantage in handling transductive tasks. GAT is better at handling inductive tasks, which means the training set and test set with different types of graphs. The importance of each node of the GAT can be different, with more expressive power.

GAT updates the node representation based on the attention of each node on its neighbors. Its calculation is mainly divided into two steps:

1) Calculate the attention coefficient: for node *i* in the graph, calculate the similarity coefficient between node *i* and its neighbors one by one, as shown in Eq. (8). *hi* and *hj* are the features of nodes *i* and *j*, respectively. *W* is used to increase the dimension of node features. | | concatenates the features of nodes i and j after transformation. *a* means of projecting the concatenated features onto the real numbers.

$$e_{ij} = ([Wh_i] \| Wh_j]), j \in N_i \tag{8}$$

GAT uses mask attention to allocate attention to all neighbor node *j* of node *i*, as shown in Eq. (9).

$$\alpha_{ij} = \text{softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k\in N_i}\exp(e_{ik})} \tag{9}$$

2) Update node features: According to the attention coefficient calculated above, a weighted summation of node features is performed, as shown in Eq. (10), where *hi* is the output of the new node feature.

$$h_i^{'} = \text{LeakyReLU}(\sum_{j\in N_i} a_{ij}Wh_j) \tag{10}$$

3.3.3 SAGPool

SAGPool[33] is a self-attentional diagram pooling method that uses an end-to-end approach to learn structural hierarchy information by calculating self-attentional differentiation between retained and discarded nodes to generate a new subgraph. The calculation process is mainly self-attentional mask and graph pooling:

1) Compute graph node score and node selection

By computing the self-attention score using graph convolution, the result of pooling is based on the features and topology of the graph. As shown in Eq. (11), *A* is the adjacency matrix of the graph, *D* is the diagonal matrix of *A*, *H* is the input node features of the graph, and θ_{att} is the only parameter of the SAGPool layer.

$$Z = \tanh(D^{-\frac{1}{2}}(A+I)D^{\frac{1}{2}}H\theta_{att}) \tag{11}$$

SAGPool's method of node selection preserves only some nodes of the input graph, as shown in Eq. (12). The pooling ratio *k∈(0, 1)* determines the proportion of all nodes occupied by the number of nodes to be retained, *top-rank* is the index of top *KN* before return, and Z_{mask} is the feature attention mask.

$$idx = \text{top-rank}(Z, \lceil kN \rceil, Z_{mask} = Z_{idx} \tag{12}$$

2) Generate a new subgraph

The subgraph generated by SAGPool is processed by operations labeled as masks in the graph. Finally, the new subgraph structure and node features after node deletion is obtained. As in Eq. (13), *X* and *A* are the new feature matrix and adjacency matrix calculated according to the index idx in Eq. (12).

$$X_{out} = X_{idx} \odot Z_{mask}, A_{out} = A_{idx,idx} \tag{13}$$

3.4 GNNGO3D

a) Protein information learning module

1DCNN is used to extract features from the pssm generated by protein sequences, and the deep information is mined. The output of 1DCNN is taken as the node features of the graph, and the protein contact map predicted by RaptorX is used as the adjacency matrix of the graph. Input into a network of four layers of GAT and SAGPool to learn the relationship between protein and function.

Each GAT layer is followed by a SAGPool layer, which learns the structural hierarchy information in an end-to-end manner, and computes self-attention to distinguish between retained and discarded nodes to generate a new subgraph. The global average pooling operation is used to get the protein-level feature of each layer. Finally, the protein-level features of each layer are added to get the final GAT-feat, as shown in Eq. (14), where x_i represents the protein-level feature computed by global pooling of the subgraph node features of SAPool at each layer.

$$GAT\text{-}feat = x1 + x2 + x3 + x4 \tag{14}$$

The output of the four-layer GAT and SAGPool networks, named *GAT-feat*, is combined with the sequence-level embeddings, sequence-embedding, generated using the pre-trained protein language model ESM-1b. The two are concatenated to obtain the final protein-level feature called *protein-feat.*

b) GO term Hierarchical information module

We download the *go-basic.obo* file from the Gene Ontology database. This is the basic version of GO and contains a hierarchy of all terms of the three ontologies. The hierarchical relationships among GO terms are *is_a*, *part of, has part*, and *regulates*. Here we consider the *is_a* relationship. If the term A is *is_a* B, it means that the term A is the subterm of B, and proteins labeled with the term A also have the function of the term B. For MF, BP, and CC ontologies, three graphs are constructed respectively, and terms in the ontologies serve as nodes of the graphs.

The adjacency matrix *Adj* contains hierarchical relationships between terms. We set the child node *c* and parent node *p* as adjacencies between all terms. Considering the number of proteins labeled by different terms in the dataset, that is, the importance of different terms, we calculated the prior probability as the weight of the adjacency matrix. The prior probability is calculated as shown in Eq. (15). *N* is the number of labels, N_c means the number of proteins in the child node term comment, and N_p represents the number of proteins in the parent node term comment.

$$P(U_c \mid U_p) = \frac{P(U_c \cap U_p)}{P(U_p)} = \frac{N_c}{N_p} \tag{15}$$

The node features $H^0 \in \mathbb{R}^{|N| \times |N|}$ are represented as a multi-hot coding matrix, and each row represents a term. The GO term and its ancestor term are coded 1, and *N* represents the number of GO terms.

The adjacency matrix and node features of the gene ontology are input into GCN to learn the semantic representation and potential interrelationships of the gene ontology while optimizing the representation of proteins. Finally, the relationship between GO terms is *term-vector*.

c) Classifier module

The final protein feature learned by the protein information learning module, *protein-feat*, and the output of the GO term hierarchical information module, *term-vector*, are fed into the classifier for multi-label classification. The output of the graph attention module, protein-feat, is firstly reduced from *N*N* to *N*d* by a linear layer, then combine with *term vector* for feature fusion, and finally fed into the classifier for classification.

Protein function prediction is a multi-label classification task, we used the sigmoid function to map the output to [0,1]. The multi-label classification task is a binary classification task for each GO term, so the loss function uses a binary cross-entropy function.

4EXPERIMENTALS AND DISCUSSION

4.1 Training Details

We employed the P100-PCIE for training GNNGO3D on a Linux system. The deep learning framework used was PyTorch [34] and Torch-Geometric [35]. To optimize the model, we utilized the binary cross-entropy loss function and AdamW optimizer with a learning rate of 1e-4. The batch size was set to 8. The hidden channels of GAT layers are set to 1024. We obtained the best results by training MF, BP, and CC for 120, 60, and 60 epochs, respectively. Additionally, we incorporated the CosineAnnealingLR learning rate adjustment strategy.

4.2 Compared Methods

BLAST [11] is a commonly used sequence alignment tool that infers functional information of proteins by comparing the predicted protein sequence with sequences in a known protein database. For each alignment hit, the corresponding GO terms are assigned to the target protein based on the similarity score, which is treated as a prediction probability. By extracting the maximum similarity score for each GO term, the predicted probability can be determined and assigned to the target protein's potential GO annotations.

DeepGO utilizes CNN [14] to automatically learn relevant features in protein sequences. DeepGO employs a 1D CNN model architecture consisting of 16 parallel single-layer convolutional operations, with each convolutional layer having different kernel sizes ([8, 16, ..., 128]) and 512 filters. This structure enables the model to capture local and global features of protein sequences at different scales.

The DeepFRI [4] is a two-stage architecture. The first part of the model is LSTM-LM, which is used to extract amino acid features from the PDB sequence. The second part consists of three layers of GCN.

The GAT-GO [17] utilizes a CNN to encode residue-level sequence features extracted from one-hot encoding, PSSM, and language models. It uses four layers of GAT to take the protein contact map and the representation vectors generated by the CNN as inputs. Through a classifier, the GAT-GO model predicts protein functions.

4.3 Comparison of GNNGO3D with other protein function prediction methods

Our method is compared with other methods based on protein sequence information and protein tertiary structure information on the PDB-cdhit test set with experimentally determined functional annotations. Sequence-based approaches include the standard Naive baseline used in

TABLE 2
PREDICTION PERFORMANCE IN TERMS OF FMAX AND AUPRC ON PDB-CDHIT DATASET

	Fmax			AUPRC		
Model	MF	BP	CC	MF	BP	CC
Naive	0.156	0.244	0.318	0.075	0.131	0.158
BLAST	0.498	0.400	0.398	0.120	0.120	0.163
DeepGO	0.359	0.295	0.420	0.368	0.210	0.302
DeepFRI	0.542	0.425	0.424	0.313	0.159	0.193
GAT-GO	0.632	0.479	0.544	0.659	0.378	**0.474**
Ours	**0.662**	**0.499**	**0.552**	**0.675**	**0.390**	0.472

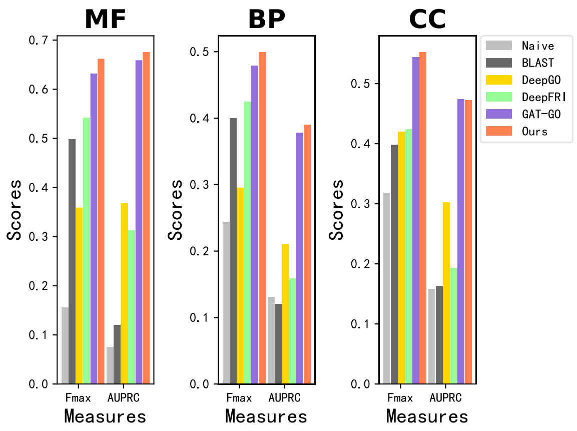

Fig. 2. Fmax and AUPRC performance over GO terms in different ontologies

the CAFA benchmark, BLAST sequence homology alignment methods, and the state-of-the-art sequence-only deep learning method DeepGO[15]. Structure-based approaches include DeepFRI [4] and GAT-GO [17].

All gene ontology are separately trained and evaluated on the PDB-cdhit dataset, and the predictive performance of these methods is evaluated using the protein-centric metric Fmax and the GO-term-centric metric AUPRC. Since we used the same dataset as GAT-GO, we implemented the method of GAT-GO on our device, and the comparison metrics in the PDB-cdhit test set in GAT-GO [17] are used for NAIVE, BLAST, DeepGO, and DeepFRI. The experimental results are shown in Fig.2 and Table 2.

Based on the experimental results, the following conclusions can be drawn:

- In MF and BP, the GNNGO3D outperforms other methods, including Naive, BLAST, DeepGO, DeepFRI, and GAT-GO, in terms of both Fmax and AUPRC metrics.
- In CC, the GNNGO3D surpasses other protein function prediction methods in terms of the Fmax metric and performs comparably to the GAT-GO method in terms of the AUPRC metric.
- Compared to the state-of-the-art model GAT-GO, GNNGO3D improves the Fmax from 0.632 to 0.662 and the AUPRC from 0.659 to 0.675 in MF.

Overall, GNNGO3D demonstrates excellent performance in protein function prediction tasks across multiple gene ontologies. Particularly in MF and BP, GNNGO3D outperforms other methods in terms of Fmax and AUPRC metrics. This suggests that the GNNGO3D method has significant potential in utilizing protein sequence and structural information for function prediction, surpassing the current state-of-the-art methods.

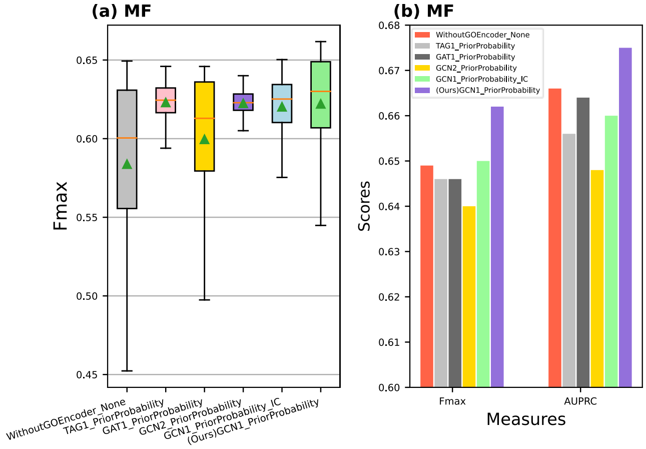

Fig. 3 Performance of different methods on GO term hierarchical information Module. (a) Distribution of the Fmax score under 100 bootstrap iterations for the different methods on GO term Module. (b) Fmax and AUPRC of different methods on GO term Module.

4.4 Comparative experiment

4.4.1 GO term hierarchy information module is helpful for protein function prediction

The GO hierarchical information module is introduced in the prediction task of protein function based on the tertiary structure to learn hierarchical information between protein functions. The hierarchical structure information between GO terms is represented as a graph. The adjacency matrix is constructed with the link relationship between all child and parent terms, and the prior probability calculated by the number of proteins annotated by GO terms is used as the edge weight. The node features are constructed as a multi-hot matrix, and each row of the matrix is a GO term, setting all ancestor nodes of the node as 1.

We first explored whether the GO hierarchy information module is helpful for the function prediction task of the protein gene ontology domain.

Experimental results are shown in Fig. 3. It can be seen that Fmax and AUPRC without GO hierarchical information encoders have only 0.649 and 0.666 respectively. When we use the hierarchical module, Fmax, and AUPRC increase by 1.3% and 0.9%, respectively. We found that the gene ontology hierarchy information can be combined with the features learned from the protein sequence and structure information by the graph neural network, which significantly improves the model prediction performance.

We represent the hierarchical structure between gene ontologies as a graph structure. The semantic representation and potential interrelationships of gene ontology are learned by using graph neural networks to optimize the representation of proteins. We also explore not using *GO term Hierarchical Information Module*, i.e. *Without GO Encoder*. And TAGConv[36], GATConv, GCNConv operators of one layer, and GCNConv operators of two layers are used to learn the strength of the GeneOntology hierarchy information. The results are shown in Fig. 3. Compared with using one-layer GATConv, using one-layer GCNConv has higher prediction performance on Fmax and AUPRC by 1.6% and 1.1%, respectively. This shows that GCN has a great advantage in handling transductive tasks, while GAT is more suitable for handling situations where the graphs processed by the training set and the test set are different.

Finally, Choi and Lee [10] hold the view that although the prior probability can include the hierarchical relationship learning between GO terms, it is highly dependent on the dataset, and the label imbalance problem in the dataset will be applied to the adjacency matrix, which can be solved by adding Information Content (IC). The specific formula is shown in Eq. (16) and Eq. (17). Where *root* stands for the root term, N_k denotes the number of proteins annotated by this term k in the training set and child(k) represents all the children of term k.

$$IC(k) = -\log \frac{freq(k)}{freq(root)} \tag{16}$$

$$freq(k) = N_k + \sum\nolimits_{i \in child(k)} freq(i) \tag{17}$$

We found that the combination of prior probability and IC as features of GO term Fmax and AUPRC have 0.650 and 0.660 respectively, which did not achieve a better performance compared with the prior probability alone. We analyzed the reason for potentially adding prior probability and IC as new GO term features, which not only failed to adequately represent the Gene Ontology information fusion but also resulted in information loss.

In summary, incorporating the GO hierarchy information module is an effective approach in protein function prediction tasks. This module utilizes the hierarchical relationships of GO terms to construct a graph structure and combines protein sequence and structural information to learn the hierarchical relationships among protein functions. Experimental results have also demonstrated that the use of the GO hierarchy information module significantly improves the accuracy of protein function prediction. However, further exploration is still needed on how to better integrate gene ontology information to enhance the predictive ability of protein functions.

4.4.2 Impact of graph convolution operator and graph pooling operator on protein function prediction task

The *protein information learning* module uses four layers of GATConv and SAGPool to learn protein structural features hierarchically.

The goal of graph classification is to use its node features and the structural information of the graph to predict the labels related to the whole graph, which requires graph-level feature representation. GNN is originally designed to learn meaningful node-level features, so a common way to generate graph-level representations from node-level representations is to globally summarize all node representations in the graph. Although this is feasible, it ignores the structural information of the whole graph. Graph pooling models leverage hierarchical learning of structural information to generate better graph-level representations.

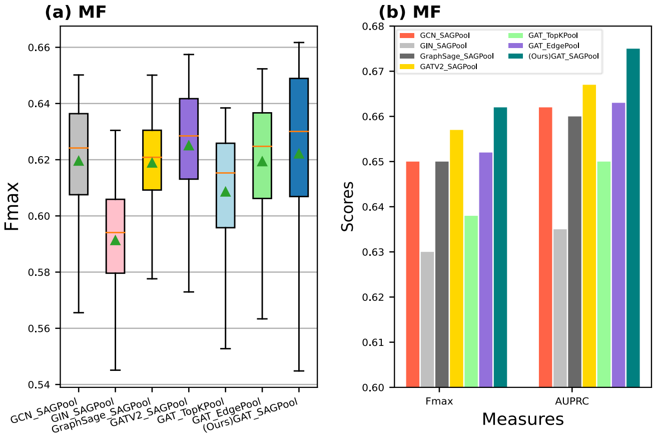

Fig. 4 Performance of different methods on the Protein Information Learning Module. (a) Distribution of the Fmax score under 100 bootstrap iterations for the different methods on the Protein Information Learning Module. (b) Fmax and AUPRC of Different Methods on Protein Information Learning Module.

We explore the influence of different graph convolution operators such as GCN, GIN [37], GraphSage [38], GAT, GATV2 [39], and different graph pooling operators of TopKpool [40], EdgePool [41] , SAGPool on the performance of protein function prediction task in MF ontology. The results are shown in Fig. 4. It can be seen that GAT has better performance than other convolution operators in learning features from graph structure information to predict protein function. This shows that GCN has a great advantage in transductive tasks. However, GCN cannot complete inductive tasks properly. That is, it cannot handle Transductive tasks differently between training set and test set. The importance of each node of GAT can be different than that of GCN, and as a result, GAT has greater presentation power.

TopKPooling scores nodes based on learnable projection vectors and sample nodes with high scores. It avoids node aggregation and calculation of soft distribution matrix and maintains sparsity in graph operation. SAGPooling improves TopKPooling by using GNNS to consider the graph structure when scoring nodes. EdgePooling designs the pooling operation by contracting edges in the graph, but it has poor flexibility because it will always pool about half of the total nodes. Compared with TopKPooling at Fmax and AUPRC of 0.638 and 0.650, and EdgePooling at Fmax and AUPRC of 0.652 and 0.663, SAGPooling has significantly higher performance in function prediction.

Additionally, we conducted a study on the impact of the number of layers in GAT and SAGPool models on protein function prediction tasks. Under the same experimental conditions, we trained GAT and SAGPool models with one to five layers and recorded the Fmax and AUPRC metrics for each group of experiments. The experimental results are shown in Fig. 5.

Between one layer of GATConv and two layers of GATConv, there is little difference in Fmax and AUPRC metrics. However, there is a slight improvement in the metrics when using three layers of GATConv, suggesting that increasing the number of GATConv layers can slightly enhance model performance. Furthermore, when the number of GATConv layers is increased to four, both Fmax and

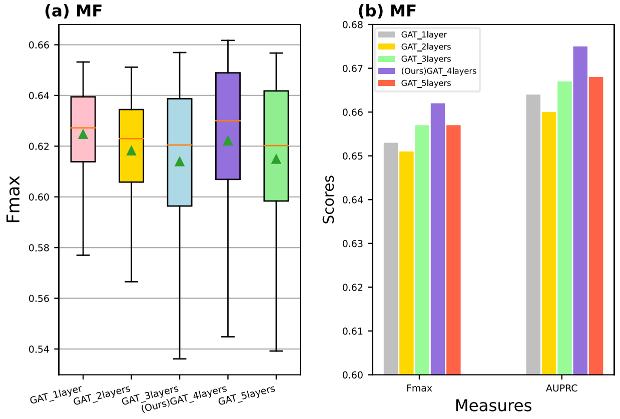

Fig. 5 Performance with Different GATConv layers on GNNGO3D. (a) Distribution of the Fmax score under 100 bootstrap iterations with Different GAT layers. (b) Fmax and AUPRC with Different GAT layers on GNNGO3D.

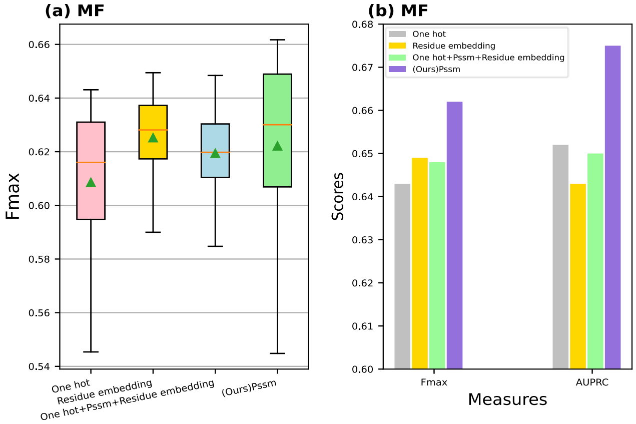

Fig. 6 Performance with Different Feature Combinations on GNNGO3D. (a) Distribution of the Fmax score under 100 bootstrap iterations with Different Feature Combinations. (b) Fmax and AUPRC with Different Feature Combinations on GNNGO3D.

AUPRC metrics reach their highest values, indicating a significant improvement in model performance. However, when the number of GATConv layers increases to five, the model performance decreases, possibly due to overfitting caused by too many layers.

In conclusion, our experimental results suggest that using four layers of GATConv is a better choice in terms of improving model performance while avoiding the overcomplexity that may lead to overfitting issues.

4.4.3 The influence of different protein information on protein function prediction task

It has been shown that sequence similarity is highly correlated with the biochemical properties of proteins, and complex sequence modeling methods can be performed using simple vector representations of sequence similar features. Experiments show that HMMER, a hidden Markov model-based biomolecular similarity detection method, can compete with deep learning-based protein representation methods [16].

Given these results, we add homology information to the training of the representation learning model and explore the impact of using protein information such as pssm, one-hot representation of protein sequences, and residue-level embeddings extracted from the pre-trained protein language model ESM-1b [32] as input to the protein information learning module on function prediction in MF ontology. Experimental results are shown in Fig. 6.

Compared to using one-hot encoding and residue-level embeddings extracted by esm-1b, adding homologous information (pssm) to the training of the representation learning model does improve the predictive performance. We conducted a control experiment based on GAT-GO, which uses a combination of one-hot encoding, pssm, and residue-level embeddings extracted by a protein language model. The results showed that when multiple features were used, the performance metrics Fmax and AUPRC decreased by 1.4% and 2.5%, respectively, and the prediction confidence was lower.

Based on these analyses, we can conclude that incorporating homologous information (pssm) in the training of the representation learning model has a positive impact on protein function prediction. However, when using multiple features, there is a possibility of introducing noise or redundant information, which can lead to decreased performance and lower prediction confidence. Therefore, it is important to carefully consider and evaluate different factors when selecting and integrating features in order to find the optimal model configuration for this task.

4.4.4 Comparison of different protein pre-training models as feature extractors for functional prediction tasks

We used the following protein pre-training model as a feature extractor to generate protein embeddings.

Bepler [42]: A protein sequence encoding model trained with a two-step feedback mechanism (global structural similarity information between proteins and residue contact maps of individual proteins) using bidirectional LSTM. It is trained on a complete protein domain sequence set (a total of 21,827,419 sequences) from the Pfam [43] database.

Cpcprot [44]: An unsupervised contrastive learning framework for protein representation based on maximizing mutual information. It is pre-trained on 32,207,059 proteins from the Pfam database.

PlusRnn [45]: A protein sequence representation model using structural informatics learning for pretraining a Bidirectional Recurrent Neural Network, resulting in the PLUS-RNN model. The Pfam dataset is used as the pretraining dataset, which includes 14,670,860 sequences from 3,150 families.

Seqvec [46] is a deep unsupervised protein sequence model called Seqvec, which is based on the natural language processing model ELMO. The model consists of a character-level CNN and two layers of bidirectional LSTM. It is pre-trained on a large-scale unlabeled dataset called UniRef50.

The experimental results of using different pre-trained models as feature extractors are shown in Fig. 7. It can be observed that compared to other pre-trained models as feature processors, the features extracted using Esm-1b exhibit significant superiority in protein function prediction tasks. This is also the reason why GNNGO3D chose ESM-1b as the sequence feature extractor.

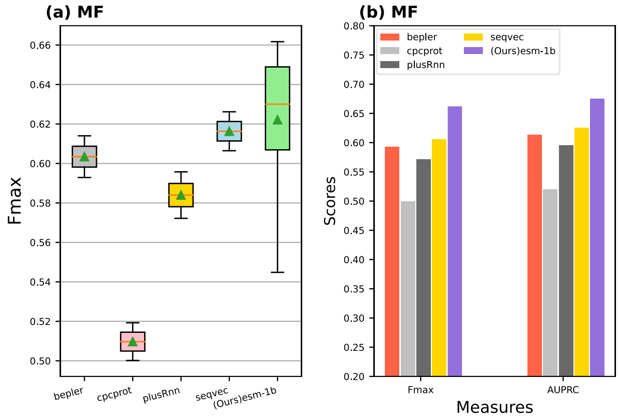

Fig. 7 Performance with Different feature extractors on GNNGO3D. (a) Distribution of the Fmax score under 100 bootstrap iterations with Different feature extractors. (b) Fmax and AUPRC with Different feature extractors on GNNGO3D.

4.5 Ablation experiment

To verify the effectiveness of the modules and innovations in our model, ablation experiments are conducted. The use of the GO hierarchical information module, GAT-SAGPool, and multi-stage feature fusion strategies are evaluated on MF ontology. The experimental results are shown in Table 3, where *Without Add Different Level* indicates that the multi-stage feature fusion strategy is not used in GNNGO3D. *WithotGATModule* indicates that the output of 1DCNN is directly concatenated with *protein-level embedding* in Fig. 1(a) without using GAT and SAGPool with four layers. *Without GO Encoder* means that the *GO term Hierarchical Information Module* is not used in Fig. 1(b).

When only the pssm is fed into CNN, without using GAT-SAGPooling to learn protein structure information, Fmax and AUPRC have 0.644 and 0.654, respectively. By using GAT-SAGPooling, the predictive performance of Fmax and AUPRC of MF ontology is improved by 1.8% and 2.1% respectively.

If we only read out the residue-level features entered by SAGPooling in the last layer into protein-level feature, Fmax and AUPRC are 0.645 and 0.655. The multi-stage feature fusion strategies Fmax and AUPRC increased by 1.7% and 2% respectively. This indicates that the multi-stage feature fusion strategy can avoid forgetting important information during training.

Regarding why GNNGO3D uses feature addition instead of feature concatenation for multi-level feature fusion, it is because when using feature concatenation at each layer, it achieved an Fmax of 0.658 and an AUPRC of 0.669, whereas when using feature addition at each layer, it obtained an Fmax of 0.662 and an AUPRC of 0.675. Compared to the feature concatenation method, the feature addition method allows for better integration of information, elimination of interference and noise, and improvement of the gradient vanishing problem. Therefore, it can yield better results in this task.

4.6 Significance test

To compare the differences in prediction performance of GNNGO3D and GAT-GO in MF, BP, and CC, we conducted significance tests using p-values and t-values.

TABLE 3
GNNGO3D PERFORMANCE WITH DIFFERENT ARCHITECTURES

Ablation experiment	Fmax	AUPRC
Without Add Different Level	0.645	0.655
Without GAT Module	0.644	0.654
Without GO Encoder	0.649	0.666
Ours	0.662	0.675

The p-value is used to evaluate the probability of observing the data under the null hypothesis. When the p-value is small (usually less than 0.05), we consider that the observed data has a low probability of occurring under the null hypothesis. The t-value is a statistic in t-tests used to assess the significance of differences between the means of two groups of samples. Specifically, the larger the absolute value of the t-value, the more significant the difference between the sample means.

TABLE 4
SIGNIFICANCE TEST FOR GNNGO3D AND GAT-GO

Test of Significance	MF	BP	CC
p-value	1.00 e-17	1.35 e-3	0.66
t-value	8.60	3.21	-0.43

The results are shown in Table 4. For the MF ontology, the p-value is 1.00e-17 (much smaller than 0.05), meaning that there is a significant difference in performance between GNNGO3D and GAT-GO in MF. The t-value is 8.60, indicating a significant difference in the sample predictions of the two methods. For the BP ontology, the p-value is still smaller than 0.05, indicating a significant difference in performance between GNNGO3D and GAT-GO in BP.

For the CC ontology, the p-value is larger than 0.05, indicating that there is no significant difference in performance between GNNGO3D and GAT-GO in CC. This could be due to the fact that both our method and GAT-GO utilize GNN networks. Additionally, in the CD-HIT dataset, the CC ontology has only 320 functional annotations, which limits the hierarchy features learned by the GO hierarchical information module and provides limited assistance for function prediction tasks, resulting in less noticeable differences.

4.7 Performance in the LIGASE protein family

To validate the GNNGO3D's broad applicability and effectiveness, experiments are planned to be conducted on a new protein family. The LIGASE [21] protein family plays a crucial role in DNA repair, replication, and recombination, as well as RNA repair and splicing processes, ensuring the accurate transmission and stability of genetic information in cells.

For this purpose, we selected 107 proteins from the LIGASE family and used our GNNGO3D to predict their MF, BP, and CC functions. The results are shown in Table 4.

The GNNGO3D achieved significant performance in predicting the MF ontology of LIGASE family proteins, with Fmax and AUPRC values of 0.764 and 0.796, respec-

TABLE 5
GNNGO3D PERFORMANCE IN THE LIGASE FAMILY

	Fmax			AUPRC		
DataSet	MF	BP	CC	MF	BP	CC
PDB-CDHIT	0.662	0.499	0.552	0.675	0.390	0.472
LIGASE	0.764	0.581	0.587	0.796	0.398	0.399

tively. This indicates that the GNNGO3D can more accurately predict the functional characteristics of LIGASE family proteins, with higher recall and precision rates. This is of great significance for a deeper understanding of the functional mechanisms of LIGASE family proteins in MF Ontology.

In terms of the BP ontology and CC ontology, the GNNGO3D's performance in predicting the biological process functions of the LIGASE family has improved compared to the PDB-CDHIT dataset, with Fmax and AUPRC values of 0.581 and 0.398, respectively. With 1943 GO terms in the BP ontology, this suggests that the biological process functions of LIGASE family proteins are more complex compared to other functional ontologies and may exhibit greater diversity. As for the CC ontology, the poor performance of AUPRC for the LIGASE family may be due to decreased accuracy in judging positive and negative instances or potential influences from class imbalance and the complexity of data features.

5 Conclusion

In this paper, we proposed GNNGO3D, a method based on protein structural information combined with functional hierarchical relationships to effectively improve protein function prediction. We used the encoder that learns the hierarchical relationship of GeneOntology in the tertiary structure-based protein function prediction task to construct the node feature matrix of GO terms and the adjacency matrix that represents the hierarchical relationship of GO terms. The experimental results shows that our method is superior to the current protein function prediction methods in MF, BP. The GO term hierarchical information module used in our approach does improve the performance of protein function prediction, and considering the output of the graph convolution combined with each layer during message passing prevents some important information from being forgotten.

Most of the existing methods of protein structure representation are only the coordinates of a certain amino acid atom or the use of pre-trained models to build contact maps, which cannot completely represent the characteristics of proteins. We will then consider other ways to better represent the spatial structure of the protein, as well as consider the addition of protein helices and amino acid angle information to design more suitable models.

Protein function prediction is a multi-label classification task, and most proteins are annotated by multiple labels, and labels are not isolated. How to better learn the relationship between labels and consider the influence of other aspects of prior knowledge on the prediction task still needs to be explored.

Acknowledgment

The research work was supported by the National Natural Science Foundation of China (No.61976247) and the Fundamental Research Funds for the Central Universities (No.2682021ZTPY110).

References

[1] P. Radivojac, “A (not so) quick introduction to protein function prediction,” 2013.

[2] A. Villegas-Morcillo, S. Makrodimitris, R. C. H. J. van Ham, … and M. J. T. Reinders, “Unsupervised protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function,” *Bioinfomatics*, vol. 37, no. 2, pp. 162–170, Jan. 2021.

[3] The UniProt Consortium, “UniProt: the universal protein knowledgebase,” *Nucleic Acids Research*, vol. 45, no. D1, pp. D158–D169, Jan. 2017.

[4] V. Gligorijević, P. D. Renfrew, T. Kosciolek,…, and R. Bonneau , “Structure-based protein function prediction using graph convolutional networks,” *Nature Communications*, vol. 12, no. 1, May 2021, doi: 10.1038/s41467-021-23303-9.

[5] N. Zhou, Y. Jiang, T. R. Bergquist, ... and I. Friedberg, “The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens,” Genome biology, 20(1): 1-23, 2019.

[6] G. Zhou, J. Wang, X. Zhang, M. Guo, and G. Yu, “Predicting functions of maize proteins using graph convolutional network,” *BMC Bioinformatics*, vol. 21, no. S16, Dec. 2020.

[7] G. Yu, H. Rangwala, C. Domeniconi, G. Zhang, and Z. Yu, “Protein Function Prediction with Incomplete Annotations,”*IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 3, pp. 579–591, May 2014.

[8] G. Yu, H. Rangwala, C. Domeniconi, G. Zhang, and Z. Yu, “Protein Function Prediction Using Multilabel Ensemble Classification,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 10, no. 4, pp. 1045–1057, Jul. 2013.

[9] D. Cozzetto, F. Minneci, H. Currant, and D. T. Jones, “FFPred 3: feature-based function prediction for all Gene Ontology domains,” *Scientific Reports*, vol. 6, no. 1, p. 31865, Aug. 2016.

[10] K. Choi, Y. Lee, C. Kim, and M. Yoon, “An Effective GCN-based Hierarchical Multi-label classification for Protein Function Prediction,” *arXiv:2112.02810*, Dec. 2021.

[11] S. Altschul, “Basic Local Alignment Search Tool,” *Journal of Molecular Biology*, vol. 215, no. 3, pp. 403–410, Oct. 1990.

[12] S. R. Eddy, “A new generation of homology search tools based on probabilistic inference,”*Genome Informatics. International Conference on Genome Informatics*, vol. 23, no. 1, pp. 205–211, Oct. 2009

[13] M. N. Wass, G. Barton, and M. J. E. Sternberg, “CombFunc: predicting protein function using heterogeneous data sources,” *Nucleic Acids Research*, vol. 40, no. W1, pp. W466–W470, May 2012.

[14] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[15] M. Kulmanov, M. A. Khan, and R. Hoehndorf, “DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier,” *Bioinformatics*, vol. 34, no. 4, pp. 660–668, Feb. 2018, doi: 10.1093/bioinformatics/btx624.

[16] S. Unsal, H. Atas, M. Albayrak, K. Turhan, A. C. Acar, and T.

Doğan, “Learning functional properties of proteins with language models,” *Nature Machine Intelligence*, vol. 4, no. 3, pp. 227–245, Mar. 2022, doi: 10.1038/s42256-022-00457-9.

[17] B. Lai and J. Xu, “Accurate protein function prediction via graph attention networks with predicted structure information,” *Briefings in Bioinformatics*, vol. 23, no. 1, Jan 2022.

[18] A. Mitchell, T. Attwood, P. Babbitt ... and RD. Finn, “InterPro in 2019: improving coverage, classification and access to protein sequence annotations,” *Nucleic acids research*, vol. 47, no. D1, pp. D351-D360, Jan. 2019.

[19] N. Weinhold, O. Sander, F. S. Domingues, T. Lengauer, and I. Sommer, “Local Function Conservation in Sequence and Structure Space,” *PLoS Computational Biology*, vol. 4, no. 7, p. e1000105, Jul. 2008, doi: 10.1371/journal.pcbi.1000105.

[20] E. Krissinel, “On the relationship between sequence and structure similarities in proteomics,” *Bioinformatics*, vol. 23, no. 6, pp. 717–723, Jan. 2007, doi: 10.1093/bioinformatics/btm006.

[21] 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.

[22] S. J. Giri, P. Dutta, P. Halani, and S. Saha, “MultiPredGO: Deep Multi-Modal Protein Function Prediction by Amalgamating Protein Structure, Sequence, and Interaction Information,” *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 5, pp. 1832-1838, May 2021.

[23] A. Amidi, S. Amidi, D. Vlachakis, V. Megalooikonomou, N. Paragios, and E. I. Zacharaki, “EnzyNet: enzyme classification using 3D convolutional neural networks on spatial representation,” *PeerJ*, vol. 6, p. e4750, May 2018, doi: 10.7717/peerj.4750.

[24] T. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” *ICLR*, 2017 .

[25] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio’, and Y. Bengio, “Graph Attention Networks,” *ArXiv*, 2017.

[26] M. Steinegger, M. Meier, M. Mirdita, H. Vöhringer, S. J. Haunsberger, and J. Söding, “HH-suite3 for fast remote homology detection, and deep protein annotation,” *BMC Bioinformatics*, vol. 20, no. 1, Sep. 2019.

[27] J. Xu, M. McPartlon, and J. Li, “Improved protein structure prediction by deep learning irrespective of co-evolution information,” *Nature Machine Intelligence*, vol. 3, no. 7, pp. 601–609, May 2021.

[28] E. Boutet, D. Lieberherr, M. Tognolli, M. Schneider, and A. Bairoch, “UniProtKB/Swiss-Prot,” *Methods in molecular biology (Clifton, N.J.)*, vol. 406, pp. 89–112, Jan. 2007.

[29] L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li, “CD-HIT: accelerate for clustering the next-generation sequencing data,” *Bioinformatics*, vol. 28, no. 23, pp. 3150–3152, Oct. 2012.

[30] J. M. Dana, A. Gutmanas, N. Tyagi, G. Qi, C. O’Donovan, M. Martin, and S. Velankar, “SIFTS: updated Structure Integration with Function, Taxonomy and Sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins,” *Nucleic Acids Research*, vol. 47, no. D1, pp. D482–D489, Nov. 2018, doi: 10.1093/nar/gky1114.

[31] 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, Dec. 2011, doi: 10.1038/nmeth.1818.

[32] 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, p. e2016239118, Apr. 2021.

[33] J. Lee, I. Lee, and J. Kang, “Self-attention graph pooling,” In *International conference on machine learning*, pp. 3734-3743, PMLR, 2019.

[34] A. Paszke, S. Gross, F. Massa…and S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” arXiv.org, 2019. https://arxiv.org/abs/1912.01703.

[35] M. Fey and J. E. Lenssen, “Fast Graph Representation Learning with PyTorch Geometric,” arXiv:1903.02428 [cs, stat], Apr. 2019, Available: https://arxiv.org/abs/1903.02428.

[36] J. Du, S. Zhang, G. Wu, J. M. F. Moura, and S. Kar, “Topology Adaptive Graph Convolutional Networks,” *arXiv:1710.10370*, Feb. 2018.

[37] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How Powerful are Graph Neural Networks?,” *ICLR*, 2019.

[38] W. L. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,“ *Advances in neural information processing systems* 30, 2017.

[39] S. Brody, U. Alon, and E. Yahav, “How attentive are graph attention networks?,” *ArXiv* abs/2105.14491, 2021.

[40] H. Gao, and S. Ji, ”Graph u-nets, ’’*IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 4948-4960, 2019.

[41] F. Diehl, “Edge contraction pooling for graph neural networks,” *ArXiv* abs/1905.10990, 2019.

[42] T. Bepler and B. Berger, “Learning protein sequence embeddings using information from structure,” arXiv.org, Oct. 16, 2019. https://arxiv.org/abs/1902.08661 (accessed Jun. 29, 2023).

[43] R. D. Finn, A. Bateman, J. Clements…M. Punta, “Pfam: the protein families database,” Nucleic Acids Research, vol. 42, no. D1, pp. D222–D230, Nov. 2013.

[44] A. Lu, H. Zhang, M. Ghassemi, and A. Moses, “Self-Supervised Contrastive Learning of Protein Representations By Mutual Information Maximizatio,” Europe PMC, 2020.

[45] S. Min, S. Park, S. Kim, H.-S. Choi, B. Lee, and S. Yoon, “Pre-Training of Deep Bidirectional Protein Sequence Representations with Structural Information,” arXiv.org, Sep. 16, 2021.

[46] M. Heinzinger, A. Elnaggar, Y. Wang, C. Dallago, D. Nechaev, F. Matthes and B. Rost, “Modeling aspects of the language of life through transfer-learning protein sequences,” BMC Bioinformatics, vol. 20, no. 1, Dec. 2019.

Liyuan Zhang received the B.Eng degree from the Zhengzhou University, Zhengzhou, China, in 2021, where he is currently working toward the master’s degree with the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China.

Yongquan Jiang is an Associate Researcher of the Artificial Intelligence Research Institute, Southwest Jiaotong University, Chengdu, China. His research interests include artificial intelligence, and bioinformatics. He received the Ph.D. degree from Southwest Jiaotong University.

Yan Yang is a Professor and the Vice Dean of the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China. Her research interests include artificial intelligence, big data analysis and mining, ensemble learning, and cloud computing. She received the Ph.D. degree from Southwest Jiaotong University. She is a Member of IEEE and ACM.