BMC Bioinformatics

RESEARCH Open Access

# SEGT-GO: a graph transformer method based on PPI serialization and explanatory artificial intelligence for protein function prediction

Yansong Wang[1], Yundong Sun[1,2], Baohui Lin[3], Haotian Zhang[1], Xiaoling Luo[4], Yumeng Liu[3], Xiaopeng Jin[3*] and Dongjie Zhu[1*]

*Correspondence:
jinxiaopeng.it@gmail.com;
zhudongjie@hit.edu.cn

[1] School of Computer Science and Technology, Harbin Institute of Technology Weihai Campus, Weihai 264209, China
[2] Department of Electronic Science and Technology, Harbin Institute of Technology, Harbin 150001, China
[3] College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China
[4] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

## Abstract

**Background:** A massive amount of protein sequences have been obtained, but their functions remain challenging to discern. In recent research on protein function prediction, Protein-Protein Interaction (PPI) Networks have played a crucial role. Uncovering potential function relationships between distant proteins within PPI networks is essential for improving the accuracy of protein function prediction. Most current studies attempt to capture these distant relationships by stacking graph network layers, but performance gains diminish as the number of layers increases.

**Results:** To further explore the potential functional relationships between multi-hop proteins in PPI networks, this paper proposes SEGT-GO, a Graph Transformer method based on PPI multi-hop neighborhood Serialization and Explainable artificial intelligence for large-scale multispecies protein function prediction. The multi-hop neighborhood serialization maps multi-hop information in the PPI Network into serialized feature embeddings, enabling the Graph Transformer to learn deeper functional features within the PPI Network. Based on game theory, the SHAP eXplainable Artificial Intelligence (XAI) framework optimizes model input and filters out feature noise, enhancing model performance.

**Conclusions:** Compared to the advanced network method DeepGraphGO, SEGT-GO achieves more competitive results in standard large-scale datasets and superior results on small ones, validating its ability to extract functional information from deep proteins. Furthermore, SEGT-GO achieves superior results in cross-species learning and prediction of the functions of unseen proteins, further proving the method's strong generalization.

**Keywords:** Protein function prediction, Graph transformer, PPI networks, Multi-hop neighborhood serialization, Explainable artificial intelligence

## Introduction

As fundamental constituents of cellular organisms, proteins play crucial roles [1]. The number of known protein sequences has significantly increased[1] thanks to the development of high-throughput protein sequencing technology [2]. However, due to the high cost of traditional biochemical methods [3], only a tiny proportion of protein sequences have credible functional annotations (0.002% in UniProtKB). To standardize protein function annotation, the Gene Ontology (GO) knowledgebase includes more than 42,000 terms (as of January 2024),[2] spanning the three biological ontology domains of Molecular Function Ontology (MFO), Biological Process Ontology (BPO), and Cellular Component Ontology (CCO). The GO knowledgebase aggregates annotation data for more than 5000 species, yet merely 0.03% possess more than 1000 annotations. Therefore, a reliable and accurate protein function prediction method is a pressing need to address the deficiency in protein functional annotation [4, 5].

Thanks to advances in deep learning technology, models for predicting protein function or structure have significantly aided the elucidation of pathogenic mechanisms and the advancement of novel pharmaceuticals [6, 7]. Qiao et al. [6] explored the relationship between sequence mutations and functional impairments by modeling the energy landscape of proteins, helping researchers understand the mechanisms of the disease at the molecular level. The empirical scoring function DockTScore [7] based on machine learning evaluates the binding affinity between proteins and enzymes, facilitating the efficient design of targeted drugs. By integrating protein language models and pre-training techniques, DeepP450 [8] accurately predicts cytochrome P450 enzymes (CYPs), significantly improving the success rate of drug development. Thus, developing a protein function annotation model will help researchers understand complex biological activities from a molecular functional perspective and optimize clinical treatment approaches.

Researchers typically predict protein functions using sequence data, such as residue sequences, or network data, such as Protein-Protein Interaction (PPI) Networks [9]. Sequence-based methods achieve protein function prediction by extracting latent information from residue sequences. Richa et al. [10] proposed a semi-supervised learning paradigm based on autoencoders to learn protein functional features from residue sequences for function prediction. DeepNF [11] learns high-dimensional representations from multiple heterogeneous PPI Networks for protein function term prediction, demonstrating the positive impact of PPI Network information on protein function prediction tasks. However, since these methods are limited to a single data type, models may face limitations in feature clues when inferring protein functions, potentially leading to challenges in protein function prediction accuracy. To further improve the accuracy of protein function prediction, many researchers [9] have begun incorporating sequence features as initial embeddings for network nodes and using network modeling to learn hybrid information from various types of data.

Research in graph representation learning [12] has demonstrated that Graph Convolutional Networks (GCNs) [12] are well suited to model graph data in non-Euclidean spaces. In protein function prediction tasks, various data, including PPI and protein

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

[2] https://geneontology.org/stats.html

structure, are represented in graph form. Consequently, many researchers have begun to explore GCN-based models to enhance protein function prediction capabilities [13]. DeepGraphGO [13] employs InterPro features derived from InterProScan as initial embeddings for nodes in the PPI Network and models protein interactions using a 2-layer GCN. By integrating attention mechanisms into the GCN, DeepHGAT [14] achieves improved prediction accuracy; however, the message-passing range within the graph remains confined to 3 hops. These GCN-based approaches face inherent limitations in capturing multi-hop neighborhood information [15]. As shown in Fig. 1a, GCN-based methods propagate neighborhood information layer by layer along a predetermined network structure. Thus, when stacking multiple layers of GCN to capture distant protein node information, the model tends to produce similar embeddings for the PPI nodes, resulting in a sharp decline in protein function prediction accuracy [15]. However, recent advances [16–18] in graph representation learning indicate that the ability of graph representation models to capture multi-hop neighborhood information substantially influences their performance. Similarly, in protein function prediction tasks, enhancing the model's ability to learn multi-hop neighborhood information in PPI Networks can improve the accuracy of protein function predictions.

To address the challenges of multi-hop neighborhood learning in graph representation, researchers have developed Graph Transformers, inspired by Transformer-based modeling methods from Natural Language Processing (NLP) [19, 20] and Computer Vision (CV) [21, 22]. Experimental results [16–18] show that node embeddings learned using the Transformer Encoder exhibit superior performance in downstream tasks such as node classification and node clustering. Therefore, leveraging Graph Transformers for modeling multi-hop neighborhoods in protein graphs such as PPI Networks can further enhance the model's protein function prediction performance. However, the inherent quadratic computational complexity of Graph Transformers leads to an exponential increase in computational resource requirements as the scale of the graph grows [18]. This significantly limits the application of Graph Transformers on large-scale networks; specifically, PFreshGO [23], which utilizes Transformers to model GO term Directed Acyclic Graphs (DAGs), experiences GPU memory overflow in the BPO dataset, which comprises 1943 GO terms, due to its prohibitively high computational demands. Therefore, reducing the computational complexity of the Graph Transformer to enable it to collect multi-hop information from large-scale PPI Networks for enhancing protein prediction accuracy is a critical issue that requires attention.

With the above questions, this paper proposes SEGT-GO, a Graph Transformer method based on PPI multi-hop neighborhood Serialization and Explanatory artificial intelligence for large-scale, multispecies protein function prediction. Figure 1b shows that SEGT-GO employs a Graph Transformer Encoder to learn multi-hop neighborhood information in PPI networks and capture potential functional relationships between distant proteins. By integrating PPI multi-hop neighborhood serialization and eXplainable Artificial Intelligence (XAI) [4, 24] techniques, SEGT-GO effectively mines multi-hop neighborhood information within large-scale PPI networks. The contributions of this paper can be summarized as follows:

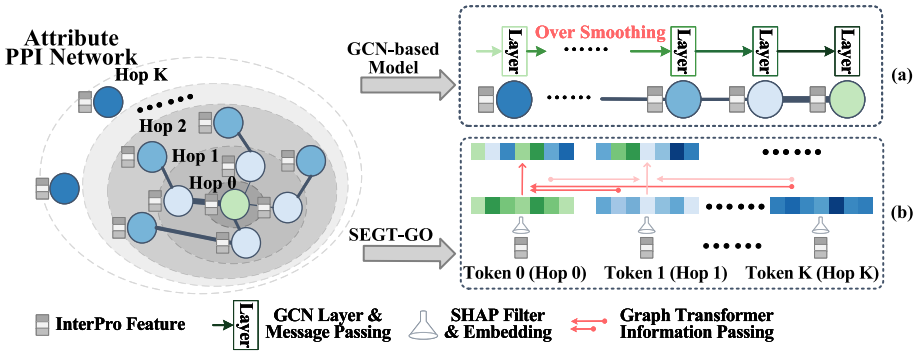


**Fig. 1** Difference between GCN-based protein function prediction model and SEGT-GO. **a** When learning information from distant neighborhoods, GCN-based models are constrained by over-smoothing. **b** SEGT-GO ensures the propagation of neighborhood information decoupled from the PPI Network, enabling learning distant neighborhood information. SHAP Filter & Embedding is responsible for filtering and mapping the InterPro Feature

- We propose a novel Graph Transformer utilizing PPI multi-hop neighborhood serialization encoding to construct serialization feature embeddings computable by the Graph Transformer. This approach enables SEGT-GO to operate efficiently on large-scale PPI networks, ensuring acceptable computational resource consumption and minimal information loss.
- We introduce the SHAP framework, grounded in game theory, to investigate the potential of eXplainable Artificial Intelligence (XAI) in protein function prediction tasks. Experiments demonstrate that SHAP effectively differentiates the contributions of various features to function prediction, further enhancing SEGT-GO's performance.
- Compared to the advanced network method DeepGraphGO, SEGT-GO demonstrates competitive performance on standard large-scale datasets and outperforms small-scale datasets, validating its ability to extract deep functional information from PPI networks. Furthermore, its predictive capacity for cross-species and unseen protein functions has also been confirmed in experiments, indicating a strong generalizability in different scenarios.

## Methods

### Data preparation

#### *Protein representation*

- **PPI network**

In the PPI Network, each node represents a protein. The weighted edges in the PPI Network indicate the interaction relationships between two proteins. The PPI Network has also been successfully applied in DeepGraphGO [13] and HNetGO [25]. The PPI Network used in this study is downloaded from v11.0 of the STRING database.

- **InterPro feature**

**Table 1** Data statistics of proteins of MFO, BPO, and CCO ontologies on the train, valid, and test sets

| Dataset | Ontology | Human | Mouse | All data | Ours |
|---|---|---|---|---|---|
| Train | MFO | 10458 | 8331 | 51549 | 35092 |
| | BPO | 12095 | 9927 | 85104 | 54276 |
| | CCO | 18842 | 8482 | 76098 | 48093 |
| Valid | MFO | 86 | 103 | 490 | 490 |
| | BPO | 138 | 299 | 1570 | 1570 |
| | CCO | 137 | 228 | 923 | 923 |
| Test | MFO | 41 | 65 | 426 | 426 |
| | BPO | 87 | 156 | 925 | 925 |
| | CCO | 767 | 130 | 1224 | 1224 |

By integrating more than a dozen different protein feature databases such as CATH-Gene3D [26], CDD [27] and Pfam [28], InterPro has become a leading resource pool for protein families, domains, and functional sites. The InterProScan [29] tool can generate a $d$-dimensional vectorized non-redundant protein sequence feature based on InterPro, known as InterPro Feature. We extract protein sequence data from UniProtKB to generate InterPro Features [30].

#### *Datasets*

The protein data and GO terms used in this study come from open-source work [13, 31], so no ethical approval is required. You can download the dataset directly from the Web.[3] Table 1 shows the data statistics for the train, valid, test sets, and species-specific subsets. We extract annotations from the GO experiment and divide the dataset following CAFA standards [5] and DeepGraphGO practices. SEGT-GO and DeepGraphGO use training samples in both the PPI Network and residue sequence, while the baselines use all training samples. Figure 2 shows that MFO, BPO and CCO have 6640, 21288, and 2729 GO terms, respectively. The broader coverage of GO terms of SEGT-GO distinguishes it from other methods [1, 2, 23, 32].

We analyze the scale of the PPI Network, as shown in Fig. 2, indicating that our study falls under large-scale graph representation learning. We also normalize the initial PPI Network to avoid degree bias and gradient vanishing during model training [33].

### The proposed methodology: SEGT-GO

#### *Serializing the neighborhood of PPI networks based on matrix multiplication*

As shown in Fig. 3a, inspired by the Hop2Token in NAGphormer [33], SEGT-GO utilizes matrix multiplication-based multi-hop neighborhood serialization encoding to compute PPI multi-hop neighborhood feature sequences. Specifically, we first obtain the adjacency matrix $A \in \mathbb{R}^{N\times N}$ from a PPI Network with $N$ nodes, where $A_{ij} = w_{ij}$. If $w_{ij} \neq 0$, it means an interaction between the protein nodes $i$ and $j$, with a corresponding interaction weight of $w_{ij}$. The normalized adjacency matrix $\hat{A}$ is computed as

[3] [Online]. Available: https://github.com/yourh/DeepGraphGO, http://bliulab.net/CFAGO/

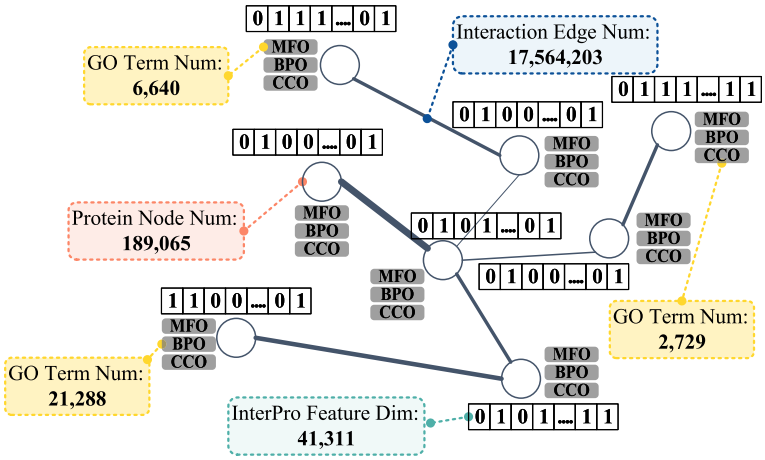


**Fig. 2** Statistical information of the PPI Network. The number of protein nodes, edges, and the InterPro Feature dimension are given using orange, blue, and green dashed boxes, respectively. Edges of different thicknesses indicate different weights. Yellow dashed boxes give the number of GO terms on different ontologies

$\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$, where $\tilde{A}$ is the adjacency matrix $A$ with self-loops, and $\tilde{D}$ represents the degree matrix. We use InterProScan [29] to generate the initial InterPro Feature $H \in \mathbb{R}^{N\times d}$, where $d = 41,311$ is the dimension of the InterPro Feature. By repeatedly left-multiplying $H$ with $\hat{A}$ $K$ times, we obtain a sequence $S$ consisting of $K+1$ neighborhood feature matrices, ordered from nearest to farthest:

$$S = \{H_k | H_k = \hat{A}H_{k-1}, k = 1, ..., K\} \tag{1}$$

where $H_0 = H$ is defined as the initial feature and $H_k \in \mathbb{R}^{N\times d}$ represents the $k$-hop neighborhood feature matrix, with $K$ denoting the maximum neighborhood aggregation range. Extracting the $v$-th row from each feature matrix in the sequence $S$ yields the PPI multi-hop neighborhood token sequence for node $v$, denoted as $T_v = \{H_v^0, H_v^1, \ldots, H_v^K\}$.

The matrix multiplication-based PPI multi-hop neighborhood serialization encoding, an efficient offline nonparametric method, offers the following advantages: 1) SEGT-GO can extract information from a broader neighborhood range within the PPI Network, enhancing function prediction precision. 2) By generating offline PPI multi-hop neighborhood feature sequences that support a mini-batch strategy, SEGT-GO can scale to large-scale PPI Networks.

### *Encoding, aggregating, and classing of PPI serialization information based on graph transformer*

- **Transformer encoder-based PPI serialized information encoding**

As shown in Fig. 3b Embedding Layer, to retain the information contained in the initial InterPro Feature, the sequence $T_v$ must undergo feature space mapping in the Embedding Layer before feature extraction with the Transformer Encoder:

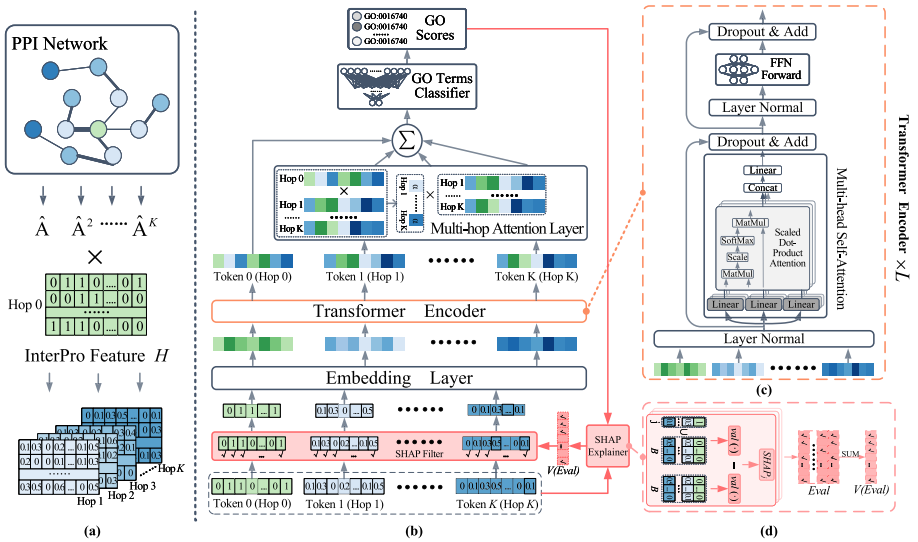


**Fig. 3** The schematic of SEGT-GO. **a** The PPI multi-hop neighborhood serialization encoding module. The normalized multi-hop adjacency matrix from the PPI Network is multiplied by the InterPro Feature $H$ to obtain the serialized PPI multi-hop neighborhood feature tokens. **b** SEGT-GO architecture diagram. The serialized PPI multi-hop neighborhood tokens are filtered by the SHAP Filter and fed into the Embedding Layer for feature mapping. The Transformer Encoder captures potential functional relationships between PPI multi-hop neighborhoods. The Multi-hop Attention Layer aggregates neighborhood information based on the importance of different hops relative to Hop 0. The GO Terms Classifier predicts protein functions using the aggregated information. The SHAP Explainer analyzes the predictions and provides an evaluation vector $V(Eval)$. **c** Detailed structure of the Transformer Encoder. $L$ determines the number of layers in the Transformer Encoder. **d** SHAP Explainer workflow. The SHAP Explainer uses the value function $val(\cdot)$ to assess the impact of various features on protein function prediction. The final feature evaluation values ($Eval$) are summed to produce the feature filter's importance evaluation vector $V(Eval)$

$$
\begin{aligned}
Z_v^{(0)} &= \left[\mathrm{Pro}(H_v^0); \mathrm{Pro}(H_v^1); \ldots; \mathrm{Pro}(H_v^K)\right] \\
\mathrm{Pro}(H_v^k) &= H_v^k M_{\mathrm{Pro}}
\end{aligned}
\tag{2}
$$

Let $Z_v^{(0)} \in \mathbb{R}^{(K+1)\times d_h}$ denote the matrix formed by the mapped token sequence, and $M_{\mathrm{Pro}} \in \mathbb{R}^{d\times d_h}$ represent a learnable vector matrix. In other words, each dimension of the InterPro Feature corresponds to a learnable vector in $M_{\mathrm{Pro}}$, and these vectors are weighted and summed along the column direction using the initial InterPro Feature $H_v^k$ as weights.

Following the design of the vanilla Transformer Encoder Block [19, 34], we constructed the Transformer Encoder, as shown in Fig. 3c, using LayerNorm (LN), Multi-head Self-Attention (MSA), position-wise Feed-Forward Network (FFN), and Dropout layer (Drop):

$$
\begin{aligned}
Z_v^{*(l)} &= \mathrm{Drop}\left(\mathrm{MSA}\left(\mathrm{LN}\left(Z_v^{(l-1)}\right)\right)\right) + Z_v^{(l-1)} \\
Z_v^{(l)} &= \mathrm{Drop}\left(\mathrm{FFN}\left(\mathrm{LN}\left(Z_v^{*(l)}\right)\right)\right) + Z_v^{*(l)}
\end{aligned}
\tag{3}
$$

where $Z_v^{(l)} \in \mathbb{R}^{(K+1)\times d_h}$ represents the multi-hop neighborhood feature encoded by the $l$-th layer of the Transformer Encoder. $Z_v^{*(l)}$ denotes an intermediate result, with the hyperparameter $l \in \{1, \ldots, L\}$ determining the number of Transformer Encoder layers. The Multi-head Self-Attention (MSA) mechanism, fundamental to the Transformer

Encoder, is crucial for learning high-quality protein embeddings. As shown in Fig. 3c, Multi-head Self-Attention (MSA) is described as multiple single-attention heads operating in independent feature spaces. For ease of description, we use a single-head self-attention mechanism as an example to explain it:

$$\begin{aligned} Que &= Z_v^{(l)} W^{Que},\ Key = Z_v^{(l)} W^{Key},\ Val = Z_v^{(l)} W^{Val} \\ Z'^{(l)}_v &= soft\max\left(\frac{Que \cdot Key^\top}{\sqrt{d_{Key}}}\right) Val \end{aligned} \tag{4}$$

where we use three learnable matrices, $W^{\mathrm{Que}} \in \mathbb{R}^{d_h \times d_{\mathrm{Key}}}$, $W^{\mathrm{Key}} \in \mathbb{R}^{d_h \times d_{\mathrm{Key}}}$, and $W^{\mathrm{Val}} \in \mathbb{R}^{d_h \times d_{\mathrm{Val}}}$, to project $Z_v^{(l)}$ into the Que, Key and Val feature spaces, respectively. Scaled Dot-Product Attention learns the potential function relationships between different neighborhoods and quantifies this information using a row-wise softmax function. $Z_v'^{(l)} \in \mathbb{R}^{(K+1) \times d_h}$ represents the final output of the Multi-head Self-Attention (MSA) in the $l$-th layer of the Transformer Encoder.

- **Attention-based aggregation of encoded PPI multi-hop neighborhood information**

Inspired by NAGphormer [33] and GAT [35], we adopt an attention-based multi-hop neighborhood feature aggregation method to effectively aggregate the PPI multi-hop neighborhood feature sequence output by the Transformer Encoder, $Z_v = Z_v^{(L)} = [Z_{v,0}; Z_{v,1}; \ldots; Z_{v,K}]$. As shown in Fig. 3b, the Multi-hop Attention Layer can learn the importance coefficients $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_K)$ of other tokens relative to Token 0 (Hop 0). We set $\alpha_0 = 1$. The importance coefficient $\alpha_{v,k}$ of the $k$-hop neighborhood $Z_{v,k}$ (Token $k$) relative to node $v$ ($Z_{v,0}$, Token 0) is calculated as follows:

$$\alpha_{v,k} = \frac{\exp\left((Z_{v,0} \parallel Z_{v,k})E^\top\right)}{\sum_{i=1}^{K} \exp\left((Z_{v,0} \parallel Z_{v,i})E^\top\right)} \tag{5}$$

where $E^\top \in \mathbb{R}^{1 \times 2d_h}$ denotes the learnable weight matrix. SEGT-GO employs $\alpha$ as summation weights to achieve PPI multi-hop neighborhood feature aggregation. This process can be expressed as:

$$Z_{v,\mathrm{final}} = Z_{v,0} + \sum_{j=1}^{K} \alpha_{v,j} Z_{v,j} \tag{6}$$

where $Z_{v,\mathrm{final}} \in \mathbb{R}^{1 \times d_h}$ represents the aggregated PPI multi-hop neighborhood feature result corresponding to node $v$.

- **Large-scale protein function classifier based on aggregated PPI multi-hop neighborhood information**

As shown in Fig. 3b GO Terms Classifier, to narrow the gap between the hidden embedding dimension $d_m$ and the number of extensive GO terms $C$, we have devised a classifier based on transitional Multi-Layer Perceptron (trans-MLP) following experimentation. Specifically, trans-MLP consists of 4 layers with neurons set to $d_m$, $C/4$, $C/2$, and $C$, respectively. We define trans-MLP as follows:

$$\hat{y}_v = \text{trans-MLP}(Z_{v,\text{final}}) \tag{7}$$

where $\hat{y}_v \in \mathbb{R}^{1\times C}$ represents the confidence of node/protein $v$ being annotated with different GO terms.

Finally, we optimize the multi-label binary classification problem modeled by SEGT-GO using binary cross-entropy loss:

$$Loss = -\frac{1}{BC}\sum_{i=1}^{B}\sum_{j=1}^{C}\left[y_{ij}\log(\hat{y}_{ij}) + (1-y_{ij})\log(1-\hat{y}_{ij})\right] \tag{8}$$

where $B$ represents the number of protein samples, $\hat{y}_{ij}$ denotes the probability of the $i$-th protein being predicted as GO term $j$, and $y_{ij} \in \{0,1\}$ is the ground truth.

The time and space complexity of SEGT-GO is $O(N(K+1)^2 d_h)$ and $O(B(K+1)^2 + B(K+1)^2 d_h + {d_h}^2 L)$, respectively, where $N$ represents the number of nodes, $K$ is the maximum neighborhood aggregation range, $d_h$ is the dimension of the hidden layer vectors, $L$ is the number of layers in the Graph Transformer, and $B$ denotes the number of protein samples in each batch. For detailed theoretical analysis and runtime statistics, see Appendix C.

#### *Enhancing SEGT-GO with explainable framework SHAP for feature optimization*

Given the possibility of negative features in high-dimensional InterPro features (as shown in Fig. 2) that may interfere with functional prediction, we incorporate SHapley Additive exPlanations (SHAP) [24], a framework based on game theory eXplainable Artificial Intelligence (XAI) [36], to identify features that positively contribute to function prediction and mask negative ones. First, we present the calculation method for the Shapley values:

$$\text{SHAP}_j = \sum_{B\subseteq P\setminus\{j\}} \frac{|B|!(P-|B|-1)!}{|B|!}\Big(\text{val}(B\cup\{j\}) - \text{val}(B)\Big) \tag{9}$$

where $\text{SHAP}_j$ represents the Shapley value corresponding to a feature point $j$ of the InterPro Feature (player, in game theory terminology, the same applies hereafter). When $\text{SHAP}_j > 0$, it indicates that the feature point $j$ positively impacts the prediction of the model. $P$ denotes the set of all feature points (the coalition of all players), $B$ represents a subset of feature points (a coalition), $|B|$ indicates the number of feature points in the set (the size of the coalition), ! is the factorial operation, and $\text{val}(\cdot)$ is used to evaluate the contribution of feature points to the prediction result (value function).

To reduce the computational cost of SHAP, we control the number of training samples $a$, testing samples $b$, and focus categories $\theta$ that SHAP covers. The evaluation values provided by SHAP are denoted as $Eval \in \mathbb{R}^{\theta\times b\times K\times d}$. By summing $Eval$ along each dimension, we convert it into an importance evaluation vector $V(Eval) \in \mathbb{R}^d$ (as shown in Fig. 3d). We define $t \geq 0$ as the filtering threshold: when $V(Eval)_j > t$, the SHAP Filter retains the feature point $j$.

## Results

### Experiment setup

#### *Implementation details*

We train SEGT-GO to predict test protein samples in different ontologies, optimizing hyperparameters in the valid set for best performance. Specifically, we try the maximum propagation range for PPI multi-hop neighborhoods $K$ in $\{1, \ldots, 9\}$, the number of Transformer Encoder layers $L$ in $\{1, \ldots, 5\}$, the number of heads in $\{4, 8, 16\}$, the hidden dimension $d_h$ in $\{128, 256, 512, 1024\}$, the dropout rate for the Dropout Layers in $\{0.1, 0.3, 0.5\}$, and the SHAP filtering threshold $t$ in $\{0, 0.1, \ldots, 0.6\}$. The learning rate and weight decay for the AdamW optimizer [37] are searched within the ranges $\{1e-4, 5e-4, 1e-3, 5e-3\}$ and $\{1e-5, 5e-4, 1e-4\}$, respectively. The batch size is fixed at 256. SEGT-GO is trained for a maximum of 1000 epochs, with early stopping if there is no improvement for 30 epochs. We apply SHAP to the best SEGT-GO without SHAP (SEGT-GO$_{\text{w/o SHAP}}$) in each ontology. Considering computational efficiency, we set $a$, $b$, and $\theta$ in SHAP to 5000, 100, and 500, respectively. On an NVIDIA Tesla P40 24GB, the training durations for SEGT-GO in the MFO, BPO, and CCO ontologies are 20 min, 3 h, and 1.5 h, respectively. We organize the hyperparameter settings of the baselines according to DeepGraphGO [13] and other official baseline implementations [38–40].

#### *Baselines and evaluation criterion*

Based on the data types, we classify the 7 baselines into 2 categories: sequence-based (BLAST-KNN [39], LR-InterPro [39], DeepGOCNN [40], DeepGOPlus [40], and PO2GO [41]) and network-based (Net-KNN [38] and DeepGraphGO [13]). For details on the baselines, see Appendix A. Since SEGT-GO and DeepGraphGO use the same dataset and baselines, we directly cite the test results of DeepGraphGO [13].

Referring to other protein function prediction works [1, 13], we use AUPR (Area Under the Precision-Recall curve) and $F_{max}$ (maximum protein-centric F-measure) as evaluation metrics. For detailed calculations of these metrics, see Appendix B.

### Comparison of SEGT-GO with GCN-based method and baselines

To demonstrate the advantages of SEGT-GO over GCN-based models in mining protein networks of varying scales, we first compare SEGT-GO with the best GCN-based method in the baselines, DeepGraphGO, across various dataset scales. Moreover, evaluating models with different datasets can validate the model generalization [1]. We introduce the Human dataset from CFAGO [31] as Dataset B (refer to Appendix D). The dataset in Sect. Datasets is Dataset A.

Table 2 shows that SEGT-GO provides more accurate protein function predictions than DeepGraphGO. Across 12 test items on datasets of different scales, SEGT-GO leads in 10 items, with the highest relative performance improvement reaching 59.3% (on $F_{max}$ CCO, Dataset B). This demonstrates that SEGT-GO's advanced architecture effectively learns protein-protein interactions from a broader range within the PPI Network. Additionally, results in Fig. 4 highlight SEGT-GO's strong stability and scalability when handling PPI Networks of varying scales and structures, particularly in the smaller Dataset B. It is important to note that although SEGT-GO leads by a

**Table 2** Comparison of SEGT-GO and DeepGraphGO on datasets of different scales

| Dataset | Methods | AUPR | | | $F_{max}$ | | |
|---|---|---|---|---|---|---|---|
| | | MFO | BPO | CCO | MFO | BPO | CCO |
| A | DeepGraphGO | 0.543 | 0.194 | 0.695 | **0.623** | 0.327 | **0.692** |
| | SEGT-GO(Ours) | **0.555** | **0.217** | **0.703** | 0.619 | **0.328** | 0.683 |
| B | DeepGraphGO | 0.098 | 0.133 | 0.113 | 0.142 | 0.327 | 0.209 |
| | SEGT-GO(Ours) | **0.101** | **0.173** | **0.167** | **0.213** | **0.454** | **0.333** |

Bold for "the best"

smaller margin in AUPR and $F_{max}$ on the large-scale Dataset A compared to Dataset B, the experimental results demonstrate that SEGT-GO successfully captures a broader range of neighborhood information with less computational consumption. This proves SEGT-GO's potential for scalable application on large networks and its ability to learn distant protein functional information. For further details, please refer to Sect. Neighborhood Aggregation Range K and Appendix C.

We compare SEGT-GO with 6 baselines on dataset A. Details can be found in Table 3. As shown in Fig. 5, the conclusions can be summarized as follows:

- SEGT-GO has achieved competitive results across the 3 ontologies. Compared to the second-best, SEGT-GO exhibits AUPR improvements of 2.2%, 11.9%, and 1.2% in MFO, BPO, and CCO, respectively. This demonstrates SEGT-GO's excellent capability in multispecies function prediction. However, higher False Positive Rates (FPRs) lead to SEGT-GO being suboptimal in $F_{max}$ for MFO and CCO (see Table 3). We speculate that this may be related to the completeness of the annotation of the GO term [25]. In other words, instances predicted as counterexamples by SEGT-GO in this study might be evaluated differently in a more comprehensive annotation set.
- Combining sequence and network features improves precision in prediction. SEGT-GO and DeepGraphGO outperform other baselines across the 3 ontologies by leveraging both sequence and network information. However, the limitations of GCNs result in DeepGraphGO achieving a lower AUPR in all three ontologies than SEGT-GO. The LR-InterPro model, which uses only InterPro Features, further shows that relying solely on sequence data fails to achieve good performance across all ontologies, especially in BPO, which contains more abstract GO terms. Additionally, PO2GO, which uses the improved GO term representation PO2Vec, did not show a significant performance advantage.
- SEGT-GO achieves significant improvements in BPO, which has more complex annotations. Consistent with previous studies, baselines perform worse in BPO, indicating that improving prediction accuracy in BPO is challenging. However, SEGT-GO significantly increases AUPR in BPO (MFO: +2.2%; BPO: +11.9%; CCO: +1.2%). We believe that SEGT-GO's superior PPI multi-hop neighborhood learning enables it to derive more accurate protein embeddings from a broader range, resulting in better performance in more difficult prediction tasks.

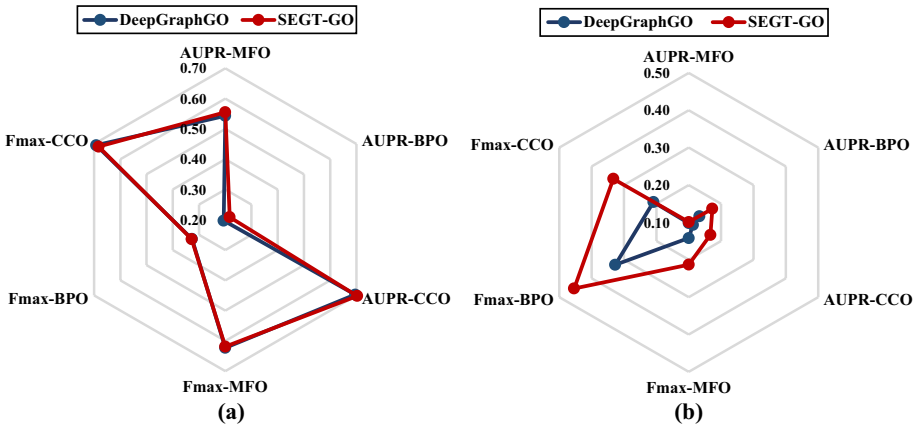


**Fig. 4** Radar charts of SEGT-GO and DeepGraphGO on datasets of different scales. **a** and **b** are datasets A and B, respectively

### Generalization studies of unseen species and cross-species

To explore the cross-species generalization of SEGT-GO, we retrain SEGT-GO on specific species datasets. Specifically, we introduce 2 variants: 1) SEGT-GO$_{\text{Species}}$: both the training and test datasets contain only the specific species; 2) SEGT-GO$_{\text{w/o Species}}$: the training dataset contains all the species except the specific one, while the test dataset contains only the specific species. The results are shown in Table 4. SEGT-GO achieves 10 best results and 2 s-best results of 12 items. Furthermore, even when the specific species is not present in the training phase, SEGT-GO$_{\text{w/o Species}}$ achieves 8 s-best performances, making it the second-best. This indicates that SEGT-GO can learn information from other species between species. Similar conclusions are also found in DeepGraphGO [13]. The analogous results between SEGT-GO and DeepGraphGO further validate the effectiveness of the cross-species learning strategy: training models using a large number of samples from other species endow them with strong cross-species generalization capabilities.

We also compare the generalization of unseen species [1] of SEGT-GO and DeepGraphGO on Human samples. Details about DeepGraphGO are in Appendix E. As shown in Fig. 6, although the performance of SEGT-GO$_{\text{w/o Human}}$ slightly decreases after excluding Human samples, it still surpasses DeepGraphGO$_{\text{w/o Human}}$ in BPO and CCO. Moreover, we observe that except for MFO, SEGT-GO$_{\text{w/o Human}}$, which never encountered Human proteins during training, outperforms DeepGraphGO trained on all species. The excellent unseen species generalization of SEGT-GO enables it to perform function prediction on new species proteins without prior knowledge.

### Methodological contribution analysis and GO terms group assessment

#### *Methodological contribution analysis*

This section addresses two questions: 1) Does the trans-MLP transition from hidden feature to functional annotation space improve prediction precision? 2) Does the SHAP successfully mitigate the negative impact of feature noise on the function prediction task? We conduct ablation experiments on the full dataset as well as on Human and Mouse test samples. Specifically, we construct two variants: 1) SEGT-GO$_{\text{w/oSHAP[NONSPACE]\&MLP}}$,

**Table 3** Performance comparison between SEGT-GO and other baselines

| Method | AUPR | | | $F_{max}$ | | |
|---|---|---|---|---|---|---|
| | MFO | BPO | CCO | MFO | BPO | CCO |
| BLAST-KNN | 0.455 | 0.113 | 0.570 | 0.590 | 0.274 | 0.650 |
| LR-InterPro | 0.530 | 0.144 | 0.672 | 0.617 | 0.278 | 0.661 |
| Net-KNN | 0.276 | 0.157 | 0.641 | 0.426 | 0.305 | 0.667 |
| DeepGOCNN | 0.306 | 0.101 | 0.573 | 0.434 | 0.248 | 0.632 |
| DeepGOPlus | 0.398 | 0.108 | 0.595 | 0.593 | 0.290 | 0.672 |
| PO2GO | 0.380 | 0.179 | 0.587 | 0.506 | 0.290 | 0.596 |
| DeepGraphGO | 0.543 | 0.194 | 0.695 | **0.623** | 0.327 | **0.692** |
| SEGT-GO(Ours) | **0.555** | **0.217** | **0.703** | 0.619 | **0.328** | 0.683 |

Bold for "the best, and underline for "the second best"

which disables both SHAP and trans-MLP; 2) SEGT-$GO_{w/o\ SHAP}$, which only disables SHAP. Details can be found in Appendix F.

As shown in Fig. 7, when SHAP and trans-MLP are disabled, SEGT-$GO_{w/oSHAP[NONSPACE]\&MLP}$ cannot effectively transition hidden features to large-scale protein function annotations. Meanwhile, it cannot filter out InterPro Feature noise, resulting in significant performance degradation. With the aid of trans-MLP, SEGT-$GO_{w/o\ SHAP}$ significantly improves the accuracy of protein function prediction and becomes the second-best model. Furthermore, the SEGT-GO that incorporates SHAP achieves the best performance. This demonstrates that SHAP accurately filters out InterPro Feature noise, negatively impacting function prediction tasks. Moreover, the successful application of SHAP highlights the potential of XAI in protein function prediction. In addition, to analyze the contribution of protein features, we replace the InterPro features with ProtBert pre-training features. The results indicate that SEGT-GO, when using InterPro features, outperforms the model with ProtBert features. Details are provided in Appendix F.

#### *Assessment on different frequency GO term groups*

To analyze SEGT-GO's sensitivity to different GO terms, we follow the approach in DeepGraphGO [13] and group the GO terms based on the number of annotations: 10–30, 31–100, 101–300 and > 300. Table 5 presents SEGT-GO's AUPR across different frequency groups of GO terms in the MFO.

The experimental results show that: 1) SEGT-GO performs excellently in both rare and high-frequency GO terms, demonstrating the effectiveness of its novel model architecture and training paradigm in handling prediction tasks for both uncommon and high-frequency GO terms; 2) SEGT-GO performs moderately in the range of 31–300 GO term frequencies, indicating a shortcoming in learning the classification features of medium-frequency GO terms, which leads to a decline in classification accuracy; 3) DeepGraphGO and LR-InterPro integrate and learn InterPro features in different ways, achieving good results in medium-frequency GO term prediction tasks, thereby confirming the effectiveness of InterPro features.

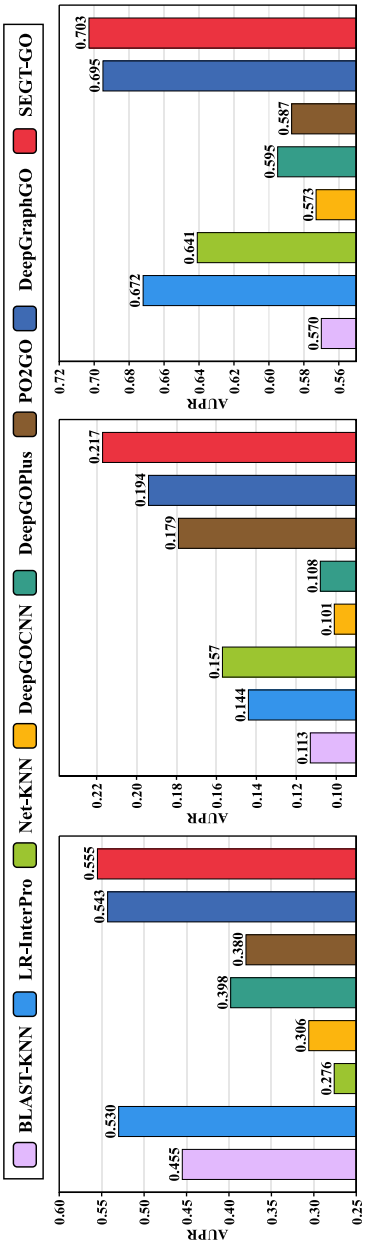


**Fig. 5** Performance comparison of SEGT-GO with other baselines on the full dataset. **a**–**c** show performance on different ontologies. SEGT-GO has achieved competitive results across all 3 ontologies

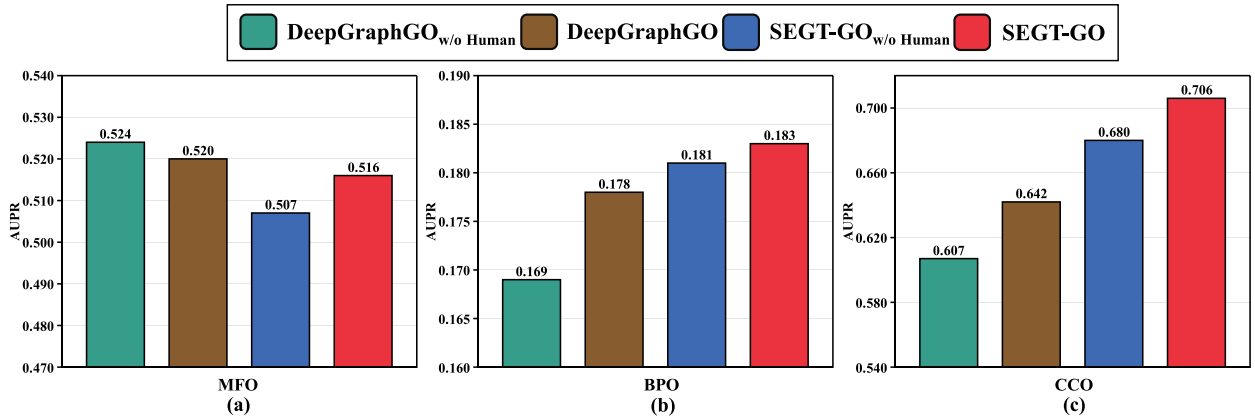


**Fig. 6** Evaluation of the unseen species generalization of DeepGraphGO and SEGT-GO on all species or all species except Human (w/o Human). **a**–**c** show performance on different ontologies. SEGT-GO exhibits great inference capabilities for unseen species, particularly in BPO and CCO

**Table 4** Performance comparison of cross-species generalization of SEGT-GO and two variants on Human (pid: 9606) and Mouse (pid: 10090) proteins

| Methods | AUPR | | | $F_{max}$ | | |
|---|---|---|---|---|---|---|
| | MFO | BPO | CCO | MFO | BPO | CCO |
| | HUMAN(9606) | | | | | |
| SEGT-GO$_{Human}$ | 0.446 | 0.171 | 0.696 | 0.575 | 0.289 | **0.711** |
| SEGT-GO$_{w/o\ Human}$ | 0.507 | 0.181 | 0.680 | 0.592 | 0.279 | 0.674 |
| SEGT-GO | **0.516** | **0.183** | **0.706** | **0.635** | **0.304** | 0.709 |
| | MOUSE(10090) | | | | | |
| SEGT-GO$_{Mouse}$ | 0.509 | 0.154 | 0.569 | **0.630** | 0.284 | 0.613 |
| SEGT-GO$_{w/o\ Mouse}$ | 0.536 | 0.163 | 0.638 | 0.610 | 0.313 | 0.632 |
| SEGT-GO | **0.621** | **0.185** | **0.641** | 0.629 | **0.324** | **0.647** |

Bold for "the best, and underline for "the second best"

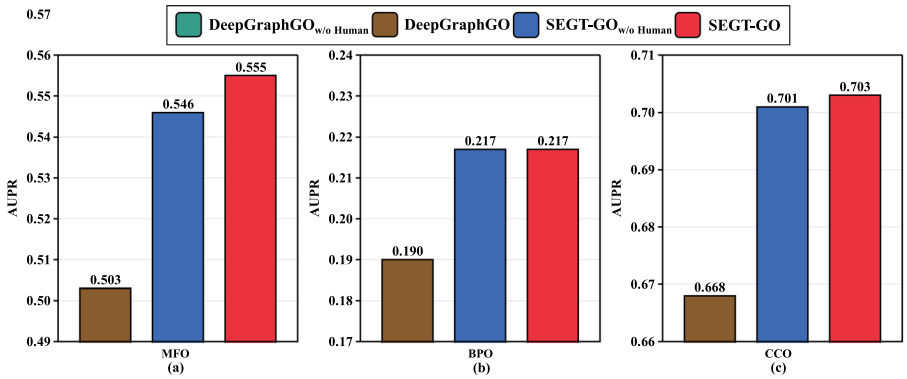


**Fig. 7** Ablation experiments of SEGT-GO and its variants on the full dataset. The variant w/o SHAP&MLP disables SHAP and trans-MLP, while the variant w/o SHAP disables only SHAP

**Table 5** AUPR of SEGT-GO and other baselines on different GO term groups in MFO

| Method | GO Term Groups | | | |
|---|---|---|---|---|
| | 10–30 | 31–100 | 101–300 | >300 |
| BLAST-KNN | 0.590 | 0.579 | 0.533 | 0.500 |
| LR-InterPro | 0.544 | **0.652** | 0.560 | 0.545 |
| Net-KNN | 0.281 | 0.371 | 0.301 | 0.273 |
| DeepGOCNN | 0.014 | 0.045 | 0.235 | 0.252 |
| DeepGOPlus | 0.309 | 0.322 | 0.414 | 0.427 |
| PO2GO | 0.018 | 0.028 | 0.046 | 0.354 |
| DeepGraphGO | 0.597 | 0.632 | **0.571** | 0.575 |
| SEGT-GO(Ours) | **0.651** | 0.573 | 0.508 | **0.585** |

Bold for "the best, and underline for "the second best"

### Visualization of potential relationships between interPro feature and GO terms

As shown in Figs. 8 and 9, to demonstrate the impact of the SHAP Explainer on protein function prediction more intuitively, we visualize the importance evaluation provided by SHAP on MFO.

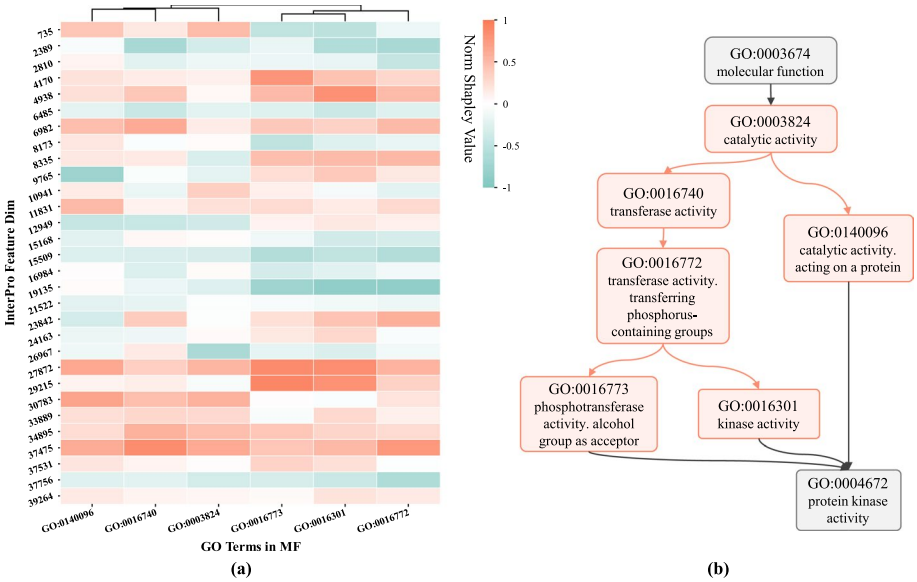


**Fig. 8** Visualization of SHAP evaluation results in MFO. **a** Normalized Shapley value heatmap including 30 randomly selected InterPro Features and 6 GO terms in MFO. Positive values represent these features that contribute to the accurate prediction, and negative values indicate adverse effects. **b** The GO term hierarchical relationships of GO:0004672 (the 6 GO terms given in **a** are highlighted in orange)

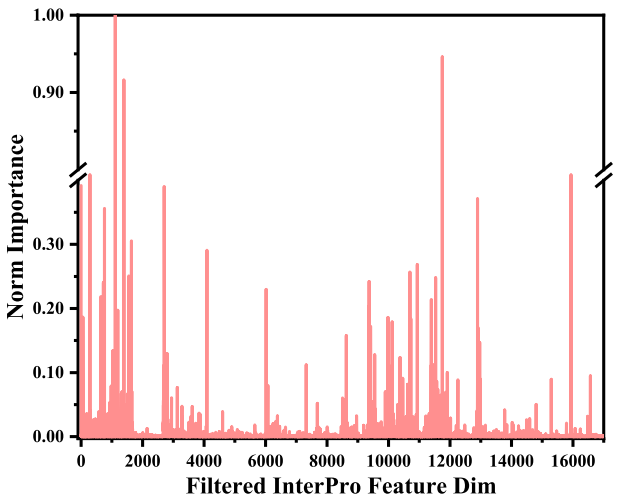


**Fig. 9** Distribution of importance evaluation values on the MFO. The horizontal coordinates represent the parts of the importance evaluation values that satisfy $t > 0$, and the vertical coordinates are the normalized importance evaluation values. InterPro Features with larger importance evaluation values are more useful for SEGT-GO prediction

As shown in Fig. 8a, the same InterPro Feature exhibits varying effects on different GO term prediction tasks (row direction). For instance, the 23842nd InterPro Feature is detrimental to the prediction of GO:0140096 and GO:0003824 but enhances the prediction of 4 other GO terms. Similarly, GO term prediction tasks are influenced by the cumulative effect of different InterPro Features (column direction). For example, the

4170th, 27872nd and 29215th InterPro Features positively contribute to the prediction of GO:0016773, whereas the 15509th and 19135th do not. This shows that SHAP can effectively identify the diverse impacts of different InterPro Features on function prediction tasks, which helps SEGT-GO in select InterPro Features.

Furthermore, we employ the Complete Linkage method [42] based on Canberra Distance [43] to cluster the 6 GO terms in Fig. 8a and construct a dendrogram. As shown in Fig. 8b, we discover a high correlation between the hierarchical relationships of the GO terms obtained from QuickGO[4] and the dendrogram. SHAP's ability to capture the hierarchical structure of GO terms in its analysis validates the reliability of its results.

As shown in Fig. 9, we visualize the distribution of InterPro Features with importance evaluation values in MFO. The distribution of importance evaluation values exhibits significant variation. This shows that SHAP effectively assists SEGT-GO in selectively extracting beneficial features from tens of thousands of InterPro features for protein function prediction.

### Hyperparameter studies

The hyperparameter studies evaluate SEGT-GO's performance under different combinations to explore its sensitivity to various hyperparameters. They provide theoretical support for other researchers who deploy SEGT-GO in different domains or datasets. This section discusses the impact of 3 critical hyperparameters: neighborhood aggregation range $K$, number of Transformer Encoder layers $L$, and input feature filtering threshold $t$. The results are shown in Fig. 10. Details can be found in Appendix G.

#### *Neighborhood aggregation range K*

We evaluate the impact of different neighborhood aggregation ranges, $K \in \{1, 2, \ldots, 9\}$, on SEGT-GO, with the results shown in Fig. 10a. We observe that the best $K$ varies between different ontologies, indicating that the information required for function prediction tasks differs between ontologies. Additionally, we find that, except for MFO, SEGT-GO's performance on BPO and CCO does not decline with increasing $K$ values; instead, there is an upward trend (BPO from $K = 8$ to $K = 9$, CCO from $K = 7$ to $K = 9$). This demonstrates the necessity of learning the multi-hop neighborhood information of the PPI to improve prediction accuracy, consistent with previous research [16–18]. We set the best $K$ for MFO, BPO, and CCO to 4, 9, and 6, respectively.

#### *Transformer encoder layers L*

We fix $K$ to its best value and evaluate the impact of the number of Transformer Encoder layers, $L \in \{1, 2, \ldots, 5\}$, on SEGT-GO's performance. The experimental results shown in Fig. 10b indicate that, except for CCO, a higher $L$ reduces the precision of the prediction. We attribute this phenomenon to an increased likelihood of overfitting. In practice, we set the best $L$ for MFO, BPO and CCO at 1, 1, and 5, respectively.

[4] https://www.ebi.ac.uk/QuickGO/term/GO:0004672

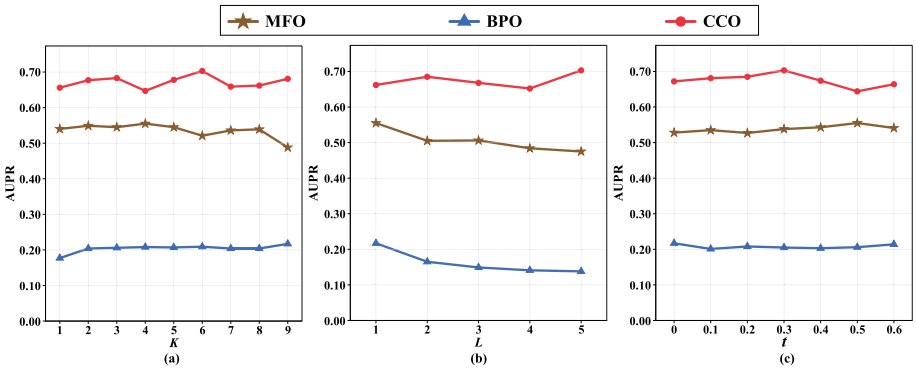


**Fig. 10** Performance of SEGT-GO when using different combinations of hyperparameters. Each subplot represents **a** the neighborhood aggregation range *K*, **b** the number of Transformer Encoder layers *L*, and **c** the filtering threshold *t*. Evaluations are performed using the full dataset

#### *Input feature filtering threshold t*

We fix $K$ and $L$ to their best values and evaluate the impact of different input feature filtering thresholds $t$ within $\{0, 0.1, \ldots, 0.6\}$. The experimental results are shown in Fig. 10c. Compared to $K$ and $L$, the variation in $t$ has a more minor impact on SEGT-GO's performance. This indicates that SEGT-GO can effectively adapt to different scales of feature spaces, helping practitioners mitigate obstacles during deployment. The best $t$ values for MFO, BPO, and CCO are set to 0.5, 0, and 0.3, respectively.

## Conclusion

This paper introduces SEGT-GO, a Graph Transformer method based on PPI multi-hop neighborhood Serialization and Explanatory artificial intelligence for large-scale, multi-species protein function prediction. The novel PPI multi-hop neighborhood serialization Graph Transformer enables SEGT-GO to effectively address the challenges GCN-based models face in learning high-quality protein node embeddings within the multi-hop neighborhood in PPI Networks. The offline generation of feature sequences supporting the mini-batch strategy allows SEGT-GO to scale to larger PPI Networks. SHAP mitigates the negative impact of feature noise, demonstrating the potential of XAI in protein function prediction. Experiments with datasets of varying scales and settings show that SEGT-GO can better exploit PPI multi-hop neighborhood information related to protein function in PPI Networks, thereby improving the prediction precision.

In comprehensive experiments, SEGT-GO demonstrates limitations in specific settings: it exhibits weaknesses when predicting GO terms within the frequency range of 31–300, leading to a decrease in classification accuracy (Sect. Assessment on Different Frequency GO Term Groups). In future work, we can improve the model's ability to learn medium-frequency GO terms by incorporating external knowledge and optimizing the model architecture. Furthermore, by comparing existing work with SEGT-GO, we identify complementary relationships among some studies. For example, incorporating features of the spatial structure could enhance SEGT-GO performance. The rise of Artificial Intelligence Generated Content (AIGC) may offer researchers a new approach to protein function prediction.

**Abbreviations**

| | |
|---|---|
| PPI Network | Protein-protein interaction network |
| XAI | eXplainable Artificial Intelligence |
| SHAP | SHapley Additive exPlanations |
| GO | Gene ontology |
| MFO | Molecular function ontology |
| BPO | Biological process ontology |
| CCO | Cellular component ontology |
| GCNs | Graph convolutional networks |
| NLP | Natural language processing |
| CV | Computer vision |
| DAGs | Directed acyclic graphs |
| trans-MLP | Transitional multi-layer perceptron |
| AUPR | Area under the precision-recall curve |
| $F_{max}$ | Maximum protein-centric F-measure |
| FPRs | False positive rates |
| AIGC | Artificial intelligence generated content |

## Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-025-06059-7.

Additional file 1

**Acknowledgements**
Not applicable.

**Author contributions**
Conceptualization, Y.W., and Y.S.; methodology, Y.W., Y.S., and X.J.; software, Y.W., B.L., and H.Z.; validation, Y.W., and H.Z.; data curation, Y.W. and H.Z.; writing—original draft preparation, Y.W.; writing—review and editing, Y.W., Y.S., B.L., H.Z., X.L., Y.L., X.J., and D.Z. All authors have read and agreed to the published version of the manuscript.

**Funding**
This work was supported by the National Natural Science Foundation of China (Grant No.62302317), the project of Shenzhen Science and Technology Innovation Committee (Grant No.JCYJ20240813141424032), the Natural Science Foundation of Top Talent of SZTU (Grant No.GDRC202319), the Foundation for Young Innovative Talents in Ordinary Universities of Guangdong (Grant No.2024KQNCX042), and the Shenzhen Colleges and Universities Stable Support Program (Grant No.20220715183602001, No.20231122005530001).

**Data availability**
The key implementations of SEGT-GO are available at https://github.com/SpaceZore/SEGT-GO.

## Declarations

**Ethics approval and consent to participate**
Not applicable.

**Consent for publication**
Not applicable.

**Competing interests**
The authors declare that they have no Conflict of interest.

Received: 5 November 2024 Accepted: 20 January 2025
Published online: 10 February 2025

**References**

1. Yuan Q, Xie J, Xie J, Zhao H, Yang Y. Fast and accurate protein function prediction from sequence through pretrained language model and homology-based label diffusion. Brief Bioinform. 2023;24(3):117.
2. Jiao P, Wang B, Wang X, Liu B, Wang Y, Li J. Struct2go: protein function prediction based on graph pooling algorithm and alphafold2 structure information. Bioinformatics. 2023;39(10):637.
3. Costanzo M, VanderSluis B, Koch EN, Baryshnikova A, Pons C, Tan G, Wang W, Usaj M, Hanchard J, Lee SD, et al. A global genetic interaction network maps a wiring diagram of cellular function. Science. 2016;353(6306):1420.
4. Jiang Y, Oron TR, Clark WT, Bankapur AR, D'Andrea D, Lepore R, Funk CS, Kahanda I, Verspoor KM, Ben-Hur A, et al. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. Genome Biol. 2016;17:1–19.
5. Zhou N, Jiang Y, Bergquist TR, Lee AJ, Kacsoh BZ, Crocker AW, Lewis KA, Georghiou G, Nguyen HN, Hamid MN, et al. The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. Genome Biol. 2019;20:1–23.

6. Qiao W, Akhter N, Fang X, Maximova T, Plaku E, Shehu A. From mutations to mechanisms and dysfunction via computation and mining of protein energy landscapes. BMC Genom. 2018;19:1–13.
7. Guedes IA, Barreto AM, Marinho D, Krempser E, Kuenemann MA, Sperandio O, Dardenne LE, Miteva MA. New machine learning and physics-based scoring functions for drug discovery. Sci Rep. 2021;11(1):3198.
8. Chang J, Fan X, Tian B. Deepp450: predicting human p450 activities of small molecules by integrating pretrained protein language model and molecular representation. J Chem Inf Model. 2024;64(8):3149–60.
9. Lin B, Luo X, Liu Y, Jin X. A comprehensive review and comparison of existing computational methods for protein function prediction. Brief Bioinform. 2024;25(4):289.
10. Dhanuka R, Tripathi A, Singh JP. A semi-supervised autoencoder-based approach for protein function prediction. IEEE J Biomed Health Inform. 2022;26(10):4957–65.
11. Gligorijević V, Barot M, Bonneau R. deepnf: deep network fusion for protein function prediction. Bioinformatics. 2018;34(22):3873–81.
12. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2022)
13. You R, Yao S, Mamitsuka H, Zhu S. Deepgraphgo: graph neural network for large-scale, multispecies protein function prediction. Bioinformatics. 2021;37(1):262–71.
14. Zhao Y, Yang Z, Wang L, Zhang Y, Lin H, Wang J. Predicting protein functions based on heterogeneous graph attention technique. IEEE J Biomed Health Inform. 2024. https://doi.org/10.1109/JBHI.2024.3357834.
15. Bose K, Das S. Can graph neural networks go deeper without over-smoothing? Yes, with a randomized path exploration! IEEE Trans Emerg Top Comput Intell. 2023;7(5):1595–604.
16. Sun Y, Zhu D, Du H, Tian Z. MHNF: multi-hop heterogeneous neighborhood information fusion graph representation learning. IEEE Trans Knowl Data Eng. 2022;35(7):7192–205.
17. Zhu J, Yan Y, Zhao L, Heimann M, Akoglu L, Koutra D. Beyond homophily in graph neural networks: current limitations and effective designs. Adv Neural Inf Process Syst. 2020;33:7793–804.
18. Sun Y, Zhu D, Wang Y, Fu Y, Tian Z. GTC: gnn-transformer co-contrastive learning for self-supervised heterogeneous graph representation. Neural Netw. 2024;181: 106645.
19. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Advances in neural information processing systems. **30** (2017)
20. Ahmadi N, Sand H, Papotti P. Unsupervised matching of data and text. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 1058–1070. IEEE (2022)
21. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N. An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
22. Pan Z, Cai J, Zhuang B. Fast vision transformers with HILO attention. Adv Neural Inf Process Syst. 2022;35:14541–54.
23. Pan T, Li C, Bi Y, Wang Z, Gasser RB, Purcell AW, Akutsu T, Webb GI, Imoto S, Song J. Pfresgo: an attention mechanism-based deep-learning approach for protein annotation by integrating gene ontology inter-relationships. Bioinformatics. 2023;39(3):094.
24. Strumbelj E, Kononenko I. An efficient explanation of individual classifications using game theory. J Mach Learn Res. 2010;11:1–18.
25. Zhang X, Guo H, Zhang F, Wang X, Wu K, Qiu S, Liu B, Wang Y, Hu Y, Li J. Hnetgo: protein function prediction via heterogeneous network transformer. Brief Bioinform. 2023;24(6):556.
26. Lewis TE, Sillitoe I, Dawson N, Lam SD, Clarke T, Lee D, Orengo C, Lees J. Gene3d: extensive prediction of globular domains in proteins. Nucl Acids Res. 2018;46(D1):435–9.
27. Marchler-Bauer A, Bo Y, Han L, He J, Lanczycki CJ, Lu S, Chitsaz F, Derbyshire MK, Geer RC, Gonzales NR, et al. Cdd/sparcle: functional classification of proteins via subfamily domain architectures. Nucl Acids Res. 2017;45(D1):200–3.
28. Finn RD, Coggill P, Eberhardt RY, Eddy SR, Mistry J, Mitchell AL, Potter SC, Punta M, Qureshi M, Sangrador-Vegas A, et al. The pfam protein families database: towards a more sustainable future. Nucleic Acids Res. 2016;44(D1):279–85.
29. Jones P, Binns D, Chang H-Y, Fraser M, Li W, McAnulla C, McWilliam H, Maslen J, Mitchell A, Nuka G, et al. Interproscan 5: genome-scale protein function classification. Bioinformatics. 2014;30(9):1236–40.
30. Consortium U. Uniprot: a worldwide hub of protein knowledge. Nucl Acids Res. 2019;47(D1):506–15.
31. Wu Z, Guo M, Jin X, Chen J, Liu B. CFAGO: cross-fusion of network and attributes based on attention mechanism for protein function prediction. Bioinformatics. 2023;39(3):123.
32. Gu Z, Luo X, Chen J, Deng M, Lai L. Hierarchical graph transformer with contrastive learning for protein function prediction. Bioinformatics. 2023;39(7):410.
33. Chen J, Gao K, Li G, He K. NAGphormer: A tokenized graph transformer for node classification in large graphs. In: The Eleventh International Conference on Learning Representations (2023)
34. Xiong R, Yang Y, He D, Zheng K, Zheng S, Xing C, Zhang H, Lan Y, Wang L, Liu T. On layer normalization in the transformer architecture. In: International Conference on Machine Learning, pp. 10524–10533. PMLR (2020)
35. Velickovic P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: International Conference on Learning Representations (2018)
36. Gilpin LH, Bau D, Yuan BZ, Bajwa A, Specter M, Kagal L. Explaining explanations: An overview of interpretability of machine learning. In: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), pp. 80–89. IEEE (2018)
37. Loshchilov I, Hutter F. Decoupled weight decay regularization. In: International Conference on Learning Representations (2018)
38. You R, Yao S, Xiong Y, Huang X, Sun F, Mamitsuka H, Zhu S. Netgo: improving large-scale protein function prediction with massive network information. Nucl Acids Res. 2019;47(W1):379–87.
39. You R, Zhang Z, Xiong Y, Sun F, Mamitsuka H, Zhu S. Golabeler: improving sequence-based large-scale protein function prediction by learning to rank. Bioinformatics. 2018;34(14):2465–73.
40. Kulmanov M, Hoehndorf R. Deepgoplus: improved protein function prediction from sequence. Bioinformatics. 2020;36(2):422–9.

41. Li W, Wang B, Dai J, Kou Y, Chen X, Pan Y, Hu S, Xu ZZ. Partial order relation-based gene ontology embedding improves protein function prediction. Brief Bioinform. 2024;25(2):077.
42. Defays D. An efficient algorithm for a complete link method. Comput J. 1977;20(4):364–6.
43. Faisal M, Zamzami E, et al. Comparative analysis of inter-centroid k-means performance using Euclidean distance, Canberra distance and Manhattan distance. J Phys: Conf Ser. 2020;1566: 012112.

## Publisher's Note