SCIENCE CHINA
Information Sciences

• Supplementary File •

A multimodal learning framework for protein function prediction based on multi-grained joint embedding

Jiangyi SHAO[1,2], Ziwen WANG[1], Zixu CHEN[1,2] , Bin LIU[1,2*]

[1]*School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China*
[2]*Zhongguancun Academy, Beijing 100094, China*

Appendix A Methods

We propose a multimodal protein function prediction framework integrating sequence representation, multi-sequence alignment (MSA) embedding, geometric graph modeling, and Gene Ontology (GO) semantic priors. The model consists of: (1) sequence embeddings extracted by a pretrained S-PLM model; (2) multi-level geometric graph encoding using E(n)-Equivariant Graph Neural Networks (EGNN); (3) pooled MSA representations; and (4) a cross-modal attention mechanism for protein–GO interaction modeling. The final prediction is generated using a multilayer fusion module.

Appendix A.1 Sequence Representation

Given an amino acid sequence:

$$s = (a_1, a_2, \ldots, a_L), \tag{A1}$$

where L denotes the sequence length and a_i the i-th residue, the pretrained Sequence Representation Model (S-PLM) yields [1]:

$$(\mathbf{h}^{\text{seq}}, \mathbf{H}^{\text{res}}, m), \tag{A2}$$

where $\mathbf{h}^{\text{seq}} \in \mathbb{R}^{1280}$ is the global protein embedding, $\mathbf{H}^{\text{res}} = (\mathbf{h}_1, \ldots, \mathbf{h}_L)$ with $\mathbf{h}_i \in \mathbb{R}^{1280}$ being the residue-level embedding, and m is a residue mask used to remove special tokens. Thus the residue embedding set is formalized as:

$$\mathbf{H}^{\text{res}} = (\mathbf{h}_1, \ldots, \mathbf{h}_L). \tag{A3}$$

Appendix A.2 MSA Representation

Given the MSA embedding matrix:

$$\mathbf{M} = (\mathbf{m}_1, \ldots, \mathbf{m}_N), \qquad \mathbf{m}_i \in \mathbb{R}^{768}, \tag{A4}$$

where N is the number of aligned sequences, we compute the pooled MSA feature via mean pooling [2]:

$$\mathbf{h}^{\text{msa}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{m}_i. \tag{A5}$$

A linear transformation is then applied:

$$\mathbf{h}^{\text{msa}} = \mathbf{W}_{\text{msa}} \mathbf{h}^{\text{msa}} + \mathbf{b}_{\text{msa}}, \tag{A6}$$

where $\mathbf{W}_{\text{msa}}$ and $\mathbf{b}_{\text{msa}}$ are trainable projection parameters.

Appendix A.3 EGNN-based Graph Structural Encoding

Each residue is treated as a graph node with coordinates $\mathbf{x}_i \in \mathbb{R}^3$ and feature $\mathbf{h}_i$. Edges are constructed using spatial proximity or inter-chain combinations [3].

(1)Edge Message Construction

For an edge (i, j), the EGNN message is computed as:

$$\mathbf{m}_{ij} = \phi_e(d_{ij}, \mathbf{h}_i, \mathbf{h}_j, \mathbf{e}_{ij}), \tag{A7}$$

* Corresponding author (email: bliu@bliulab.net)

where $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$ is the squared Euclidean distance, $\mathbf{e}_{ij}$ denotes edge attributes (set to None in our case), and $\phi_e(\cdot)$ is a multilayer perceptron (MLP).

(2)Node Feature Update

Node states are updated by aggregating neighboring messages:

$$\mathbf{h}_i' = \phi_h\left(\mathbf{h}_i, \sum_{j\in\mathcal{N}(i)} \mathbf{m}_{ij}\right), \tag{A8}$$

where $\phi_h(\cdot)$ is another MLP and $\mathcal{N}(i)$ denotes the neighbor set.

(3)Coordinate Update

EGNN updates coordinates as:

$$\mathbf{x}_i' = \mathbf{x}_i + C\sum_{j\in\mathcal{N}(i)} (\mathbf{x}_i - \mathbf{x}_j)\,\phi_x(\mathbf{m}_{ij}), \tag{A9}$$

where $\phi_x(\cdot)$ is a scalar-valued MLP and C is a normalization constant preventing numerical instability.

(4)Multi-level EGNN Stacks

Four EGNN modules operate at different semantic levels: residue-level, compressed residue-level, pooled sequence-level graph, and deep residue-level representation. For the k-th EGNN block, we obtain:

$$\mathbf{g}_k = \mathrm{Pool}\Big(\mathbf{H}^{(k)}\Big), \qquad k = 1, \ldots, 4, \tag{A10}$$

and the structural representation is

$$\mathbf{g} = [\mathbf{g}_1,\, \mathbf{g}_2,\, \mathbf{g}_3,\, \mathbf{g}_4]. \tag{A11}$$

Appendix A.4 Function Embedding Representation

Let the GO vocabulary be [4]:

$$\mathcal{T} = \{t_1, \ldots, t_K\}. \tag{A12}$$

A pretrained GO embedding model generates:

$$\mathbf{E}^{\mathrm{GO}} = \{\mathbf{e}_1, \ldots, \mathbf{e}_K\}, \qquad \mathbf{e}_k \in \mathbb{R}^{128}, \tag{A13}$$

where $\mathbf{e}_k$ is the semantic embedding of the k-th GO term.

Appendix A.5 Cross-Modal Attention for Protein-Function Interaction

Given a protein embedding $\mathbf{p} \in \mathbb{R}^{protein_dim}$ and a GO embedding $\mathbf{g} \in \mathbb{R}^{128}$, we first apply linear projections:

$$\mathbf{q} = W_p\mathbf{p}, \quad \mathbf{k} = W_g\mathbf{g}, \quad \mathbf{v} = W_g\mathbf{g}, \tag{A14}$$

where W_p and W_g are projection matrices. The multi-head attention mechanism computes:

$$\mathrm{Attention}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = \mathrm{softmax}\left(\frac{\mathbf{q}\mathbf{k}^\top}{\sqrt{d}}\right)\mathbf{v}, \tag{A15}$$

where d denotes the hidden dimensionality. The fused cross-modal embedding is:

$$\mathbf{f} = \mathbf{q} + \mathrm{Attention}(\mathbf{q}, \mathbf{k}, \mathbf{v}). \tag{A16}$$

Appendix A.6 Fusion and Prediction

We aggregate structural, semantic, and evolutionary representations:

$$\mathbf{z} = [\mathbf{g},\ \mathbf{f},\ \mathbf{h}^{\mathrm{msa}}], \tag{A17}$$

and feed $\mathbf{z}$ into two fully-connected layers:

$$\mathbf{z}' = \sigma(W_2\,\mathrm{BN}(W_1\mathbf{z})), \tag{A18}$$

where $\sigma(\cdot)$ is ReLU and BN denotes batch normalization. The final prediction is:

$$\hat{\mathbf{y}} = \mathrm{Sigmoid}(W_3\mathbf{z}'). \tag{A19}$$

Appendix A.7 Training Objective

As the task is multi-label classification, we use binary cross-entropy:

$$\mathcal{L} = -\sum_{k=1}^{K}\Big(y_k \log \hat{y}_k + (1 - y_k)\log(1 - \hat{y}_k)\Big), \tag{A20}$$

where $y_k \in \{0, 1\}$ is the ground truth label and $\hat{y}_k \in [0, 1]$ the predicted probability.

Appendix A.8 Model Building

The term multi-grained in this work is defined by the biological abstraction scope of each representation level, following a principled hierarchy in molecular biology: Residue → Protein → Protein Family. Specifically: i) Fine-grained representations operate at the residue level within a single protein, encoding local physicochemical and structural properties (sequence embeddings and EGNN graph features); ii) Medium-grained representations operate at the protein level, capturing functional semantics of entire proteins through GO hierarchical embeddings; these aggregate information across residues but remain specific to individual protein instances; iii) Coarse-grained representations operate at the protein family level, modeling evolutionary conservation and co-variation patterns across homologous sequences via MSA embeddings.

At the fine-grained level, the framework focuses on sequence–structure representation. A structure-aware protein language model (S-PLM) [1] based on ESM-2 is employed to extract sequence embeddings while incorporating structural cues through additional Transformer modules. AlphaFold2-predicted [5] contact maps are processed by a Swin-Transformer to encode three-dimensional topology and inter-residue spatial dependencies. These representations are further refined through an equivariant graph neural network (EGNN) [3], which treats residues as nodes and spatial proximities as edges, learning both residue-level and global protein-level representations that capture complex geometric relationships.

At the medium-grained level, functional knowledge from the Gene Ontology is incorporated using the Anc2Vec model [4]. Importantly, the GO embeddings used in MJEGO are not retrieved from homologous proteins via sequence search tools (e.g., BLAST or DIAMOND). Instead, Anc2Vec is pretrained solely on the Gene Ontology graph structure, learning vector representations that encode hierarchical semantic relationships such as ancestor–descendant and sibling dependencies across the three GO sub-ontologies (Biological Process, Molecular Function, and Cellular Component). These embeddings are sample-independent: they capture the structural semantics of GO terms themselves, without incorporating any protein-specific ground-truth annotations. Therefore, no label information leakage occurs during training or inference. By sampling annotated GO terms and aggregating their hierarchical embeddings, the framework generates medium-grained representations that bridge the fine-grained structural–sequence features and coarse-grained evolutionary context.

At the coarse-grained level, evolutionary information is introduced through the MSA Transformer [2], which models homologous sequences aligned by residue position. The network alternates between attention across sequences and along residue positions to learn both co-evolutionary dependencies and long-range contextual information. Trained under a masked language modeling objective, this module produces coarse-grained embeddings that capture evolutionary conservation and variability across protein families.

Finally, the outputs from all three levels are fused through a hierarchical strategy. The fine-grained EGNN integrates sequence and structure features into a geometric representation (Eq. A11). At the medium-grained level, cross-modal attention aligns this representation with GO embeddings, enabling functional ontology to actively modulate protein features (Eqs. A14-A16). At the coarse-grained level, MSA features are concatenated with the cross-modal output, and the fused vector z=[g,f,h_{msa}]is processed by a two-layer MLP for GO term prediction (Eqs. A17-A19).

Appendix B Evaluation metrics

To comprehensively evaluate the predictive performance of our model, we employed two widely used evaluation metrics: Fmax and the Area Under the Precision-Recall Curve (AUPR) [6].

Appendix B.1 Fmax Metric

The Fmax metric represents the maximum F-measure obtained across all possible prediction thresholds. Specifically, for each threshold t, the F-measure is defined as the harmonic mean of precision and recall:

$$F_t = \frac{2 \times \text{precision}(t) \times \text{recall}(t)}{\text{precision}(t) + \text{recall}(t)}, \tag{B1}$$

where precision and recall are defined as:

$$\text{precision}(t) = \frac{tp(t)}{tp(t) + fp(t)}, \tag{B2}$$

$$\text{recall}(t) = \frac{tp(t)}{tp(t) + fn(t)}. \tag{B3}$$

Then, Fmax is defined as:

$$\text{Fmax} = \max_t F(t), \tag{B4}$$

where tp, fp, and fn denote the numbers of true positives, false positives, and false negatives, respectively.

Appendix B.2 AUPR Metric

The AUPR metric quantifies the area under the precision-recall curve, which reflects the trade-off between precision and recall across different thresholds. It is computed using the trapezoidal rule as follows:

$$\mathrm{AUPR} = \sum_{i=1}^{n-1} (r_{i+1} - r_i) \times \frac{p_{i+1} + p_i}{2}, \tag{B5}$$

where p_i and r_i denote the precision and recall at the i-th threshold, respectively. The AUPR provides an overall assessment of model performance across varying decision boundaries and is particularly informative for imbalanced classification tasks.

Appendix C Results

Table C1 Performance comparison

	Fmax			AUPR		
Method	BP	CC	MF	BP	CC	MF
TransFun [5]	0.4451	0.6672	0.5971	0.4005	0.6661	0.5860
GVP [**?**]	0.3468	0.5699	0.3027	0.2492	0.4930	0.1948
MSA Transformer [2]	0.4372	0.6537	0.5084	0.3748	0.6331	0.4752
Ours	**0.4508**	**0.6861**	**0.6172**	**0.4024**	**0.6869**	**0.6070**

As shown in Table C1, our method achieves the best performance across all three GO sub-ontologies, Biological Process (BP), Cellular Component (CC), and Molecular Function (MF), in both Fmax and AUPR. For BP terms, the proposed method increases Fmax from 0.4451 to 0.4508 and AUPR from 0.4005 to 0.4024, reflecting the benefits of incorporating additional functional and evolutionary signals into a unified representation. The performance gains are more substantial on CC and MF, two ontologies where localized biochemical properties and structural context play a crucial role. On CC terms, Fmax improves from 0.6672 to 0.6861 and AUPR from 0.6661 to 0.6869. On MF terms, Fmax increases from 0.5971 to 0.6172 and AUPR from 0.5860 to 0.6070. These improvements highlight the effectiveness of combining fine-grained structure features with medium- and coarse-grained functional and evolutionary features.

Compared with GVP, which uses only structural information, and MSA Transformer, which relies solely on MSA-based evolutionary patterns, our framework shows markedly superior performance across all metrics. This demonstrates that single-source models, even those using strong encoders such as GVP or MSA Transformer, are limited by the incompleteness of any individual modality. By contrast, the proposed multi-grained joint embedding integrates complementary biological evidence and produces richer, more discriminative protein representations.

Overall, the results confirm that combining sequence, structure, functional, and evolutionary information within a unified multi-scale learning framework significantly enhances protein function prediction, providing robust and generalizable improvements across all three GO sub-ontologies.

Table C2 Ablation study on Fine-grained representation

		Fmax			AUPR		
Method	Representation	BP	CC	MF	BP	CC	MF
TransFun	ESM-1b	0.4451	0.6672	0.5971	0.4005	0.6661	0.5860
Ours	ESM-1b	0.4602	0.6858	0.6197	0.3810	0.6831	0.6076
TransFun	ESM-IF	0.4149	0.4666	0.3409	0.3410	0.2829	0.2284
Ours	ESM-IF	0.4342	0.6673	0.5948	0.3710	0.6491	0.5696
TransFun	ESM-1v	0.4413	0.6650	0.5924	0.3950	0.6576	0.5838
Ours	ESM-1v	0.4510	0.6817	0.6139	0.3981	0.6778	0.6056
TransFun	ESM2	0.3372	0.6715	0.6016	0.2309	0.6564	0.5934
Ours	ESM2	0.4545	**0.6883**	0.6164	0.3889	0.6789	**0.6082**
TransFun	Swin	**0.4620**	0.6660	0.5999	**0.4160**	0.6612	0.5925
Ours	Swin	0.4508	0.6861	**0.6172**	0.4024	**0.6869**	0.6070

As shown in Table C2, we investigated the impact of different information sources on function prediction, examining the experimental performance using ESM-1b, ESM-IF, ESM-1v, ESM2, and Swin-Transformer as information sources. The experimental results indicate that embeddings generated by Swin-Transformer achieved superior performance. For the

Fmax metric, Swin-Transformer attained the best performance across BP and MF. For the AUPR metric, it achieved the best performance in BP and CC tasks, while ESM2 performed best in MF. These results strongly validate that using Swin-Transformer to generate embeddings in our model is highly effective, yielding better performance compared to embedding generation methods based on the ESM series.

The models based on ESM2 embeddings also exhibit strong MF performance, indicating that pre-trained protein language models retain valuable contextual information from sequence data. Nevertheless, the superiority of Swin-based encoders highlights the importance of integrating explicit spatial representations into the fine-grained layer. These findings demonstrate that the proposed structure-aware fine-grained representation effectively captures complementary information from both sequential and geometric dimensions, serving as a solid foundation for higher-level multimodal fusion.

As shown in Table C3, we investigated the effects of incorporating different information sources, explored methods of information fusion, and analyzed the impact of adding various information sources. From Table C3, it can be seen that simultaneously incorporating both GO and MSA, with GO added first followed by MSA, achieves better performance. In terms of the Fmax metric, after adding GO first and then MSA, the model utilizes four types of information sources: structural information, sequence information, GO information, and MSA information. The best performance was achieved in the CC and MF tasks, while the method using only the GO information source achieved the best performance in the BP task. The results for the AUPR metric are similar to those for the Fmax metric. This also demonstrates the effectiveness of our model; through the combined contribution of multiple information sources, we have obtained a more comprehensive representation of proteins.

Table C3 Ablation study on multi-grained joint embedding

	Fmax			AUPR		
Method	BP	CC	MF	BP	CC	MF
TransFun	0.4451	0.6672	0.5971	0.4005	0.6661	0.5860
Only GO	**0.4681**	0.6766	0.6136	**0.4216**	0.6751	0.5993
Only MSA	0.4517	0.6741	0.6020	0.4019	0.6756	0.5899
MSA first, then GO	0.3860	0.6804	0.6138	0.3237	0.6801	0.6018
GO first, then MSA	0.4602	**0.6858**	**0.6197**	0.3810	**0.6831**	**0.6076**

The performance improvement on BP is comparatively marginal relative to CC and MF, consistent with the recognized difficulty of BP prediction. BP annotations describe high-level cellular pathways involving complex multi-protein interactions and context-dependent regulation, which are not fully captured by sequence, structure, or evolutionary features alone. CC and MF annotations, by contrast, more directly reflect localized biochemical properties amenable to these representations. Despite modest absolute gains, BP improvements are consistent across both metrics and all ablation configurations (Tables C1 and C3), suggesting genuinely complementary information rather than random variation. Incorporating protein–protein interaction networks may further improve BP prediction.

References

1 Wang D, Pourmirzaei M, Abbas U L, et al. S-plm: structure-aware protein language model via contrastive learning between sequence and structure. Advanced Science, 2025, 12: 2404212
2 Rao R M, Liu J, Verkuil R, et al. Msa transformer. In: Proceedings of International conference on machine learning. PMLR, 2021. 8844–8856
3 Satorras V G, Hoogeboom E, Welling M. E (n) equivariant graph neural networks. In: Proceedings of International conference on machine learning. PMLR, 2021. 9323–9332
4 Edera A A, Milone D H, Stegmayer G. Anc2vec: embedding gene ontology terms by preserving ancestors relationships. Briefings in Bioinformatics, 2022, 23: bbac003
5 Jumper J, Evans R, Pritzel A, et al. Highly accurate protein structure prediction with alphafold. nature, 2021, 596: 583–589
6 Boadu F, Cao H, Cheng J. Combining protein sequences and structures with transformers and equivariant graph neural networks to predict protein function. Bioinformatics, 2023, 39: i318–i325