Large Language and Protein Assistant for Protein-Protein Interactions Prediction

Anonymous ACL submission

Abstract

Predicting the types and affinities of protein-protein interactions (PPIs) is crucial for understanding biological processes and developing novel therapeutic approaches. While encoding proteins themselves is essential, PPI networks can also provide rich prior knowledge for these predictive tasks. However, existing methods oversimplify the problem of PPI prediction in a semi-supervised manner when utilizing PPI networks, limiting their practical application. Furthermore, how to effectively use the rich prior knowledge of PPI networks for novel proteins not present in the network remains an unexplored issue. Additionally, due to inflexible architectures, existing methods cannot handle complexes containing an flexible number of proteins. To overcome these limitations, we introduce LLaPA (Large Language and Protein Assistant), a multimodal large language model that integrates proteins and PPI networks. LLaPA offers a more rational approach to utilizing PPI networks for PPI prediction and can fully exploit the information of PPI networks for unseen proteins. Through natural language instructions, LLaPA can accept flexible number of protein sequences and has the potential to perform various protein tasks. Experiments show that LLaPA achieves state-of-the-art performance in multi-label PPI (mPPI) type prediction and is capable of predicting the binding affinity between multiple interacting proteins based on sequence data.

1 Introduction

Protein-protein interactions (PPIs) are fundamental to biological processes and critical in drug discovery (Wells and McClendon, 2007; Braun and Gingras, 2012). Traditional high-throughput screening methods, such as yeast two-hybrid screens (Ito et al., 2001) and tandem affinity purification (Gavin et al., 2002), are both expensive and time-consuming. Recently, advancements in deep learning have led to numerous approaches for predicting PPIs. These approaches can be divided into those that utilize PPI networks and those that do not. Methods that do not use PPI networks include DPPI (Hashemifar et al., 2018), DNN-PPI (Li et al., 2018), PIPR (Chen et al., 2019), TAGPPI (Song et al., 2022), and Geo-PPI (Liu et al., 2021). These methods encode proteins individually and then concatenate the features of paired proteins for downstream tasks.

Methods based on PPI networks encode not only proteins but also the PPI network. In a PPI network, nodes represent proteins, and edges, often multi-labeled, indicate relationships between them. PPI networks are essential for predicting PPIs, as protein interactions depend on both individual features and their positions within the larger network (Lee, 2023). GNN-PPI (Lv et al., 2021) was the pioneering method leveraging PPI networks, achieving significant improvements in the mPPI task. Subsequent methods, such as SemiGNN-PPI (Zhao et al., 2023), HIGH-PPI (Gao et al., 2023b), and MAPE-PPI (Wu et al., 2024b), built upon GNN-PPI's settings and demonstrated even better performance in mPPI prediction.

Despite significant advancements, these methods face three critical limitations: **(1) Oversimplified mPPI Task Setting:** Existing methods utilize connection information between unseen proteins in a semi-supervised manner (Kipf and Welling, 2016; Lv et al., 2021; Gao et al., 2023b; Zhao et al., 2023; Wu et al., 2024b), which oversimplifies task difficulty. Current benchmarks separate a portion of the PPI network data as the test set, and the topological information of the test set is also input into the model. This approach explicitly informs the model of relationships between protein pairs being tested, facilitating information exchange and simplifying PPI prediction. Unlike readily available protein sequences, acquiring connection information between proteins often requires extensive biological experiments and analysis, making this

approach impractical for real-world applications. **(2) Ineffectiveness of PPI Network Information for Unseen Proteins:** In real-world scenarios, we frequently encounter unseen proteins that do not exist in any PPI network. Existing methods fail to effectively utilize PPI network information in such cases, as the model cannot extract useful information from the network topology, thereby affecting prediction accuracy and practicality. **(3) Limitations in Multi-Protein Interactions:** These models, with their fixed architectures, can only handle interactions between two proteins and cannot predict relationships involving multiple proteins or the affinity of multi-protein complexes. Many biological processes depend on multi-protein complexes, such as antigen-antibody complexes, which typically consist of three chains: the antigen, the heavy chain of the antibody, and the light chain of the antibody (Wu et al., 2024a). The challenge lies in the unknown number of proteins, requiring models to be flexible enough to accept an arbitrary number of proteins as input. These methods struggle with such complex multi-protein interactions, limiting their applicability in practical biological research.

Recently, some studies have achieved notable performance in protein encoding and understanding through joint learning of proteins and natural language, such as ProtLLM (Zhuo et al., 2024), ProLLama (Lv et al., 2024), Prot2Text (Abdine et al., 2024), and ProteinGPT (Xiao et al., 2024). Pre-trained on large-scale protein databases, these methods exhibit strong generalization capabilities. The flexibility of LLMs enables them to handle tasks involving multiple protein sequences, addressing Challenge (3) effectively. Nonetheless, they did not further explore the task of multi-sequence proteins, nor did they utilize the rich information provided by the PPI network.

In this work, we propose a multimodal model called LLaPA (Large Language and Protein Assistant), which effectively addresses the aforementioned three challenges simultaneously. LLaPA integrates protein representations and PPI networks into a large language model (LLM). We construct a more general PPI network, inputting both network topology information and protein information into the LLM to assist in decision-making. During both training and inference, we completely remove edges that overlap between the PPI network and the test set. Treating the PPI network as external knowledge, we inject this knowledge into the LLM prompt using Retrieval-Augmented Generation (RAG) (Gao et al., 2023a). For proteins not present in the PPI network, we find similar protein nodes within the PPI network and provide their topology as additional information. Leveraging the flexibility of large language models, LLaPA can accept flexible number of proteins as input and use natural language instructions for downstream tasks.

The contributions of this paper can be summarized as follows:

- We reveal the limitations of existing methods in utilizing PPI networks and provide a straightforward method for more reasonable utilization of PPI networks.
- We propose treating the PPI network as external knowledge and injecting it into LLMs through RAG to assist downstream tasks. This approach is also effective for unseen proteins. We also constructed a more general PPI network called UPPIN.
- We develop a protein natural multimodal large language model, LLaPA, which integrates the protein encoder EMS-2 (Lin et al., 2022), the PPI network encoder SGC (Wu et al., 2019), and the large language model llama3-8b (Touvron et al., 2023). LLaPA can handle flexible numbers of proteins and has the potential perform diverse protein tasks.
- Experiments show that LLaPA achieves state-of-the-art (SOTA) performance on the mPPI task and demonstrates significant accuracy in multi-sequence affinity prediction.

2 Related work

2.1 Protein-protein interactions

Protein-Protein Interactions (PPIs) are crucial components of cellular activities and play significant roles in various biological functions (Lu et al., 2020; Bryant et al., 2022; Richards et al., 2021). The interactions among multiple proteins form complex PPI networks, which implicitly represent the signaling processes and pathways of various life activities within organisms. Understanding PPIs not only helps us decipher complex biological systems but also aids in identifying potential targets for disease intervention.

With the rise of deep learning technologies, researchers have proposed numerous deep learning-based methods for PPI prediction. From a task

perspective, PPI tasks include: (1) Binary Classification: This task involves inputting a pair of protein sequences and determining whether these two proteins can interact. Methods such as DPPI, DNN-PPI, PIPR, and TAGPPI typically include a convolutional neural network module as the protein encoder. After encoding the two proteins separately, a feature fusion module combines the encoded features, and a binary classifier outputs the classification result. (2) Multi-label PPI Type Prediction: This task focuses on identifying the types of interactions between two proteins. PIPR and TAGPPI can also handle this task. GNN-PPI introduces the topological information of the PPI network, combining the topological information of proteins in the PPI network with protein features, achieving significant improvements in the mPPI task. Subsequent works like HIGH-PPI and MAPE-PPI use the same PPI network. (3) Protein-Protein Binding Affinity Prediction: This task typically focuses on predicting changes in binding affinity between protein complexes due to mutations, as seen in works like Geo-PPI, DDAffinity (Yu et al., 2024), and top-Nettree (Wang et al., 2020). These methods input the original and mutated protein features to predict the affinity changes caused by specific mutations. Few works directly predict the binding affinity of protein complexes, with PIPR being one known example. These methods can only handle pairwise protein interactions and cannot predict the affinity of multi-sequence complexes. (4) PPI Binding Site Prediction: This task requires amino acid-level encoding. Representative works include DeepHomo (Yan and Huang, 2021), GLINTER (Xie and Xu, 2022), and DeepInter (Lin et al., 2023), which are beyond the scope of this discussion. (5) Protein-Protein Conformation Prediction: Similar to task (4), this also requires amino acid-level encoding and is not covered in this paper.

2.2 Multimodal Large Language Models

Multimodal Large Language Models (MLLMs) are dedicated to enabling LLMs to recognize and understand non-natural language modality data, such as images, sounds, etc. A common approach involves first using multimodal encoders to encode data from various modalities. Then, a projector module aligns the output space of these modalities with the input space of the LLM. This process injects multimodal output features into the LLM, enabling it to understand non-natural language modalities. Subsequently, Multimodal Instruction Tuning is employed, which mixes natural language instructions with multimodal data, allowing the MLLM to perform downstream tasks based on the given multimodal data and natural language instructions. Representative works include LLaVA (Liu et al., 2024a), InstructBLIP (Dai et al., 2023), VisionLLM (Wang et al., 2024), and MultiModal-GPT (Gong et al., 2023). Recently, some efforts have been made to integrate protein modality into LLMs, endowing LLMs with the ability to understand proteins. Relevant work includes ProtLLM (Zhuo et al., 2024), Prot2Text (Abdine et al., 2024), and ProteinGPT (Xiao et al., 2024), which have achieved noteworthy performance in gene ontology term prediction, as well as understanding of protein sequences and structures.

3 Method

3.1 Problem Settings

This work focuses on two tasks: (1) Multi-label PPI type prediction. Given a pair of proteins (p_1, p_2), the goal is to predict the types of interactions between them, which is a multi-class classification task. (2) Multi-sequence Affinity prediction. Given a complex $C = (B, T)$, where B refers to the binder and T refers to the target, predict its logarithmic dissociation constant $\log Kd = \log \frac{[B][T]}{[BT]}$. B and T can each be a single protein sequence or a complex containing multiple sequences. For the PDB2020 (PP) dataset from PDBBind (Liu et al., 2017), which includes 2852 complexes, accurately extracting the binder and target based on the given information is very challenging and requires manual analysis of the papers corresponding to each PDB entry. Therefore, we have simplified this task in the form of: Given a set of proteins $(p_1, p_2, ..., p_k)$, predict its $\log Kd$ specified by the dataset. We leverage a PPI network to obtain prior knowledge about the target proteins to aid in the prediction. The PPI network is represented as a graph $G = \{V, \mathbf{A}\}$, where $V = \{v_1, v_2, .., v_n\}$ are the nodes of the graph, with each node v_i corresponding to a protein. $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the adjacency matrix of the graph, where $a_{ij} = 1$ if there is an interaction between proteins p_i and p_j, and $a_{ij} = 0$ otherwise. We use $\mathbf{X} \in \mathbb{R}^{n \times d}$ to represent the feature matrix of the graph G, where each row x_i represents the features of the i-th protein.

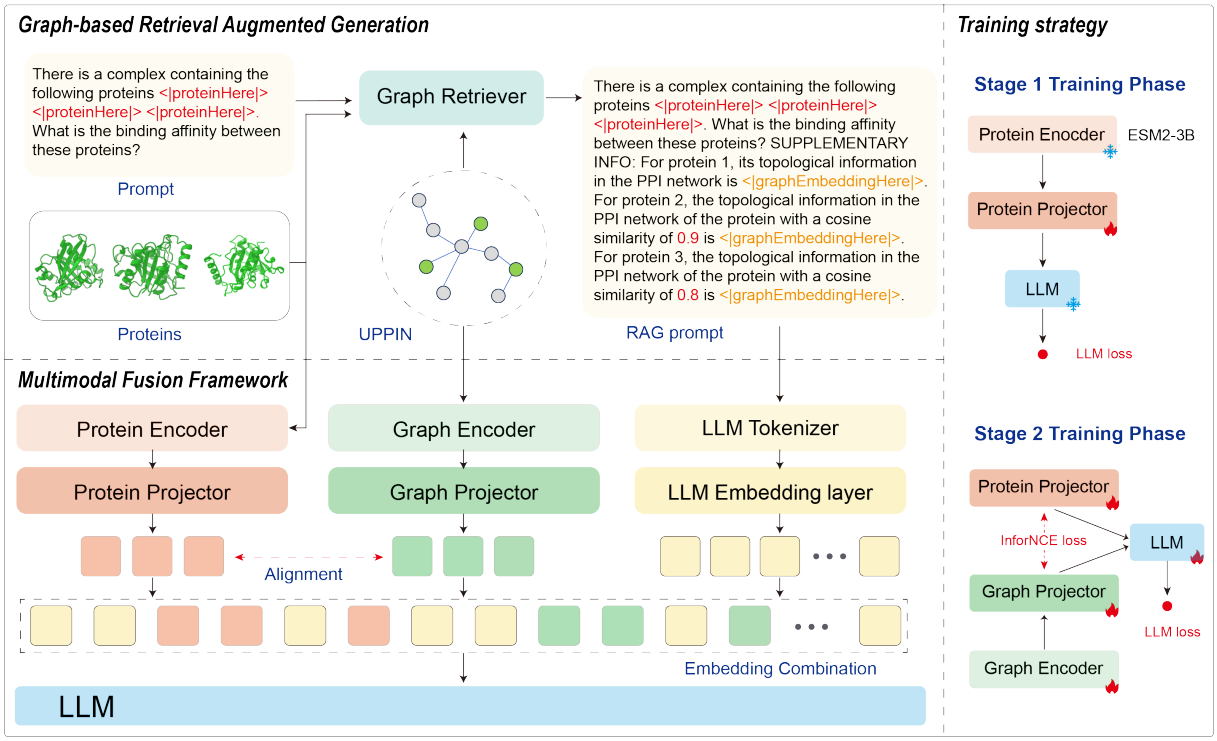

Figure 1: An overview of LLaPA. It consists of two main components: **Graph-based Retrieval Augmented Prompt Preparation** and a **Multimodal Fusion Framework**. In the Graph-based Retrieval Augmented Prompt Preparation stage, we search for matching topological information in UPPIN (a unified PPI network we constructed) based on the input protein and task instructions. This process yields additional topological features and information-enhanced natural language instructions. Subsequently, we input the protein, graph features, and enhanced instructions into our Multimodal Fusion Framework for training and inference.

3.2 Overall Architecture

LLaPA is an integrated large language model that combines protein and graph data, as shown in Figure 1. It consists of two main components: **Graph-based Retrieval Augmented Prompt Preparation** and a **Multimodal Fusion Framework**. In the Graph-based Retrieval Augmented Prompt Preparation stage, we search for matching topological information in the PPI network based on the input proteins. This allows us to obtain additional topological features and enriched natural language instructions. Subsequently, we input the proteins, graph features, and the augmented instructions into our Multimodal Fusion Framework for training and inference. Before formally introducing our method, we will briefly discuss the limitations of existing PPI network-based methods.

3.3 Limitations of Oversimplified mPPI Task Setting

PPI networks are essential for PPI-related tasks, as a protein's position within the network provides valuable prior knowledge. (Lv et al., 2021) were the first to apply PPI networks to the mPPI task with their GNN-PPI model, which uses a graph isomorphism network (GIN) (Xu et al., 2018) to encode PPI network topology. They validated the model using three data partitioning methods: random, DFS (depth-first search), and BFS (breadth-first search). As shown in Figure 2-(a), (b), and (c), these methods partition a portion of the edges as the test set, which the labels are not used for training but the structural information are retained for message passing in the GNN model. This allows message exchange between the protein pairs being tested, reducing prediction difficulty. During training, GNN-PPI uses the training edges (green solid edges in Figure 2), but during testing, the test edges (dashed edges in Figure 2) are also included within the graph encoder. Subsequent work, HIGH-PPI, followed GNN-PPI's setup, while MAPE-PPI used all topological information during both training and inference phases.

Unlike protein sequences, which are relatively easy to obtain, acquiring connection information between proteins requires extensive biological experiments and analysis. This makes it challenging to effectively use connection information between

test proteins in practical applications. However, removing test set edges during testing may bring another issue, as shown in Figure 2-(d), (e), and (f), where gray nodes become isolated and cannot obtain useful information from the PPI network. Additionally, an unseen protein inherently has no edges in the PPI network, making it an isolated node. Existing methods do not address how to handle this situation.

3.4 Graph-based Retrieval Augmented Prompt Preparation

To address the aforementioned issues, we propose a novel method that utilizes the PPI network as an external knowledge source. By employing the RAG (Retrieval-Augmented Generation) technique, we integrate the knowledge from the PPI network into the input of the large language model (LLM). Given a set of proteins $P = \{p_1, p_2, ..., p_m\}$ and a textual task prompt W, we first locate the corresponding nodes $V = \{v_1, v_2, ..., v_m\}$ in the PPI network for each protein. Next, we construct an enhanced natural language instruction W_{RAG}. Finally, we input this set of proteins, along with the graph node information and the enhanced instruction into the LLM.

In real-world applications, we may lack prior knowledge about a new protein, meaning it might not exist in any PPI network. In the biological domain, Multiple Sequence Alignment (MSA) (Edgar and Batzoglou, 2006) is a common method for analyzing protein functions. MSA aligns multiple protein sequences to study the structure, function, and evolutionary relationships of the target protein. A key aspect of MSA is the use of reference sequences. Inspired by MSA, for a protein that is an isolated node in the PPI network, we can approximate its topological reference by comparing it with proteins in the PPI network. Specifically, for a protein p not present in the PPI graph G, we calculate its similarity to each protein in G, denoted as $S = \{s(p, p_i)|p_i \in G\}$. This similarity can be computed using methods such as sequence alignment scores, structural similarity, or other biological metrics; here, we use the cosine similarity between protein features. With the similarity scores, we retrieve the most similar proteins in the PPI network and use their topological information as a proxy for the isolated protein. This enables us to construct an enriched instruction W_{RAG} that includes relevant topological features, which we then integrate into the LLM prompt for prediction tasks. We did not perform any additional processing on the similarity scores; instead, we directly placed the similarity scores explicitly into W_{RAG}.

Our instruction template, shown in Appendix A Figure 3, uses a structured data representation, separating text, protein sequences, and graph node indices. In the text, we use two special tokens, `<|proteinHere|>` and `<|graphEmbeddingHere|>`, to denote the protein embedding and the PPI network node embedding, respectively. This method can also address isolated nodes caused by the removal of test edges. When constructing W_{RAG}, we initially assess the degree of the target protein p within the PPI network. If the degree is 0, indicating that p is an isolated node, we employ the same methodology used for an unseen protein to derive the topological information of similar proteins in the PPI network.

3.4.1 Unified PPI network

Although leveraging protein similarity to utilize the topological information of similar proteins in the PPI network is beneficial, the number and diversity of protein nodes remain a limitation. Intuitively, a PPI network with more proteins and greater diversity enhances the model's generalization ability. To achieve better generalization, we constructed a larger PPI network called UPPIN (Unified PPI Network), which consists of three sub-datasets: STRING (Homo sapiens subset) (Szklarczyk et al., 2016), PDBBind (Liu et al., 2017), and SAbDab (Dunbar et al., 2014). UPPIN includes a total of 26,180 unique proteins and 594,216 unique edges. By constructing UPPIN, we expanded the coverage and diversity of the PPI network, enhancing the model's generalization capability. This larger-scale PPI network provides richer topological information and better supports prediction tasks for unseen proteins. Refer to Appendix A.2 for more details.

3.5 Multimodal Fusion Framework

As illustrated in Figure 1, we encode the protein, graph, and text separately, then fuse them to form the input for the LLM. For a given protein p, we use an encoder $f_p(\cdot)$ to obtain the protein features $\mathbf{Z_P} = f_p(p)$. Similar to LLaVA (Liu et al., 2024a), we use a learnable mapping matrix $\mathbf{W}_p$ to map $\mathbf{Z}_p$ to the embedding tokens $\mathbf{H}_p$ for the LLM:

$$\mathbf{H}_p = \mathbf{W}_p \cdot \mathbf{Z}_p, \;\; with \;\; \mathbf{Z}_p = f_p(p).$$

While sophisticated designs like QFormer (Li et al., 2023), C-Abstractor, and D-Abstractor (Cha

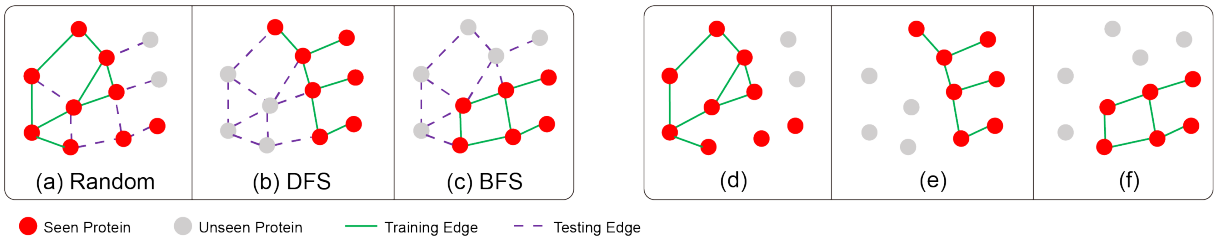

Figure 2: Existing data splitting methods for models based on PPI networks include: (a) Random: randomly selecting a portion of edges from the PPI network as the test set; (b) DFS: using depth-first approach to traverse the PPI network and selecting a portion of edges as the test set; (c) BFS: using breadth-first approach to traverse the PPI network and selecting a portion of edges as the test set. Regardless of the splitting method used, all edges are input into the GNN during the inference process. This allows information exchange between the proteins being tested, greatly simplifying the difficulty of multi-label PPI type prediction and limiting their practical value. We believe that all test edges should be removed during both the training and testing phases, as shown in (d)(e)(f).

et al., 2024) exist for connecting different data modalities with LLMs, recent research suggests that a Linear Projector may be optimal when sufficient computational resources are available (Yao et al., 2024). We chose ESM2-3B (Lin et al., 2022) as the protein encoder, a transformer-based model capable of directly encoding amino acid sequences.

For the PPI graph, we first use a graph encoder $f_v(\cdot)$ to encode it as $\mathbf{X}^{'} = f_v(\mathbf{X})$. We then extract the embedding of the node corresponding to protein p in the graph, denoted as $\mathbf{X}_p^{'}$. A mapping matrix $\mathbf{W}_v$ is used to map $\mathbf{X}_p^{'}$ to the embedding tokens $\mathbf{H}_v$ for the LLM:

$$\mathbf{H}_v = \mathbf{W}_v \cdot \mathbf{X}_p^{'}, \quad with \;\; \mathbf{X}^{'} = f_v(\mathbf{X}).$$

We use SGC (Wu et al., 2019) as the graph encoder, which balances encoding capability and computational efficiency. The graph is encoded using the following formula:

$$\mathbf{X}^{'} = (\hat{\mathbf{D}}^{\frac{-1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{\frac{-1}{2}})^K \mathbf{X} \mathbf{\Theta},$$

where $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ denotes the adjacency matrix with inserted self-loops, $\hat{\mathbf{D}}_{ii} = \sum_j \hat{\mathbf{A}}_{ij}$ is the diagonal degree matrix, and $\mathbf{\Theta} \in \mathbb{R}^{d \times d'}$ is the weight matrix. The parameter K controls the number of hops or the receptive field of the convolution.

For an input (P, V, W_{RAG}), we obtain the protein embeddings $(\mathbf{H}_{p_1}, \mathbf{H}_{p_2}, ..., \mathbf{H}_{p_m})$ and the corresponding graph node embeddings $(\mathbf{H}_{v_1}, \mathbf{H}_{v_2}, ..., \mathbf{H}_{v_m})$ using the methods described above. We then use the encoding layer of the LLM to obtain the corresponding text embeddings $(\mathbf{H}_{w_1}, \mathbf{H}_{w_2}, ..., \mathbf{H}_{w_n})$. Finally, we combine these embeddings into a complete input: $(\mathbf{H}_w, \mathbf{H}_p, \mathbf{H}_v)$, where the order of these embeddings depends on the positions of different types of tokens in W_{RAG}.

4 Model Training

Our training process is divided into two steps. In the first step, we remove the graph encoder and freeze both the LLM and the protein encoder, training only the protein projector $\mathbf{W}_p$ to map the protein features into the LLM's input space. For this step, we use the UniProtQA (Luo et al., 2023) dataset, which contains 569,516 proteins and 1,891,506 protein question-answer pairs. Each QA record includes only one protein sequence, with questions covering protein functions, official names, families, and sub-cellular locations. Given a QA record (Q, A, p), where Q represents the question, A represents the response, and p is a protein sequence, the objective is to maximize the probability $\mathcal{P}(A|Q, p)$. We optimize this probability using the LLM's autoregressive objective:

$$L_{LLM} = -\sum_{t=1}^{T} \log \mathcal{P}(w_t|w_1, ..., w_{t-1}; p)$$

In the second step, we fine-tune directly on the downstream task. Here, we freeze the protein encoder and load the graph encoder. We update the graph encoder $f_v(\cdot)$, the graph mapping matrix $\mathbf{W}_v$, the protein mapping matrix $\mathbf{W}_p$, and the weights of the LLM. Unlike the protein encoder and protein projector, which have been pre-trained, the weights of the graph encoder and graph projector are randomly initialized. Since the topological information provided by the graph complements the corresponding protein information, and the protein feature projector already connects protein features to the LLM, we can leverage this complementarity to accelerate the alignment of graph features with the LLM. We use InfoNCE (Oord et al., 2018) to maximize the mutual information between

the protein representation and the corresponding topological information. The objective function is: $L_{infoNCE} = -\mathbb{E}[log\frac{exp(E(\mathbf{H}_p,\mathbf{H}_v))}{\sum_{v'} exp(E(\mathbf{H}_p,\mathbf{H}_{v'}))} + log\frac{exp(E(\mathbf{H}_p,\mathbf{H}_v))}{\sum_{p'} exp(E(\mathbf{H}_{p'},\mathbf{H}_v))}]$, where $E(\cdot)$ is an energy function, which can be of flexible form. We use the dot product for its simplicity, i.e., $E(\mathbf{H}_p, \mathbf{H}_v) = \mathbf{H}_p \cdot \mathbf{H}_v$. The advantage of this approach is that we do not need to design additional tasks to pre-train the graph encoder and projector. Therefore, the loss function for the second stage of training is:

$$Loss = L_{infoNCE} + L_{LLM}.$$

5 Experiments

We evaluated LLaPA's capabilities on multi-label PPI type prediction (mPPI) and multi-sequence affinity prediction (MA).

Datasets. For the mPPI task, we used two subsets of STRING, SHS27k and SHS148k, constructed by (Chen et al., 2019). We used the same data splitting methods as GNN-PPI: random (randomly selecting test edges from the PPI network), DFS (depth-first search for test edges), and BFS (breadth-first search for test edges). We split these datasets into training, validation, and test sets in a 60%:20%:20% ratio. For the MA task, we used PDB2020, splitting it into training and test sets at an 80%:20% ratio.

Baselines. For the mPPI task, we compared our approach against DPPI, DNN-PPI, PIPR, ESM2-3B (fixed), ESM2-3B (ft), ProtLLM, GNN-PPI, HIGH-PPI, and MAPE-PPI. Since GNN-PPI, HIGH-PPI, and MAPE-PPI have access to the complete PPI network during the test phase, we removed the edges contained in the test set from the PPI network for a fairer comparison. These modified versions are denoted as GNN-PPI/R, HIGH-PPI/R, and MAPE-PPI/R, respectively. For ESM2-3B (fixed), we fixed the parameters of ESM2-3B and trained a multi-classifier on top of it. For ESM2-3B (ft), we fine-tuned all the weights of ESM2-3B in addition to training multi-classifiers, keeping all training hyperparameters consistent with LLaPA. For ProtLLM, we adapted the original code provided by the authors to support the mPPI task and fine-tuned it using the pre-trained weights and hyperparameters supplied by the authors. For the MA task, we used PIPR to predict the binding affinity of all two-protein complexes in the test set. Additionally, we trained three affinity prediction models using ESM2-3B, named E(2), E(3), and E(4). E(2) predicts the binding affinity of two-protein complexes, E(3) for three-protein complexes, and E(4) for four-protein complexes.

5.1 Multi-label PPI Type Prediction

The experimental results are shown in Table 1. From these results, we can draw two key insights:

(1) Underperformance of PPI network-Based Methods After Edge Removal. PPI network-based methods yield unsatisfactory results after removing the edges contained in the test set. These results are expected because graph encoders heavily rely on the graph structure, and removing test set edges significantly alters this structure, making it difficult for the learned weights to be effective. This issue is particularly pronounced with DFS and BFS data splitting methods, which can result in isolated nodes that cannot obtain useful information during the graph message-passing process.

(2) Superior Performance of LLaPA. LLaPA demonstrates superior performance across all task settings. Under the random splitting method, LLaPA achieves modest improvements over the second-best model on the SHS27k and SHS148k. However, the improvements are much more significant under the DFS and BFS methods, with LLaPA outperforming the second-best model by a substantial margin. The performance on the larger SHS148k dataset is better than on SHS27k, likely due to the increased dataset size facilitating model fitting. Despite having similar architectures, ProtLLM's lack of PPI network information led to suboptimal performance on the mPPI task.

5.2 Multi-sequence Affinity Prediction

The PDB2020 dataset contains complexes with a range of 2 to 16 unique proteins. To evaluate LLaPA's prediction capabilities, we categorized these complexes based on the number of unique proteins and assessed performance for each group. Most complexes contain fewer than 5 unique proteins, while those with 7, 9, 13, 14, and 16 unique proteins each have only one instance. The dataset was randomly divided into training and test sets with an 80:20 ratio, resulting in no test data for groups with 9, 13, 14, and 16 unique proteins, as shown in Appendix B.3. The experimental results is presented in Table 2. LLaPA achieved the best MAE and PCC performance within each group. We also present the training and prediction results for group 6 in the Appendix B.6.

	SHS27k			SHS148k		
	random	**dfs**	**bfs**	**random**	**dfs**	**bfs**
DPPI	70.45	43.69	43.87	76.10	51.43	50.80
DNN-PPI	75.18	48.90	51.59	85.44	56.70	54.56
PIPR	79.59	52.19	47.13	88.81	61.38	58.57
ESM2-3B (fixed)	47.58	42.50	41.97	48.92	43.06	41.25
ESM2-3B (ft)	79.23	63.38	48.80	87.86	66.92	61.88
ProtLLM	48.67	42.77	41.94	49.29	42.66	40.33
GNN-PPI/R	40.53	43.19	42.52	39.48	40.96	41.42
HIGH-PPI/R	41.51	40.06	39.87	42.81	51.06	45.94
MAPE-PPI/R	76.84	51.69	55.21	85.96	62.13	56.68
LLaPA	**82.49** (+2.90)	**69.54** (+6.16)	**67.21** (+12)	**91.78** (+2.97)	**73.93** (+7.01)	**70.90** (+9.02)

Table 1: Experimental results for multi-label PPI type prediction (micro-F1). Bold and underline are used to highlight the first and second scores respectively.

We also considered the scenario where all sequences in the complex were input into the model as shown in Appendix A.

Sequence Number	Methods	MAE (↓)	PCC (↑)
2	PIPR	1.42	0.34
	E(2)	1.43	-0.11
	LLaPA	**1.35**	**0.41**
3	E(3)	1.24	0.13
	LLaPA	**1.11**	**0.51**
4	E(4)	1.82	-0.24
	LLaPA	**1.09**	**0.76**
5	LLaPA	1.02	0.35
6	LLaPA	2.37	0.96
7	LLaPA	0.82	N/A
all	LLaPA	1.26	0.49

Table 2: Experimental results of MA prediction on PDB2020, measured by mean absolute error (MAE) and Pearson correlation coefficient (PCC).

5.3 Ablation Study

We conducted ablation experiments to evaluate three components: (1) the utility of the pre-trained protein projector, (2) the effectiveness of the constructed UPPIN network, and (3) the impact of the designed alignment loss function $L_{infoNCE}$. These experiments were performed on the SHS27k dataset, partitioned by DFS. As shown in Table 3, pretraining improved results by 7.34. Utilizing UPPIN improved results by 2.62 compared to the original PPI network of SHS27k, and by 26.23 compared to not using a PPI network. This is intuitive as UPPIN introduces more proteins and edges, enriching topological information, whereas not using a PPI network is akin to encoding proteins with fixed parameters from ESM-3B. Additionally, using $L_{infoNCE}$ for aligning graphs and proteins improved results by 3.19, confirming the efficacy of this alignment method.

Pretrain	PPIs Network	LinfoNCE	F1
✓		✓	43.31
	UNI	✓	62.20
✓	OR	✓	66.92
✓	UNI		66.35
✓	UNI	✓	**69.54**

Table 3: Ablation experiments on SHS27k using DFS for data partitioning

5.4 Conclusion

We identified and addressed limitations in current multi-label PPI type predictions based on PPI networks. Our solution, a multimodal large language model named LLaPA, incorporates the PPI network as external knowledge, integrating it into the model via RAG. We developed an innovative modality alignment method that uses pre-aligned protein modalities to facilitate graph modality alignment. LLaPA is capable of predicting affinities for multi-sequence complexes with a flexible number of protein sequences. Additionally, LLaPA shows potential for a wide range of other protein-related tasks.

Limitations

As a large model integrating proteins, PPI networks, and natural language, LLaPA utilizes natural language instructions and a unified training method for downstream tasks. It can accept a flexible number of protein inputs and has the potential to handle more complex protein tasks. However, LLaPA focuses on protein-level features and is ineffective for tasks requiring amino acid-level features, such as PPI binding site prediction and PPI conformation prediction. Additionally, since we directly input protein embeddings into the LLM, we cannot leverage the textual features corresponding to protein entities. This is not an issue for novel proteins, but for well-studied proteins with existing literature, utilizing these resources for better analysis is crucial. Furthermore, constructing a larger and more diverse UPPIN is also very important.

References

Hadi Abdine, Michail Chatzianastasis, Costas Bouyioukos, and Michalis Vazirgiannis. 2024. Prot2text: Multimodal protein's function generation with gnns and transformers. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, pages 10757–10765.

Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. 2000. The protein data bank. Nucleic acids research, 28(1):235–242.

Pascal Braun and Anne-Claude Gingras. 2012. History of protein–protein interactions: From egg-white to complex networks. Proteomics, 12(10):1478–1498.

Patrick Bryant, Gabriele Pozzati, and Arne Elofsson. 2022. Improved prediction of protein-protein interactions using alphafold2. Nature communications, 13(1):1265.

He Cao, Zijing Liu, Xingyu Lu, Yuan Yao, and Yu Li. 2023. Instructmol: Multi-modal integration for building a versatile and reliable molecular assistant in drug discovery. arXiv preprint arXiv:2311.16208.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. 2024. Honeybee: Locality-enhanced projector for multimodal llm. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13817–13827.

Ziwei Chai, Tianjie Zhang, Liang Wu, Kaiqiao Han, Xiaohai Hu, Xuanwen Huang, and Yang Yang. 2023. Graphllm: Boosting graph reasoning ability of large language model. arXiv preprint arXiv:2310.05845.

Muhao Chen, Chelsea J-T Ju, Guangyu Zhou, Xuelu Chen, Tianran Zhang, Kai-Wei Chang, Carlo Zaniolo, and Wei Wang. 2019. Multifaceted protein–protein interaction prediction based on siamese residual rcnn. Bioinformatics, 35(14):i305–i314.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng, Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. arxiv.

James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M Deane. 2014. Sabdab: the structural antibody database. Nucleic acids research, 42(D1):D1140–D1146.

Robert C Edgar and Serafim Batzoglou. 2006. Multiple sequence alignment. Current opinion in structural biology, 16(3):368–373.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023a. Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997.

Ziqi Gao, Chenran Jiang, Jiawen Zhang, Xiaosen Jiang, Lanqing Li, Peilin Zhao, Huanming Yang, Yong Huang, and Jia Li. 2023b. Hierarchical graph learning for protein–protein interaction. Nature Communications, 14(1):1093.

Anne-Claude Gavin, Markus Bösche, Roland Krause, Paola Grandi, Martina Marzioch, Andreas Bauer, Jörg Schultz, Jens M Rick, Anne-Marie Michon, Cristina-Maria Cruciat, et al. 2002. Functional organization of the yeast proteome by systematic analysis of protein complexes. Nature, 415(6868):141–147.

Tao Gong, Chengqi Lyu, Shilong Zhang, Yudong Wang, Miao Zheng, Qian Zhao, Kuikun Liu, Wenwei Zhang, Ping Luo, and Kai Chen. 2023. Multimodal-gpt: A vision and language model for dialogue with humans. arXiv preprint arXiv:2305.04790.

Somaye Hashemifar, Behnam Neyshabur, Aly A Khan, and Jinbo Xu. 2018. Predicting protein–protein interactions through sequence-based deep learning. Bioinformatics, 34(17):i802–i810.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

Takashi Ito, Tomoko Chiba, Ritsuko Ozawa, Mikio Yoshida, Masahira Hattori, and Yoshiyuki Sakaki. 2001. A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proceedings of the National Academy of Sciences, 98(8):4569–4574.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Minhyeok Lee. 2023. Recent advances in deep learning for protein-protein interaction analysis: A comprehensive review. Molecules, 28(13):5169.

Hang Li, Xiu-Jun Gong, Hua Yu, and Chang Zhou. 2018. Deep neural network based predictions of protein interactions using primary sequences. Molecules, 23(8):1923.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In International conference on machine learning, pages 19730–19742. PMLR.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190.

Peicong Lin, Huanyu Tao, Hao Li, and Sheng-You Huang. 2023. Protein–protein contact prediction by geometric triangle-aware protein language models. Nature Machine Intelligence, 5(11):1275–1284.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. 2022. Language models of protein sequences at the scale of evolution enable accurate structure prediction. BioRxiv, 2022:500902.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024a. Visual instruction tuning. Advances in neural information processing systems, 36.

Pengfei Liu, Yiming Ren, Jun Tao, and Zhixiang Ren. 2024b. Git-mol: A multi-modal large language model for molecular science with graph, image, and text. Computers in biology and medicine, 171:108073.

Xianggen Liu, Yunan Luo, Pengyong Li, Sen Song, and Jian Peng. 2021. Deep geometric representations for modeling effects of mutations on protein-protein binding affinity. PLoS computational biology, 17(8):e1009284.

Zhihai Liu, Minyi Su, Li Han, Jie Liu, Qifan Yang, Yan Li, and Renxiao Wang. 2017. Forging the basis for developing protein–ligand interaction scoring functions. Accounts of chemical research, 50(2):302–309.

Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. 2023. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. arXiv preprint arXiv:2310.12798.

Haiying Lu, Qiaodan Zhou, Jun He, Zhongliang Jiang, Cheng Peng, Rongsheng Tong, and Jianyou Shi. 2020. Recent advances in the development of protein–protein interactions modulators: mechanisms and clinical trials. Signal transduction and targeted therapy, 5(1):213.

Yizhen Luo, Jiahuan Zhang, Siqi Fan, Kai Yang, Yushuai Wu, Mu Qiao, and Zaiqing Nie. 2023. Biomedgpt: Open multimodal generative pre-trained transformer for biomedicine. arXiv preprint arXiv:2308.09442.

Guofeng Lv, Zhiqiang Hu, Yanguang Bi, and Shaoting Zhang. 2021. Learning unknown from correlations: Graph neural network for inter-novel-protein interaction prediction. arXiv preprint arXiv:2105.06709.

Liuzhenghao Lv, Zongying Lin, Hao Li, Yuyang Liu, Jiaxi Cui, Calvin Yu-Chian Chen, Li Yuan, and Yonghong Tian. 2024. Prollama: A protein large language model for multi-task protein language processing. arXiv e-prints, pages arXiv–2402.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

Alicia L Richards, Manon Eckhardt, and Nevan J Krogan. 2021. Mass spectrometry-based protein–protein interaction networks for the study of human diseases. Molecular systems biology, 17(1):e8792.

Bosheng Song, Xiaoyan Luo, Xiaoli Luo, Yuansheng Liu, Zhangming Niu, and Xiangxiang Zeng. 2022. Learning spatial structures of proteins improves protein–protein interaction prediction. Briefings in bioinformatics, 23(2):bbab558.

Damian Szklarczyk, John H Morris, Helen Cook, Michael Kuhn, Stefan Wyder, Milan Simonovic, Alberto Santos, Nadezhda T Doncheva, Alexander Roth, Peer Bork, et al. 2016. The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible. Nucleic acids research, page gkw937.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

A Vaswani. 2017. Attention is all you need. Advances in Neural Information Processing Systems.

Menglun Wang, Zixuan Cang, and Guo-Wei Wei. 2020. A topology-based network tree for the prediction of protein–protein binding affinity changes following mutation. Nature Machine Intelligence, 2(2):116–123.

Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. 2024. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. Advances in Neural Information Processing Systems, 36.

James A Wells and Christopher L McClendon. 2007. Reaching for high-hanging fruit in drug discovery at protein–protein interfaces. Nature, 450(7172):1001–1009.

Fandi Wu, Yu Zhao, Jiaxiang Wu, Biaobin Jiang, Bing He, Longkai Huang, Chenchen Qin, Fan Yang, Ningqiao Huang, Yang Xiao, et al. 2024a. Fast and accurate modeling and design of antibody-antigen complex using tfold. bioRxiv, pages 2024–02.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In International conference on machine learning, pages 6861–6871. PMLR.

Lirong Wu, Yijun Tian, Yufei Huang, Siyuan Li, Haitao Lin, Nitesh V Chawla, and Stan Z Li. 2024b. Mape-ppi: Towards effective and efficient protein-protein interaction prediction via microenvironment-aware protein embedding. arXiv preprint arXiv:2402.14391.

Yijia Xiao, Edward Sun, Yiqiao Jin, Qifan Wang, and Wei Wang. 2024. Proteingpt: Multimodal llm for protein property prediction and structure understanding. arXiv preprint arXiv:2408.11363.

Ziwei Xie and Jinbo Xu. 2022. Deep graph learning of inter-protein contacts. Bioinformatics, 38(4):947–953.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? arXiv preprint arXiv:1810.00826.

Yumeng Yan and Sheng-You Huang. 2021. Accurate prediction of inter-protein residue–residue contacts for homo-oligomeric protein complexes. Briefings in bioinformatics, 22(5):bbab038.

Linli Yao, Lei Li, Shuhuai Ren, Lean Wang, Yuanxin Liu, Xu Sun, and Lu Hou. 2024. Deco: Decoupling token compression from semantic abstraction in multimodal large language models. arXiv preprint arXiv:2405.20985.

Guanglei Yu, Qichang Zhao, Xuehua Bi, and Jianxin Wang. 2024. Ddaffinity: predicting the changes in binding affinity of multiple point mutations using protein 3d structure. Bioinformatics, 40(Supplement_1):i418–i427.

Mengmei Zhang, Mingwei Sun, Peng Wang, Shen Fan, Yanhu Mo, Xiaoxiao Xu, Hong Liu, Cheng Yang, and Chuan Shi. 2024. Graphtranslator: Aligning graph model to large language model for open-ended tasks. In Proceedings of the ACM on Web Conference 2024, pages 1003–1014.

Ziyuan Zhao, Peisheng Qian, Xulei Yang, Zeng Zeng, Cuntai Guan, Wai Leong Tam, and Xiaoli Li. 2023. Semignn-ppi: Self-ensembling multi-graph neural network for efficient and generalizable protein-protein interaction prediction. arXiv preprint arXiv:2305.08316.

Le Zhuo, Zewen Chi, Minghao Xu, Heyan Huang, Heqi Zheng, Conghui He, Xian-Ling Mao, and Wentao Zhang. 2024. Protllm: An interleaved protein-language llm with protein-as-word pre-training. arXiv preprint arXiv:2403.07920.

A More implementation details

A.1 Implementation details and hyperparameters

We finetune the parameters of Llama3-8b using LoRA (Hu et al., 2021). The LoRA target modules are `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `down_proj`, `up_proj`, and `lm_head`. The model is trained using 8 NVIDIA A100 GPUs (80G). Other parameters are detailed in Table 4.

Parameter	Value
lora_alpha	64
lora_dropout	0.1
lora_rank	256
learning_rate	4e-5
global_batch	512
lr_scheduler_type	cosine
num_warmup_steps	100
weight_decay	0.05
max_grad_norm	0.03
warmup_ratio	0.03
bf16	TRUE

Table 4: Training parameters.

A.2 Construction of UPPIN

STRING (Homo sapiens subset) is a multi-source PPI network comprising 15,202 unique proteins and 581,161 unique edges. It includes seven types of protein interactions: activation, binding, catalysis, expression, inhibition, posttranslational modification (ptmod), and reaction. For UPPIN, we retained all nodes and edges from STRING but removed the edge labels.

PDBBind is a database derived from the PDB (Protein Data Bank) (Berman et al., 2000), containing biomolecular complexes with experimentally determined binding affinities. We used the 2020 version of PDBBind, which includes 2,852 protein-protein complexes, totaling 5,711 unique proteins. Due to often incomplete protein sequences in the crystal data, we first obtained the fasta data for each protein from the PDB. We then connected each pair of proteins within a complex with an edge, resulting in a total of 5,978 edges.

SAbDab is an antibody-antigen database that includes complexes and experimental information, and it is continuously updated. We used data up to PDB 8cds, comprising 16,226 complexes and 6,315 unique proteins. As with PDBBind, we first obtained the fasta data for each complex from the Protein Data Bank and then constructed edges between each pair of proteins within a complex.

We merged these three datasets to create our UPPIN, which includes a total of 26,180 unique proteins and 594,216 unique edges. Detailed information is provided in Appendix A.2 Table 5.

	nodes	edges
STRING	15,202	581,161
PDB2020	5,711	5,978
SabDab	6,315	7,424
sum	27,228	597,563
unique	26,180	594,216

Table 5: Information on the constructed UPPIN.

A.3 Implementation of models E(2), E(3), E(4)

Models E(2), E(3), and E(4) are all based on ESM2-3B and are used to predict the affinity of complexes consisting of 2, 3, and 4 proteins, respectively. We fixed the parameters of ESM2-3B, encoded the protein sequences, and concatenated them. Then, we trained the predictors of E(2), E(3), and E(4), each of which is a simple linear layer. The learning rate was set to 5e-4, and the models were trained for 500 epochs. The loss function used was mean square error:

$$L_{MSE} = \sum (y_i - \hat{y}_i)^2.$$

A.4 Inputs and Outputs Examples

A.4.1 Example for mPPI task.

INPUTS:

Instruction: There are two proteins, `<|proteinHere|>` and `<|proteinHere|>`. Among the following seven types of relationships (reaction, binding, ptmod, activation, inhibition, catalysis, expression), list all possible relationships between these two proteins. Carefully analyze the given protein features, based on the definition of the seven protein relations, answer this question in the form of 'According to the given protein information, Their relationships include relation(s).' If multiple relationships may exist, separate them with comma. SUPPLEMENTARY INFO: For protein 1, its topological information in the PPI network is `<|graphEmbeddingHere|>`. For protein 2, its topological information in the PPI network is `<|graphEmbeddingHere|>`.

RAG Prompt for Multi-label PPI type Prediction

Instruction: There are two proteins, <|proteinHere|> and <|proteinHere|>. Among the following seven types of relationships (reaction, binding, ptmod, activation, inhibition, catalysis, expression), list all possible relationships between these two proteins. Carefully analyze the given protein features and their corresponding topological information, based on the definition of the seven protein relations, answer this question in the form of 'According to the given protein information, Their relationships include relation(s).' If multiple relationships may exist, separate them with comma. **SUPPLEMENTARY INFO:** For protein 1, the topological information in the PPI network of the protein with a cosine similarity of 1.0 is <|graphEmbeddingHere|>. For protein 2, its topological information in the PPI network is <|graphEmbeddingHere|>.

Proteins: [seq1,seq2]

Protein_idx: [idx1, idx2]

RAG Prompt for Multi-sequence Affinity Prediction

Instruction: There is a complex containing the following protein sequences <|proteinHere|> <|proteinHere|> <...>. What is the binding affinity (logKd) between these proteins? Carefully analyze the given protein features and their corresponding topological information, based on the definition of logKd, answer this question in the form of 'Based on the given protein information, the binding affinity of this compound is logKd= [predicted value].' **SUPPLEMENTARY INFO:** For protein 1, the topological information in the PPI network of the protein with a cosine similarity of 1.0 is <|graphEmbeddingHere|>. For protein 2, its topological information in the PPI network is <|graphEmbeddingHere|>. For protein 3 .....

proteins: [seq1,seq2,...]

protein_idx: [idx1, idx2, ...]

Figure 3: The input template for Graph-based Retrieval Augmented Prompt. It includes text instructions, protein sequences, and the position of the protein (or similar proteins) in the PPI network. The **Instruction** is encoded by the embedding layer of the LLM, the **Proteins** are encoded by ESM2-3B, and the **Protein Indexes** are used to find the corresponding proteins' positions in UPPIN and obtain the embeddings of these nodes in UPPIN encoded by SGC. Finally, these three parts of the encoding are fused to form the input for the LLM.

Protein 1: MGLTVSALFSRIFGKKQMRILMVGLDAAG KTTILYKLKLGEIVTTIPTIGFNVETVEYKNICFTVWDVG GQDKIRPLWRHYFQNTQGLIFVVDSNDRERVQESADELQK MLQEDELRDAVLLVFANKQDMPNAMPVSELTDKLGLQHLR SRTWYVQATCATQGTGLYDGLDWLSHELSKR

Protein 2: MTECFLPPTSSPSEHRRVEHGSGLTRTPS SEEISPTKFPGLYRTGEPSPPHDILHEPPDVVSDDEKDHG KKKGKFKKKEKRTEGYAAFQEDSSGDEAESPSKMKRSKGI HVFKKPSFSKKKEKDFKIKEKPKEEKHKEEKHKEEKKEKK SKDLTAADVVKQWKEKKKKKKPIQEPEVPQIDVPNLKPIF GIPLADAVERTMMYDGIRLPAVFRECIDYVEKYGMKCEGI YRVSGIKSKVDELKAAYDREESTNLEDYEPNTVASLLKQY LRDLPENLLTKELMRFEEACGRTTETEKVQEFQRLLKELP ECNYLLISWLIVHMDHVIAKELETKMNIQNISIVLSPTVQ ISNRVLYVFFTHVQELFGNVVLKQVMKPLRWSNMATMPTL PETQAGIKEEMNENEEVINILLAQENEILTEQEELLAMEQ FLRRQIASEKEEIERLRAEIAEIQSRQQHGRSETEEYSSE SESESEDEEELQIILEDLQRQNEELEIKNNHLNQAIHEER EAIIELRVQLRLLQMQRAKAEQQAQEDEEPEWRGGAVQPP RDGVLEPKAAKEQPKAGKEPAKPSPSRDRKETSI

PPI network Index: 11117, 3205

OUTPUTS: According to the given protein information, their relationships include activation, inhibition, catalysis.

A.4.2 Example for MA task.

INPUTS:

Instruction: There is a complex containing the following proteins <|proteinHere|> <|proteinHere|><|proteinHere|>. What is the binding affinity (log Kd) between these proteins? Carefully analyze the given protein features, based on the definition of log Kd, answer this question in the form of 'Based on the given protein information, the binding affinity of this compound is log Kd = [predicted value].' SUPPLEMENTARY INFO: For protein 1, the topological information in the PPI network of the protein with a cosine similarity of 0.99 is <|graphEmbeddingHere|>. For protein 2, the topological information in the PPI network of the protein with a cosine similarity of 0.8 is <|graphEmbeddingHere|>. For protein 3, its topological information in the PPI network is <|graphEmbeddingHere|>.

Protein 1: MKHHHHHHPMSDYDIPTTENLYFQGAMGT NECLDNNGGCSYVCNDLKIGYECLCPDGFQLVAQRRCEDI DECQDPDTCSQLCVNLEGGYKCQCEEGFQLDPHTKACK

Protein 2: SIPWNLERITPPRYRADEYQPPDGGSLVE VYLLDTSIQSDHREIEGRVMVTDFENVPEEDGTRFHRQAS KCDSHGTHLAGVVSGRDAGVAKGASMRSLRVLNCQGKGTV SGTLIGLEFIRKSQLVQPVGPLVVLLPLAGGYSRVLNAAC QRLARAGVVLVTAAGNFRDDACLYSPASAPEVITVGATNA QDQPVTLGTLGTNFGRCVDLFAPGEDIIGASSDCSTCFVS QSGTSQAAAHVAGIAAMMLSAEPELTLAELRQRLIHFSAK DVINEAWFPEDQRVLTPNLVAALPPSTHGAAGTAAASHHH HHH

Protein 3: MKGSKGSKGSKPMSAEAPEHGTTATFHRC AKDPWRLPGTYVVVLKEETHLSQSERTARRLQAQAARRGY LTKILHVFHGLLPGFLVKMSGDLLELALKLPHVDYIEEDS SVFAQ

PPI network Index: 6091, 7833, 1781

OUTPUTS: Based on the given protein information, the binding affinity of this compound is log Kd = 8.7.

B More experimental results

B.1 More Information on Experimental Data

Refer to Table 6.

	SHS27k			SHS148k		
	random	**dfs**	**bfs**	**random**	**dfs**	**bfs**
BS	90.07%	0.00%	0.00%	95.65%	0.28%	0.00%
ES	9.28%	80.13%	69.54%	4.26%	85.62%	78.10%
NS	0.65%	19.87%	30.46%	0.09%	14.10%	21.90%

Table 6: Proportions of BS (Both have been Seen), ES (Either one protein has been Seen), and NS (Neither one has been Seen) under Different Data Splits.

	SHS27k			SHS148k		
	random	**dfs**	**bfs**	**random**	**dfs**	**bfs**
GNN-PPI	87.81	71.66	66.98	90.48	76.81	71.78
GNN-PPI/R	40.53	43.19	42.52	39.48	40.96	41.42
	47.28 ↓	*28.47* ↓	*24.46* ↓	*51.00* ↓	*35.85*	*30.36* ↓
HIGH-PPI	76.62	71.69	66.75	72.21	77.32	60.08
HIGH-PPI/R	41.51	40.06	39.87	42.81	51.06	35.94
	35.11 ↓	*31.63* ↓	*26.88* ↓	*29.40* ↓	*26.26* ↓	*24.14* ↓
MAPE-PPI	88.91	71.98	70.38	92.87	79.10	74.29
MAPE-PPI/R	76.84	51.69	55.21	85.96	61.45	56.68
	12.07 ↓	*20.29* ↓	*15.17* ↓	*6.91* ↓	*17.65* ↓	*17.61* ↓

Table 7: Comparison of PPI network-based methods on the mPPI task with and without removing test edges.

B.2 More details on the comparison method based on PPI network

Refer to Table 7.

B.3 Details of the data partitioning for the MA task

Refer to Table 8.

Sequence Number	All	Train	Test
2	1857	1485	372
3	679	535	144
4	188	156	32
5	106	89	17
6	6	3	3
7	1	0	1
9	1	1	0
13	1	1	0
14	1	1	0
16	1	1	0
sum	2841	2272	569

Table 8: The PDB2020 dataset's division into training and test sets, and the count of complexes with unique sequences.

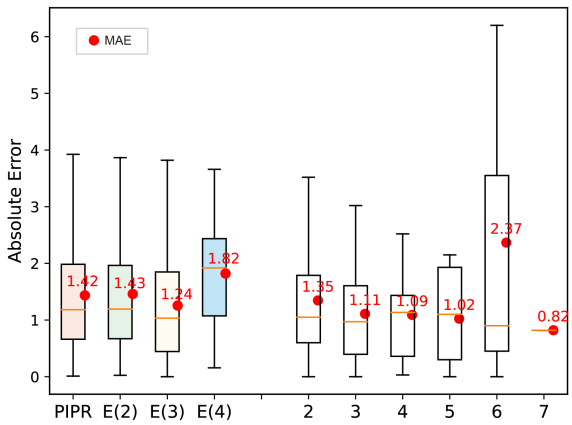

Figure 4: Detailed experimental results for MA task on PDB2020.

B.4 Detailed experimental data for MA task

Refer to Figure 4.

B.5 Additional Experiments on the MA Task

We conducted additional experiments by modifying the MA setup, inputting all protein sequences from a PDB instead of unique protein sequences. The results of these experiments are shown in Table 9.

Sequence Number	Train Number	Test Number	MAE	PCCs
2	603	150	0.7	0.69
3	195	58	0.81	0.73
4	513	115	0.77	0.75
5	136	27	1.04	0.52
6	280	83	1.04	0.47
7	30	8	1.17	0.58
8	161	42	1.04	0.63
9	32	9	1.08	0.8
10	80	20	0.74	0.91
11	27	4	1.26	0.57
12	91	20	0.97	0.51
13	2	0	N\A	N\A
14	21	5	1.9	0.67
15	9	2	2.08	N\A
16	19	4	1.02	-0.32
17	10	0	N\A	N\A
18	21	3	1.05	0.84
19	0	1	0	N\A
20	7	4	1.95	0.5
21	7	3	0.62	0.92
22	4	2	0.68	N\A
23	3	2	0.19	N\A
24	11	2	1.29	N\A
26	1	1	0.24	N\A
27	1	0	N\A	N\A
28	1	0	N\A	N\A
29	1	0	N\A	N\A
30	1	0	N\A	N\A
37	1	0	N\A	N\A
45	1	1	1.29	N\A
48	1	0	N\A	N\A
54	1	0	N\A	N\A
55	1	0	N\A	N\A
59	1	0	N\A	N\A
63	1	0	N\A	N\A
72	1	0	N\A	N\A

Table 9: Experimental results of inputting all sequences of the complex in the MA task.

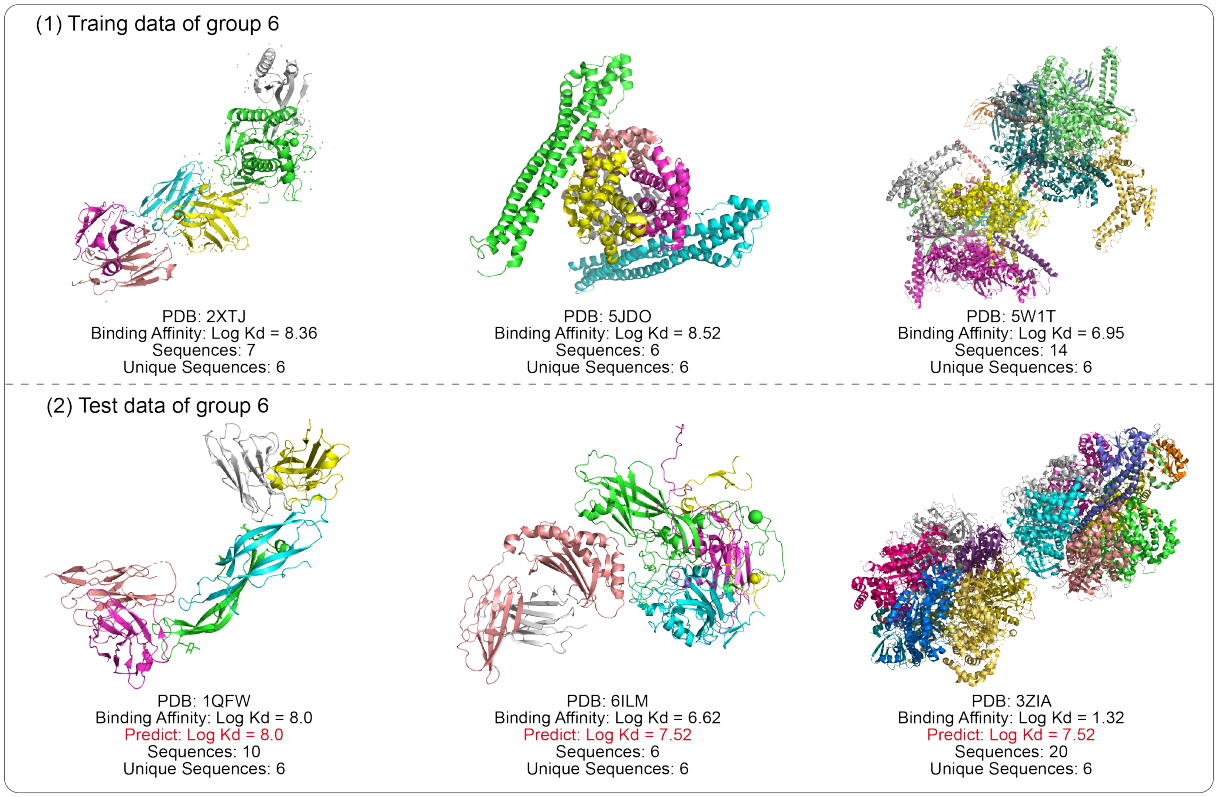

Figure 5: The training data and prediction results of the complex containing 6 unique sequences.

B.6 Examples from Group 6 in the MA task

Refer to Figure 5.

C More Related work: Injecting Graphs to Large Language Model

Graph data structures have a wide range of real-world applications, such as in financial networks, social networks, and protein interaction networks. When it comes to modeling complex network relationships, graph data structures offer unique advantages. Graph-structured data can also be considered a type of multimodal data, and the approach of injecting multimodal features into LLMs is equally applicable to graph data. For example: GraphLLM (Chai et al., 2023) uses Prefix-tuning (Li and Liang, 2021) to prepend graph features to the input of LLaMA (Touvron et al., 2023). GIT-Mol (Liu et al., 2024b) and Molca (Liu et al., 2023): These models use Q-former (Li et al., 2023) to align graph features with LLMs. The Q-former helps in bridging the gap between graph representations and the input space of LLMs. InstructMol (Cao et al., 2023) employs a mapping matrix W to map the encoded features of molecular graphs into the input space of LLMs. GraphTranslator (Zhang et al., 2024) uses a Transformer-based Translator Module (Vaswani, 2017) to convert node features of the graph into learnable token embeddings. These methods demonstrate how graph data can be integrated into LLMs, leveraging the strengths of both graph structures and large language models to handle complex, multimodal information. This integration not only improves the models' ability to understand and process graph data but also extends their applicability to a broader range of tasks involving networked data.