SemiGNN-PPI: Self-Ensembling Multi-Graph Neural Network for Efficient and Generalizable Protein-Protein Interaction Prediction

Ziyuan Zhao[1,2], **Peisheng Qian**[1], **Xulei Yang**[1*], **Zeng Zeng**[3], **Cuntai Guan**[2], **Wai Leong Tam**[4] and **Xiaoli Li**[1,2]

[1]Institute for Infocomm Research (I^{2}R), A*STAR, Singapore
[2]School of Computer Science and Engineering (SCSE), Nanyang Technological University, Singapore
[3]School of Microelectronics, Shanghai University, China
[4]Genome Institute of Singapore (GIS), A*STAR, Singapore
{zhaoz, qianp ,yangx}@i2r.a-star.edu.sg, zengz@shu.edu.cn, ctguan@ntu.edu.sg, tamwl@gis.a-star.edu.sg, xlli@i2r.a-star.edu.sg

Abstract

Protein-protein interactions (PPIs) are crucial in various biological processes and their study has significant implications for drug development and disease diagnosis. Existing deep learning methods suffer from significant performance degradation under complex real-world scenarios due to various factors, *e.g.*, label scarcity and domain shift. In this paper, we propose a self-ensembling multi-graph neural network (SemiGNN-PPI) that can effectively predict PPIs while being both efficient and generalizable. In SemiGNN-PPI, we not only model the protein correlations but explore the label dependencies by constructing and processing multiple graphs from the perspectives of both features and labels in the graph learning process. We further marry GNN with Mean Teacher to effectively leverage unlabeled graph-structured PPI data for self-ensemble graph learning. We also design multiple graph consistency constraints to align the student and teacher graphs in the feature embedding space, enabling the student model to better learn from the teacher model by incorporating more relationships. Extensive experiments on PPI datasets of different scales with different evaluation settings demonstrate that SemiGNN-PPI outperforms state-of-the-art PPI prediction methods, particularly in challenging scenarios such as training with limited annotations and testing on unseen data.

1 Introduction

Protein-protein Interactions (PPIs) are central to various cellular functions and processes, such as signal transduction, cell-cycle progression, and metabolic pathways [Acuner Ozbabacan *et al.*, 2011]. Therefore, the identification and characterization of PPIs are of great importance for understanding protein functions and disease occurrence, which can potentially facilitate therapeutic target identification [Petta *et al.*, 2016] and the novel drug design [Skrabanek *et al.*, 2008]. In past decades, high-throughput experimental methods, *e.g.*, yeast two-hybrid screens (Y2H) [Fields and Song, 1989], and mass spectrometric protein complex identification (MS-PCI) [Ho *et al.*, 2002] have been developed to identify PPIs. Nevertheless, genome-scale experiments are expensive, tedious, and time-consuming while suffering from high error rates and low coverage [Luo *et al.*, 2015]. As such, there is an urgent need to establish reliable computational methods to identify PPIs with high quality and accuracy.

In recent years, a large variety of high-throughput computational approaches for PPI prediction have been proposed, which can be broadly divided into two groups: classic machine learning (ML)-based methods [Browne *et al.*, 2007; Lin and Chen, 2013; Guo *et al.*, 2008; Wong *et al.*, 2015; Chen and Liu, 2005] and deep learning (DL)-based methods [Sun *et al.*, 2017; Du *et al.*, 2017; Hashemifar *et al.*, 2018; Chen *et al.*, 2019a; Lv *et al.*, 2021]. Compared to classic ML methods, DL algorithms are capable of processing complicated and large-scale data and extracting useful features automatically, achieving significant success in a diverse range of bioinformatics applications [Min *et al.*, 2017; Soleymani *et al.*, 2022], including PPI prediction [Soleymani *et al.*, 2022]. Most existing DL-based methods treat interactions as independent instances, ignoring protein correlations. PPI can be naturally formulated as graph networks with proteins and interactions represented as nodes and edges, respectively [Margolin *et al.*, 2006; Pio *et al.*, 2020]. To improve PPI prediction performance, recent works [Yang *et al.*, 2020; Lv *et al.*, 2021] have been proposed to investigate the correlations between PPIs using various graph neural network (GNN) architectures [Kipf and Welling, 2016; Xu *et al.*, 2019]. However, they are limited by ignoring learning label dependencies for multi-type PPI prediction. It has recently become common practice to employ Graph Convolutional Networks (GCNs) to capture label correlation in a wide range of multi-label tasks [Chen *et al.*, 2019b; Wang *et al.*, 2020]. Nevertheless, multi-label learning utilizing label graphs predominantly works in the visual domain and has yet to be extended to PPI prediction tasks.

In general, a desired PPI prediction framework should be efficient, transferable, and generalizable, whereas two ma-

*Corresponding author

jor bottlenecks deriving from imperfect datasets have hindered the development of such models. **Label scarcity:** Despite the tremendous progress in PPI research using various computational and experimental methods, many interactions still need to be annotated from experimental data. Consequently, only a small portion of labeled samples can be used for model training. It can be a significant bottleneck in obtaining robust and accurate PPI prediction models. **Domain shift:** Most existing methods are only developed and validated using in-distribution data (*i.e.*, trainset-homologous testsets), receiving severe performance degradation when being deployed to unseen data with different distributions (*i.e.*, trainset-heterologous testsets). Although [Lv *et al.*, 2021] design new evaluations to better reflect model generalization, giving instructive and consistent assessment across datasets, the domain shift issue still needs to be fully explored for PPI prediction. Therefore, how to deal with imperfect data for improving model efficiency and generalization remains a vital issue in PPI prediction. Recent studies [Zhang *et al.*, 2021; Zhao *et al.*, 2022] show that self-ensemble methods with semi-supervised learning (SSL) [Laine and Aila, 2017; Tarvainen and Valpola, 2017] have demonstrated effectiveness in addressing both label scarcity and domain shift.

In this work, to tackle the above challenges and limitations, we propose an efficient and generalizable **PPI** prediction framework, referred to as **S**elf-**e**nsembling **m**ult**i**-**G**raph **N**eural **N**etwork (**SemiGNN-PPI**). Firstly, we propose leveraging graph structure to model protein correlations and label dependencies for multi-graph learning. Specifically, we learn inter-dependent classifiers to extract information from the label graph, which are then applied to the protein representations aggregated by neighbors in the protein graph for multi-type PPI prediction. Secondly, we propose combining GNN with Mean Teacher [Tarvainen and Valpola, 2017], a powerful SSL model, to explore unlabeled data for self-ensemble graph learning. In our framework, the student model learns to classify the labeled data accurately and also distills the knowledge beneath unlabeled data from the teacher model with multiple graph consistency constraints for improving the model performance under complex scenarios. To the best of our knowledge, this is the first study to explore efficient and generalizable multi-type PPI prediction. Precisely, the main contributions of the work can be summarized as follows:

- For multi-type PPI prediction, we first investigate the limitations and challenges of existing methods under complex but realistic scenarios, and then propose an effective **S**elf-**e**nsembling **m**ult**i**-**G**raph **N**eural **N**etwork-based **PPI** prediction (**SemiGNN-PPI**) framework for improving model efficiency and generalization.
- In SemiGNN-PPI, we construct multiple graphs to learn correlations between proteins and label dependencies simultaneously. We further advance GNN with Mean Teacher to effectively utilize unlabeled data by consistency regularization with multiple constraints.
- Extensive experiments on three PPI datasets with different settings demonstrate that SemiGNN-PPI outperforms other state-of-the-art methods for multi-label PPI prediction under various challenging scenarios.

2 Related Work

Protein–Protein Interaction Prediction. Amino acid sequence-based methods have received considerable attention in PPI prediction. Early works leverage machine learning (ML) techniques [Browne *et al.*, 2007; Chen and Liu, 2005; Lin and Chen, 2013; Guo *et al.*, 2008; Wong *et al.*, 2015] to map pairs of handcrafted sequence features of proteins to interaction types. With the advent of deep learning (DL), more recent works have utilized deep neural networks [Sun *et al.*, 2017; Hashemifar *et al.*, 2018; Du *et al.*, 2017; Chen *et al.*, 2019a; Lv *et al.*, 2021] to automatically extract features from protein sequences for enhancing feature representation. Furthermore, the latest works consider protein correlations and utilize graph neural networks (GNN) to model graph-structured PPI data [Yang *et al.*, 2020; Kipf and Welling, 2016; Lv *et al.*, 2021]. However, it is essential to explore label dependencies for improving the model performance, which has long been ignored for multi-type PPI prediction. Moreover, the generalization and efficiency problems for PPI prediction are still under-explored under complex scenarios, such as data scarcity and distribution shift.

Multi-Label Learning. MLL addresses the problem of assigning multiple labels to a single instance. It has been utilized successfully in numerous fields, *e.g.*, computer vision [Liu *et al.*, 2021; Xu *et al.*, 2022]. Traditional MLL methods typically train independent classifiers for all labels but fail to consider the potential label interdependence, leading to suboptimal performance. Recent trends in MLL incorporate deep learning to capture the label dependencies [Wang *et al.*, 2016; Guo *et al.*, 2019]. For example, CNN-RNN [Wang *et al.*, 2016] leverages recurrent neural networks (RNNs) to transform the label vectors into an embedded space to learn label correlations implicitly. More recently, graph-based MLL methods have aroused great attention from researchers [Chen *et al.*, 2019b; Wang *et al.*, 2020]. Especially, ML-GCN [Chen *et al.*, 2019b] successfully applies Graph Convolutional Network (GCN) by constructing a directed graph over object labels to explicitly model the label dependencies adaptively. In this regard, we propose to explore correlations between PPI types with GCN on the structured label graph for more accurate PPI prediction.

Learning from Imperfect Data. In recent years, deep learning has made tremendous progress in numerous domains, *e.g.*, computer vision and bioinformatics. However, the applicability of deep learning is limited by heavy reliance on training data. We rarely have a perfect dataset for model training [Tajbakhsh *et al.*, 2020; Bekker and Davis, 2020], especially in biomedical imaging and bioinformatics [Zhao *et al.*, 2021; Lu *et al.*, 2022; Qu and Hickey, 2022; Pio *et al.*, 2022]. The commonly encountered challenges in PPI prediction include label scarcity, where only limited annotations are available for training (semi-supervised learning, SSL), and domain shift, where unseen data (target domain) with different distributions from training data (source domain) is used for testing (unsupervised domain adaptation, UDA). In this regard, model efficiency and generalization would be heavily constrained, limiting the wide real-world applications. Self-ensemble learning [Laine and Aila, 2017]

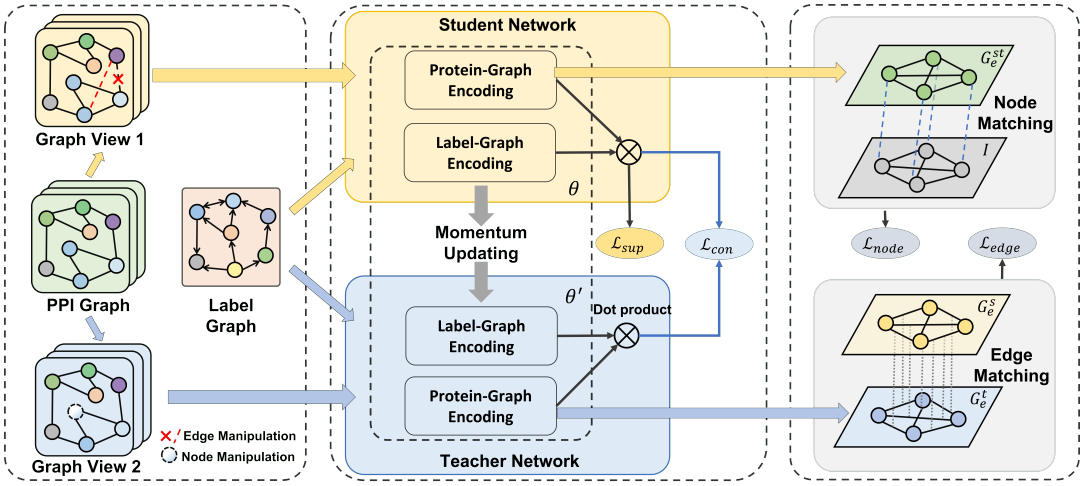

Figure 1: The overall framework of SemiGNN-PPI. First, we generate two augmented graph views with node and edge manipulations. Then, protein graphs and label graphs are fed into the multi-graph teacher-student network, which models both protein relations and label dependencies for self-ensemble learning. Simultaneously, to better capture fine-grained structural information, we align student and teacher feature embeddings by jointly optimizing multiple graph consistency constraints (node matching and edge matching).

is one of the most prevalent methods for SSL, which works by enforcing consistency in model predictions from different epochs with the network parameter average [Tarvainen and Valpola, 2017]. Recently, self-ensemble learning has been extended to visual domain adaptation tasks [Choi *et al.*, 2019; Zhang *et al.*, 2021; Zhao *et al.*, 2022], achieving promising UDA performance. Inspired by these observations, we advance GNN with self-ensemble learning to handle imperfect data for efficient and generalizable PPI prediction.

3 Methodology

3.1 Task Definition

Given a set of proteins $P = \{p_0, p_1, ..., p_n\}$ and a set of PPIs $E = \{e_{ij} = \{p_i, p_j\} | i \neq j, p_i, p_j \in P, I(e_{ij}) \in 0, 1\}$, where $I(e_{ij})$ is a binary PPI indicator function that is 1 if the PPI between proteins p_i and p_j has been confirmed, and 0 otherwise, the types of PPI can be represented by the label space $C = \{c_0, c_1, ..., c_{t-1}\}$ with t different types of interactions, and the labels for a confirmed PPI e_{ij} can be represented as $y_{ij} \subseteq C$. The goal of multi-type PPI learning is to learn a function $f : e_{ij} \rightarrow \hat{y}_{ij}$ from the training set E^s_{train} such that for any PPI $e_{ij} \in E^s_{test}$, $\hat{y}_{ij}$ is the set of predicted labels for e_{ij}. To investigate the efficiency and generalization under complex scenarios beyond the supervised learning setting, we introduce the settings of semi-supervised learning (SSL) and unsupervised domain adaptation (UDA). In the SSL setting, the training datasets consist of limited labeled data E^l_{train} and unlabeled data E^u_{train} due to label scarcity. In the UDA setting, the model trained on E^s_{train} is tested on the unseen data E^t_{test} with different distribution.

3.2 Overview

Fig. 1 depicts the overview of our proposed SemiGNN-PPI framework. We first construct the multi-graph encoding (MGE) module to effectively leverage available labeled data, which includes a protein graph encoding (PGE) network for exploring protein relations and a label graph encoding (LGE) network for learning label dependencies. To exploit knowledge from unlabeled data, we build a teacher network with the same architecture as the student network. During teacher-student training, multiple graph consistency constraints at both node and edge levels are utilized to enhance knowledge distillation for self-ensemble multi-graph learning.

3.3 Multi-Graph Encoding

Protein-Graph Encoding. Early works [Yang *et al.*, 2020; Lv *et al.*, 2021] have demonstrated the effectiveness of graph neural networks (GNNs) on PPI prediction. Considering the correlation of PPIs, we use proteins as nodes and PPIs as edges to build the PPI graph $G = (P, E)$. Then, the PPI prediction can be formulated from $f(e_{ij}|p_i, p_j, \theta) \rightarrow \hat{y}_{ij}$ to $f(e_{ij}|G, \theta) \rightarrow \hat{y}_{ij}$. GNNs take the graph structure and sequence-based protein attributes as inputs to model high-level compact representation of the nodes (proteins), denoted by $H \in \mathbb{R}^{|P| \times d}$ where $h_p = H[p, :]$ is the latent representation of node p, and d is the dimensionality of protein features. In general, GNNs follow a recursive neighborhood aggregation scheme to iteratively update the representation of each node by aggregating and transforming the representations of its neighboring nodes. After l iterations, the transformed feature of node p can be denoted as:

$$h_p^{(l)} = \phi^{(l)}(h_p^{(l-1)}, f^{(l)}(\{h_p^{(l-1)} : u \in \mathcal{N}_k(p)\})), \quad (1)$$

where $\mathcal{N}_k(p)$ denotes the set of k-hop neighbors of the node p; $f^{(l)}$ and $\phi^{(l)}$ are an aggregation function and a combination function, respectively. Following Graph Isomorphism Network (GIN) [Xu *et al.*, 2019], we adopt the summation function to aggregate the representations of neighboring nodes and use the multi-layer perceptrons (MLPs) to update the aggregated features. Then, the update rule of the hidden node

features with a learnable parameter ϵ in PGE is defined as:

$$h_p^{(l)} = g^l((1+\epsilon^l)\cdot h_p^{(l-1)} + \sum\nolimits_{u\in\mathcal{N}_k(p)} h_u^{(l-1)}). \quad (2)$$

Label-Graph Encoding. In multi-label PPI prediction, correlations exist among different types of interactions, *i.e.*, some PPI types may appear together frequently while others rarely appear together. Following [Chen *et al.*, 2019b], we model the interdependencies between different PPI types (labels) using a graph and learn inter-dependent classifiers with Graph Convolutional Network (GCN), which can be directly applied to protein features for multi-type PPI prediction. GCN aims to learn a function $f(\cdot,\cdot)$ on the graph with t nodes. Each GCN layer can be formulated as follows:

$$h_c^{(l+1)} = f(h_c^{(l)}, A), A \in \mathbb{R}^{t\times t}, \quad (3)$$

where $h_c^{(l+1)} \in \mathbb{R}^{t\times d_l'}$ and $h_c^{(l)} \in \mathbb{R}^{t\times d_l}$ are the learned d_l'-dimensional node features from current layer and the d_l-dimensional node features from previous layer, respectively. A is the corresponding correlation matrix. With the convolutional operation, $f(\cdot,\cdot)$ can be further expressed as:

$$h_c^{(l+1)} = \delta\left(\widehat{A}h_c^{(l)}W^l\right), \quad (4)$$

where $\delta(\cdot)$ is a non-linear function set as LeaklyReLU following [Chen *et al.*, 2019b], $\widehat{A}$ is the normalized version of A and $W^l \in \mathbb{R}^{d_l'\times d_l}$ is a transformation matrix. We leverage stacked GCNs to learn inter-dependent classifiers W. The first GCN layer takes word embeddings $E_l \in \mathbb{R}^{|t|\times d_l}$ of labels and the correlation matrix $A \in \mathbb{R}^{t\times t}$ as inputs. Considering that PPI type names are semantic, we apply the BioWordVec model [Zhang *et al.*, 2019] pretrained on the biomedical corpus for generating word embeddings E_l of each PPI type to better capture their semantics. To construct the label correlation matrix A, we compute the conditional probability of different labels within the training dataset. To avoid noises and over-smoothing, we binarize A with a threshold τ and then re-weight it with a weight p to obtain $\widehat{A}$.

Multi-Graph Based Classifier Learning. By applying the learned classifiers $W = \{w_i\}_{i=1}^t$ from label graph encoding (LGE) to the learned representations from protein graph encoding (PGE) for the PPI e_{ij}, we can obtain the predicted scores $\hat{y}_{ij}$, expressed as:

$$\hat{y}_{ij} = W(h_{p_i}\cdot h_{p_j}). \quad (5)$$

We use the traditional multi-label classification loss function to update the whole network in an end-to-end manner. The loss function can be written as:

$$\mathcal{L}_{sup} = \sum_{c=1}^{t}\left(y^c\log\left(\sigma\left(\hat{y}^c\right)\right) + \left(1-y^c\right)\log\left(1-\sigma\left(\hat{y}^c\right)\right)\right),$$

where $\sigma(\cdot)$ is the sigmoid function. Our model learns the aggregated features by combining protein neighbors and models the label correlations by learning inter-dependent classifiers simultaneously to improve the model generalization. In multi-graph learning, the learned classifiers are expected to be neighborhood aware at both feature and label levels.

3.4 Self-ensemble Graph Learning

To leverage unlabeled data, we adopt the mean teaching architecture for unsupervised learning, as shown in Fig. 1. We construct a teacher network f_t with the same architecture as the student network f_s based on self-ensembling [Tarvainen and Valpola, 2017]. Specifically, in each training iteration k, we update the teacher model weights θ' with the exponential moving average (EMA) weights of the student model θ by leveraging the momentum updating mechanism:

$$\theta_{\mathrm{k}}' = m\theta_{\mathrm{k}-1}' + (1-m)\theta_{\mathrm{k}}, \quad (6)$$

where m is momentum. During training, the student model is encouraged to be consistent with the teacher predictions for the inputs with different augmentations. Because of the non-euclidean graph structure, image augmentations such as crop and rotation cannot be directly applied to graphs. To facilitate self-ensemble graph learning, we construct two graph data augmentation methods at both the edge and node levels, *i.e.*, **Edge Manipulation** and **Node Manipulation** to augment graph topological and attribute information [You *et al.*, 2020]. **Edge Manipulation (EM):** To improve the robustness against connectivity variations, we randomly replace a certain percentage of edges in the input to the student and teacher models, since some edges (PPIs) between different nodes (proteins) may be unidentified or wrong in experimental procedures. Specifically, we follow an i.i.d. uniform distribution to randomly replace $em_s\%$ and $em_t\%$ of edges in the input to the student and the teacher, respectively. Different from [You *et al.*, 2020], we replace the dropped edge by linking the node with one of its neighbor's neighboring nodes for maintaining global structural information, *i.e.,* node p_s with a dropped edge connecting to node p_t could be linked to $p_u \in \{p_u | e_{ut} = 1\}$. **Node Manipulation (NM):** To improve the robustness against attribute missing, we randomly remove $nm_s\%$ and $nm_t\%$ of node features, mask them with zeros and feed them into the student and teacher models respectively, to expect the model to effectively learn the features even in the presence of missing attribute information. We construct two graph views with augmentations above to feed the student and teacher networks separately, and encourage them to generate consistent predictions using ℓ_2 loss:

$$\mathcal{L}_{con} = \|f_t(E_u|G,\theta_k',\xi') - f_s(E_u|G,\theta_k,\xi)\|_2, \quad (7)$$

where E_u is unlabeled PPIs in a batch. ξ' and ξ are different augmentation operations. We randomly comprise the different augmentations in our experiments to avoid overfitting and improve model generalization.

3.5 Graph Consistency Constraint

The consistency regularization enforces instance-wise invariance in the prediction space towards different augmentations on the same input, describing the PPI interactions between samples. For the graph-based PPI prediction task, we also need to optimize the model in the feature space, as protein nodes in the testing set differ from the training set and PPI is performed as the relationships between proteins by feature representations extracted from neighboring proteins. Therefore, we model the fine-grained structural protein-protein relations in the feature embedding space [Ma *et al.*, 2022]. We

Method		SHS27k			SHS148k			STRING		
		Random	DFS	BFS	Random	DFS	BFS	Random	DFS	BFS
ML	RF	$78.45_{0.88}$	$35.55_{2.22}$	$37.67_{1.57}$	$82.10_{0.20}$	$43.26_{3.43}$	$38.96_{1.94}$	$88.91_{0.08}$	$70.80_{0.45}$	$55.31_{1.02}$
	LR	$71.55_{0.93}$	$48.51_{1.87}$	$43.06_{5.05}$	$67.00_{0.07}$	$51.09_{2.09}$	$47.45_{1.42}$	$67.74_{0.16}$	$61.28_{0.53}$	$50.54_{2.00}$
DL	DPPI	$73.99_{5.04}$	$46.12_{3.02}$	$41.43_{0.56}$	$77.48_{1.39}$	$52.03_{1.18}$	$52.12_{8.70}$	$94.85_{0.13}$	$66.82_{0.29}$	$56.68_{1.04}$
	DNN-PPI	$77.89_{4.97}$	$54.34_{1.30}$	$48.90_{7.24}$	$88.49_{0.48}$	$58.42_{2.05}$	$57.40_{9.10}$	$83.08_{0.11}$	$64.94_{0.93}$	$53.05_{0.82}$
	PIPR	$83.31_{0.75}$	$57.80_{3.24}$	$44.48_{4.44}$	$90.05_{2.59}$	$63.98_{0.76}$	$61.83_{10.23}$	$94.43_{0.10}$	$67.45_{0.34}$	$55.65_{1.60}$
Graph	GNN-PPI	$87.91_{0.39}$	$74.72_{5.26}$	$63.81_{1.79}$	$92.26_{0.10}$	$82.67_{0.85}$	$71.37_{5.33}$	$95.43_{0.10}$	$91.07_{0.58}$	$78.37_{5.40}$
	GNN-PPI*	$88.87_{0.23}$	$75.68_{3.95}$	$68.84_{3.16}$	$92.13_{0.10}$	$83.77_{1.34}$	$69.02_{3.07}$	$94.94_{0.17}$	$90.62_{0.23}$	$79.76_{2.43}$
M-Graph	SemiGNN-PPI	$\mathbf{89.51_{0.46}}$	$\mathbf{78.32_{3.15}}$	$\mathbf{72.15_{2.87}}$	$\mathbf{92.40_{0.22}}$	$\mathbf{85.45_{1.17}}$	$\mathbf{71.78_{3.56}}$	$\mathbf{95.57_{0.08}}$	$\mathbf{91.23_{0.26}}$	$\mathbf{80.84_{2.05}}$

Table 1: Performance of SemiGNN-PPI and baseline methods over different datasets and data partition schemes. GNN-PPI: reported results in the original paper. GNN-PPI*: reproduced GNN-PPI results. The scores are presented in the format of $\mathrm{mean}_{\mathrm{std}}$.

denote the features extracted from protein-graph encoding as z_s and z_t for the student and teacher networks, respectively. **Edge matching:** We construct the student embedding graph G_e^s and the teacher embedding graph G_e^t by calculating all pairwise Pearson's correlation coefficient (PCC) between nodes in the same batch. Then, we enforce the student network to encode consistent instance-wise correlations with the teacher network in the embedding feature space by applying the edge matching loss:

$$\mathcal{L}_{edge} = ||\text{Adj}(G_e^s) - \text{Adj}(G_e^t)||_2, \tag{8}$$

where Adj refers to the adjacency matrix. **Node matching:** We further formulate the edge embedding graph G_e^{st} by calculating all pairwise PCC between student encoding z_s and teacher encoding z_t in the same batch. To explicitly align encoding of the same protein from the teacher and the student network, we design a node matching loss:

$$\mathcal{L}_{node} = ||\text{diag}(\text{Adj}(G_e^{st})) - \text{diag}(I)||_2, \tag{9}$$

where diag is an operator to create a block-diagonal matrix with the off-diagonal elements of 0, and I refers to the identity matrix. In this regard, we jointly leverage labeled and unlabeled data with graph learning in both protein and label spaces and consistency regularization in both prediction and feature spaces for PPI prediction. The overall objective function is defined as:

$$\mathcal{L} = \mathcal{L}_{sup} + \lambda_{con}\mathcal{L}_{con} + \lambda_{edge}\mathcal{L}_{edge} + \lambda_{node}\mathcal{L}_{node}, \tag{10}$$

where λ_{con}, λ_{edge} and λ_{node} are scaling factors for $\mathcal{L}_{con}$, $\mathcal{L}_{edge}$ and $\mathcal{L}_{node}$, respectively.

4 Experiment

4.1 Dataset

We perform extensive experiments on three datasets, *i.e.,* STRING, SHS148k, and SHS27k. First, we use the multi-label PPI data of Homo sapiens from the STRING database [Szklarczyk *et al.*, 2019] for training and evaluation, including $15,355$ proteins and $593,397$ PPIs. The PPIs are annotated with 7 types, *i.e.,* Activation, Binding, Catalysis, Expression, Inhibition, Post-translational modification (Ptmod), and Reaction. Each PPI is labeled with at least one of them. Moreover, we use two subsets of Homo sapiens PPIs from STRING, *i.e.*, SHS27k, and SHS148k [Chen *et al.*, 2019a], to further validate the proposed approach. SHS27k contains $1,690$ proteins and $7,624$ PPIs, while SHS148k contains $5,189$ proteins and $44,488$ PPIs.

4.2 Experimental Details

Experimental Settings. We follow partition algorithms in GNN-PPI [Lv *et al.*, 2021], including random, breath-first search (BFS), and depth-first search (DFS) to split the trainsets and testsets. For in-depth analysis, PPIs in the testset can be divided into **BS subset** (both proteins of the PPI are present in the labeled trainset), **ES subset** (either one protein of the PPI is present in the labeled trainset), and **NS subset** (neither of the proteins is present in the labeled trainset). The BFS and DFS partition schemes create more challenging paradigms than the random partitioning by including more ES and NS proteins in the testsets for the inter-novel protein interactions [Lv *et al.*, 2021]. In fully supervised experiments, we select 20% of the whole dataset for testing using the partition schemes mentioned above and use the rest for training. To simulate the label scarcity scenario, we randomly select 5%, 10%, and 20% samples from the trainset as the labeled data while keeping the rest as the unlabeled data. To assess the generalization capacity of our method, we evaluate our method trained with one dataset on another dataset, *i.e.*, a trainset-heterologous testset.

Evaluation Metrics. We use the F1 score to evaluate the model performance for multi-label PPI prediction. The score is micro-averaged over all 7 classes. The means and variances of F1 scores over three repeated experiments are reported as results, formatted as $\mathrm{mean}_{\mathrm{std}}$.

Model Training. 1) *Base train*: We follow GNN-PPI [Lv *et al.*, 2021] for protein-independent encoding to extract protein features from protein sequences as inputs to our framework. We initialize the multi-graph encoding network using the labeled data for 300 epochs with an initial learning rate of 0.001 and the Adam optimizer. 2) *Joint train*: Then, we train the self-ensemble graph learning framework on both labeled and unlabeled trainsets for 300 epochs. For label graph construction, we select the binarization threshold $\tau = 0.05$ and the re-weighting factor $p = 0.25$. We randomly comprise the different manipulations in our experiments to avoid overfitting and improve model generalization during joint training. For manipulation ratios, we use higher ratios for the student inputs so that the student can better distill knowledge from the teacher during self-ensemble learning. More specifically, the edge manipulation ratios $em_s\%$ and $em_t\%$ are fixed at 10% and 5%, respectively. The node manipulation rates $nm_s\%$, and $nm_t\%$ are set to 10% and 5%, respectively. To scale the components of the loss function, we set the value of λ_{con}, λ_{edge} and λ_{node} as 0.02, 0.01 and 0.003, respectively. More details are shown in Supplementary Material.

Method	STRING				SHS148k				SHS27k			
	5%	10%	20%	100%	5%	10%	20%	100%	5%	10%	20%	100%
Partition Scheme = Random												
GNN-PPI	$89.94_{0.29}$	$92.38_{0.51}$	$93.30_{0.56}$	$94.94_{0.17}$	$79.19_{0.67}$	$82.86_{0.49}$	$86.67_{0.22}$	$92.13_{0.10}$	$52.04_{3.32}$	$60.28_{12.26}$	$79.44_{1.19}$	$88.87_{0.23}$
Ours	$\mathbf{90.55_{0.10}}$	$\mathbf{92.66_{0.59}}$	$\mathbf{93.90_{0.41}}$	$\mathbf{95.57_{0.08}}$	$\mathbf{79.50_{0.31}}$	$\mathbf{83.48_{0.30}}$	$\mathbf{87.38_{0.24}}$	$\mathbf{92.40_{0.22}}$	$\mathbf{57.97_{1.13}}$	$\mathbf{62.67_{11.26}}$	$\mathbf{81.01_{0.47}}$	$\mathbf{89.51_{0.46}}$
Partition Scheme = DFS												
GNN-PPI	$86.60_{0.37}$	$87.91_{0.30}$	$89.42_{0.46}$	$90.62_{0.23}$	$68.77_{11.20}$	$78.36_{2.23}$	$80.96_{1.61}$	$83.77_{1.34}$	$53.41_{1.64}$	$58.43_{2.27}$	$65.73_{4.18}$	$75.68_{3.95}$
Ours	$\mathbf{87.54_{0.06}}$	$\mathbf{88.98_{0.26}}$	$\mathbf{90.23_{0.12}}$	$\mathbf{91.23_{0.26}}$	$\mathbf{69.94_{9.57}}$	$\mathbf{81.12_{0.98}}$	$\mathbf{83.63_{0.86}}$	$\mathbf{85.45_{1.17}}$	$\mathbf{58.48_{1.11}}$	$\mathbf{61.18_{1.98}}$	$\mathbf{70.31_{2.38}}$	$\mathbf{78.32_{3.15}}$
Partition Scheme = BFS												
GNN-PPI	$71.35_{4.67}$	$74.94_{2.35}$	$79.99_{2.75}$	$79.76_{2.43}$	$61.42_{3.29}$	$62.51_{3.07}$	$67.10_{3.48}$	$69.02_{3.07}$	$57.93_{4.11}$	$56.84_{12.19}$	$61.18_{6.58}$	$68.84_{3.16}$
Ours	$\mathbf{73.35_{4.90}}$	$\mathbf{76.94_{2.53}}$	$\mathbf{81.39_{2.44}}$	$\mathbf{80.84_{2.05}}$	$\mathbf{64.86_{2.97}}$	$\mathbf{68.76_{1.62}}$	$\mathbf{71.06_{3.35}}$	$\mathbf{71.78_{3.56}}$	$\mathbf{60.15_{2.09}}$	$\mathbf{66.13_{2.01}}$	$\mathbf{67.69_{8.47}}$	$\mathbf{72.15_{2.87}}$

Table 2: Performance comparison of different methods under different label ratios. The scores are presented in the format of $\mathrm{mean}_{\mathrm{std}}$.

Method	% Labels	Random Partition			DFS Partition		BFS Partition	
	100	BS (92.66%)	ES (6.95%)	NS(0.39%)	ES (75.95%)	NS(24.05%)	ES (85.70%)	NS(14.30%)
GNN-PPI		89.17	72.44	50.00	77.81	63.44	71.03	44.80
SemiGNN-PPI		**89.68**	**72.93**	50.00	**81.75**	**66.32**	**75.14**	**57.00**
	20	BS (73.18%)	ES (24.98%)	NS (1.84%)	ES (72.87%)	NS (27.13%)	ES (47.71%)	NS (52.29%)
GNN-PPI		83.46	70.10	43.68	64.40	54.21	**59.04**	66.33
SemiGNN-PPI		**84.09**	**71.95**	**45.78**	**73.30**	**55.46**	58.10	**73.82**
	10	BS (55.80%)	ES (38.03%)	NS (6.16%)	ES (63.36%)	NS (36.64%)	ES (41.14%)	NS (58.86%)
GNN-PPI		79.64	69.64	38.41	56.13	53.85	36.02	47.89
SemiGNN-PPI		**80.22**	**70.33**	**41.67**	**61.07**	**57.90**	**57.39**	**72.73**
	5	BS (38.16%)	ES (47.61%)	NS (14.23%)	ES (46.63%)	NS (53.37%)	ES (43.18%)	NS (56.82%)
GNN-PPI		53.43	44.33	40.64	53.85	49.62	56.10	51.95
SemiGNN-PPI		**59.76**	**57.82**	**42.71**	**58.25**	**56.25**	**58.18**	**58.60**

Table 3: Analysis on performance between GNN-PPI and SemiGNN-PPI over BS, ES, and NS subsets in the SHS27k dataset. The ratios of the subsets are annotated in brackets. The BS subsets are empty under DFS and BFS partitions and are omitted for brevity.

Baseline Methods. We compare SemiGNN-PPI with several representative methods in PPI prediction, including: **Machine Learning (ML)** methods include RF [Wong *et al.*, 2015] and LR [Silberberg *et al.*, 2014], which take commonly handcrafted protein features including AC [Guo *et al.*, 2008] and CTD [Du *et al.*, 2017] as inputs. **Deep Learning (DL)** approaches include DNN-PPI [Li *et al.*, 2018], PIPR [Chen *et al.*, 2019a], and GNN-PPI [Lv *et al.*, 2021], which take amino acid sequence-based features as inputs (More details are illustrated in the Supplementary Material). It is noted that GNN-PPI adopts graph learning to leverage protein correlations, achieving state-of-the-art performance on multi-type PPI prediction. In this regard, we extensively compare our method with GNN-PPI in different scenarios and settings.

4.3 Results and Analysis

Benchmark Analysis. In Table 1, we compare our methods with other baseline methods under different partition schemes and various datasets. It is observed that graph-based methods, *i.e.*, GNN-PPI and SemiGNN-PPI outperform other ML and DL methods, even under more challenging BFS and DFS partitions with more unseen proteins. It can be attributed to graph learning, which can better capture correlations between proteins despite the existence of more unknown proteins. Furthermore, our method incorporates multiple graphs (M-Graph) for feature learning, achieving state-of-the-art performance in multi-type PPI prediction. Especially, under challenging evaluations with small datasets, *e.g.*, SHS27k-DFS, our method achieves much higher F1 scores than GNN-PPI, since self-ensemble graph learning can effectively improve the model robustness against complex scenarios. Moreover, the number of parameters is 1.09M (GNN-PPI) and 1.13M (ours), and the inference time on SHS27k is 0.050s (GNN-PPI) and 0.058s (ours). GNN-PPI and our method have comparable performance in the two metrics, showing the scalability of the proposed method.

Label Efficiency. To demonstrate the feasibility of our method under the label scarcity scenario, we present experimental results under different label ratios in Table 2. We can see that GNN-PPI receives severe performance degradation with fewer labels. In comparison, our method achieves better performance under all scenarios with different datasets, label ratios, and partition schemes. Remarkably, our method under some scenarios, *e.g.*, SHS148k-BFS-20% can achieve comparable performance with GNN-PPI using 100% labeled data, indicating the annotation efficiency of our method. To further analyze the model performance on inter-novel-protein interaction prediction, we make an in-depth performance comparison between GNN-PPI and SemiGNN-PPI in different subsets (BS/ES/NS) of the testset. As shown in Table 3, the BS subset comprises most of the whole testset under the random partition, which cannot reflect the prediction performance on the inter-novel-protein interactions. In contrast, the proportions of ES and NS subsets increase under label scarcity and other partition schemes; in these settings, SemiGNN-PPI consistently outperforms GNN-PPI in both ES and NS subsets by a large margin, which demonstrates the effectiveness of SemiGNN-PPI for inter-novel-protein interaction prediction.

Performance on Different PPI Types. To study the per-class prediction performance, we present the model performance on different PPI types with corresponding type ratios in Table 4. It is observed that the PPI types are unbalanced with some under-represented types, such as Ptmod, Inhibition, and Expression. Nevertheless, SemiGNN-PPI outper-

PPI Type	Type Ratio	Random Partition		DFS Partition		BFS Partition	
		GNN-PPI	SemiGNN-PPI	GNN-PPI	SemiGNN-PPI	GNN-PPI	SemiGNN-PPI
Reaction	40.61%	$89.58_{0.15}$	$\mathbf{90.16_{0.43}}$	$81.90_{1.65}$	$\mathbf{85.86_{0.71}}$	$61.62_{1.29}$	$\mathbf{64.92_{5.73}}$
Binding	52.71%	$88.28_{0.48}$	$\mathbf{89.46_{0.57}}$	$83.52_{1.41}$	$\mathbf{86.39_{0.67}}$	$70.00_{4.10}$	$\mathbf{72.43_{6.33}}$
Ptmod	20.99%	$87.04_{0.29}$	$\mathbf{87.42_{0.33}}$	$77.94_{1.67}$	$\mathbf{82.99_{1.44}}$	$65.92_{5.52}$	$\mathbf{71.32_{5.04}}$
Activation	42.51%	$85.15_{0.38}$	$\mathbf{85.26_{0.46}}$	$73.48_{2.74}$	$\mathbf{77.95_{1.19}}$	$67.44_{8.43}$	$\mathbf{68.04_{8.06}}$
Inhibition	20.20%	$87.21_{0.18}$	$\mathbf{88.09_{0.31}}$	$72.46_{1.11}$	$\mathbf{78.12_{2.62}}$	$60.20_{4.62}$	$\mathbf{67.71_{7.21}}$
Catalysis	44.67%	$89.36_{0.44}$	$\mathbf{90.35_{0.31}}$	$82.30_{0.80}$	$\mathbf{85.77_{1.29}}$	$65.70_{4.42}$	$\mathbf{73.39_{6.33}}$
Expression	7.69%	$\mathbf{47.85_{0.79}}$	$46.99_{0.22}$	$\mathbf{34.96_{3.74}}$	$32.45_{5.96}$	$\mathbf{31.81_{6.87}}$	$28.99_{4.90}$
Macro-Average	-	$82.07_{0.39}$	$\mathbf{82.53_{0.38}}$	$72.37_{1.87}$	$\mathbf{74.16_{2.09}}$	$60.38_{5.03}$	$\mathbf{63.29_{5.29}}$
Micro-Average	-	$86.67_{0.22}$	$\mathbf{87.38_{0.24}}$	$80.96_{1.61}$	$\mathbf{83.63_{0.86}}$	$67.10_{3.48}$	$\mathbf{71.06_{3.35}}$

Table 4: Per-class results in the SHS148k dataset with 20% training labels. The type ratios are calculated over the whole dataset.

forms GNN-PPI on most PPI types, especially for relatively imbalanced types (82.99 vs. 77.94 in Ptmod-DFS and 67.71 vs. 60.20 in Inhibition-BFS). It is noted that lower performance is achieved with our method in the type Expression, which could be due to inaccurate label correlations captured with extremely low co-occurrence with other labels, which is still a direction to explore in our future work.

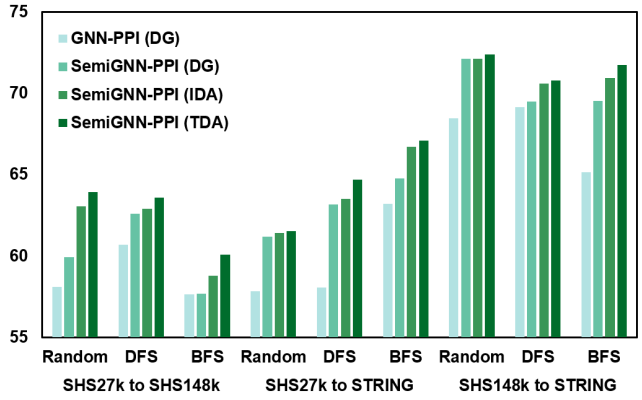

Figure 2: Performance comparison on trainset-heterologous testsets. DG: domain generalization. IDA: inductive domain adaptation. TDA: transductive domain adaptation.

Model Generalization. To access the generalization capability of the proposed method, we test the model trained using small datasets, *e.g.,* SHS27k on big datasets, *e.g.,* STRING in three evaluation settings: 1) *Domain Generalization (DG)*: The model is directly tested on the unseen dataset. 2) *Inductive Domain Adaptation (IDA)*: The model has access to unlabeled training data in the trainset-heterologous dataset during training. 3) *Transductive domain adaptation*: The model has access to the whole unlabeled trainset-heterologous dataset during training. In Fig. 2, we can observe that our method outperforms GNN-PPI in all partition schemes when tested on unseen datasets. Moreover, our model can effectively leverage unlabeled data, achieving better adaptation performance in both inductive and transductive setups.

Ablation Study. We investigate the effectiveness of different components in SemiGNN-PPI in Fig. 3. We can see that all components, *i.e.*, label graph encoding (LGE), self-ensemble (SE), and graph consistency constraint (GCC) positively contribute to the performance improvements. It is noted that too few labels *e.g.,* 10% may influence model initialization, limiting self-ensemble graph learning, while the performance gains are more evenly distributed among various components with 10% or more labeled data. Particularly, the proposed GCC can further enhance the results from the self-ensemble by providing stronger regularization in the feature space. Moreover, we have performed one experiment for each augmentation strategy (F1-score) on SHS27k (20% labeled) under random partition, *i.e.*, random edge dropout (78.92), random node dropout (79.04), centrality-based [Tang *et al.*, 2015] node and edge manipulation (81.08), and ours (81.11), which show that our strategy is comparable with centrality-based manipulation and outperforms others.

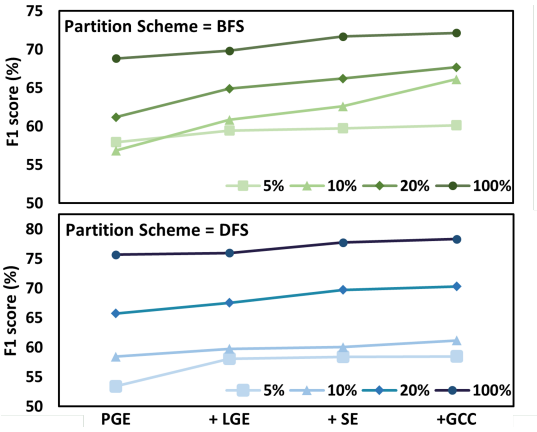

Figure 3: Results of ablation studies on different components of SemiGNN-PPI using the SHS27k dataset.

5 Conclusion

In this paper, we propose a novel self-ensembling multi-graph neural network (SemiGNN-PPI) for efficient and generalizable multi-type PPI prediction, which models both protein correlations and label dependencies by constructing and processing graphs at protein and label levels. To leverage unlabeled PPI data, we integrate GNN into Mean Teacher for self-ensemble graph learning, in which multiple graph consistency constraints are designed to align the teacher and student graphs in the feature embedding space for optimized consistency regularization. Extensive experiments have demonstrated the superiority in model performance, label efficiency and generalization ability of SemiGNN-PPI over state-of-the-art methods by large margins.

Acknowledgements

The first two authors contributed equally to this work. This research was funded by Competitive Research Programme "NRF-CRP22-2019-0003", National Research Foundation Singapore, and partially supported by A*STAR core funding.

References

[Acuner Ozbabacan *et al.*, 2011] Saliha Ece Acuner Ozbabacan, Hatice Billur Engin, Attila Gursoy, and Ozlem Keskin. Transient protein–protein interactions. *Protein engineering, design and selection*, 24(9):635–648, 2011.

[Bekker and Davis, 2020] Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: A survey. *Machine Learning*, 2020.

[Browne *et al.*, 2007] Fiona Browne, Haiying Wang, Huiru Zheng, and Francisco Azuaje. Supervised statistical and machine learning approaches to inferring pairwise and module-based protein interaction networks. In *IEEE International Symposium on BioInformatics and BioEngineering*, 2007.

[Chen and Liu, 2005] Xue-Wen Chen and Mei Liu. Prediction of protein-protein interactions using random decision forest framework. *Bioinformatics*, 2005.

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

[Chen *et al.*, 2019b] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5177–5186, 2019.

[Choi *et al.*, 2019] Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6830–6840, 2019.

[Du *et al.*, 2017] Xiuquan Du, Shiwei Sun, Changlin Hu, Yu Yao, Yuanting Yan, and Yanping Zhang. Deepppi: boosting prediction of protein–protein interactions with deep neural networks. *Journal of chemical information and modeling*, 57(6):1499–1510, 2017.

[Fields and Song, 1989] Stanley Fields and Ok-kyu Song. A novel genetic system to detect protein–protein interactions. *Nature*, 340(6230):245–246, 1989.

[Guo *et al.*, 2008] Yanzhi Guo, Lezheng Yu, Zhining Wen, and Menglong Li. Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic acids research*, 2008.

[Guo *et al.*, 2019] Chuan Guo, Ali Mousavi, Xiang Wu, Daniel N Holtmann-Rice, Satyen Kale, Sashank Reddi, and Sanjiv Kumar. Breaking the glass ceiling for embedding-based classifiers for large output spaces. *Advances in Neural Information Processing Systems*, 2019.

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

[Ho *et al.*, 2002] Yuen Ho, Albrecht Gruhler, Adrian Heilbut, Gary D Bader, Lynda Moore, Sally-Lin Adams, Anna Millar, Paul Taylor, Keiryn Bennett, Kelly Boutilier, et al. Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature*, 415(6868):180–183, 2002.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2016.

[Laine and Aila, 2017] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017.

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

[Lin and Chen, 2013] Xiaotong Lin and Xue-wen Chen. Heterogeneous data integration by tree-augmented naïve b ayes for protein-protein interactions prediction. *Proteomics*, 13(2):261–268, 2013.

[Liu *et al.*, 2021] Weiwei Liu, Haobo Wang, Xiaobo Shen, and Ivor W Tsang. The emerging trends of multi-label learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7955–7974, 2021.

[Lu *et al.*, 2022] Yingzhou Lu, Chiung-Ting Wu, Sarah J Parker, Zuolin Cheng, Georgia Saylor, Jennifer E Van Eyk, Guoqiang Yu, Robert Clarke, David M Herrington, and Yue Wang. Cot: an efficient and accurate method for detecting marker genes among many subtypes. *Bioinformatics Advances*, 2022.

[Luo *et al.*, 2015] Xin Luo, Zhuhong You, Mengchu Zhou, Shuai Li, Hareton Leung, Yunni Xia, and Qingsheng Zhu. A highly efficient approach to protein interactome mapping based on collaborative filtering framework. *Scientific reports*, 5(1):1–10, 2015.

[Lv *et al.*, 2021] Guofeng Lv, Zhiqiang Hu, Yanguang Bi, and Shaoting Zhang. Learning unknown from correlations: Graph neural network for inter-novel-protein interaction prediction. In *IJCAI International joint conference on artificial intelligence*, 2021.

[Ma *et al.*, 2022] Yuchen Ma, Yanbei Chen, and Zeynep Akata. Distilling knowledge from self-supervised teacher by embedding graph alignment. In *33rd British Machine Vision Conference*. BMVA Press, 2022.

[Margolin *et al.*, 2006] Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. Aracne: an al-

gorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. In *BMC bioinformatics*, volume 7, pages 1–15. Springer, 2006.

[Min *et al.*, 2017] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. Deep learning in bioinformatics. *Briefings in bioinformatics*, 18(5):851–869, 2017.

[Petta *et al.*, 2016] Ioanna Petta, Sam Lievens, Claude Libert, Jan Tavernier, and Karolien De Bosscher. Modulation of protein–protein interactions for the development of novel therapeutics. *Molecular Therapy*, 2016.

[Pio *et al.*, 2020] Gianvito Pio, Michelangelo Ceci, Francesca Prisciandaro, and Donato Malerba. Exploiting causality in gene network reconstruction based on graph embedding. *Machine Learning*, 2020.

[Pio *et al.*, 2022] Gianvito Pio, Paolo Mignone, Giuseppe Magazzù, Guido Zampieri, Michelangelo Ceci, and Claudio Angione. Integrating genome-scale metabolic modelling and transfer learning for human gene regulatory network reconstruction. *Bioinformatics*, 2022.

[Qu and Hickey, 2022] Xiaodong Qu and Timothy J. Hickey. Eeg4home: A human-in-the-loop machine learning model for eeg-based bci. In *Augmented Cognition*, 2022.

[Silberberg *et al.*, 2014] Yael Silberberg, Martin Kupiec, and Roded Sharan. A method for predicting protein-protein interaction types. *PLoS One*, 9(3):e90904, 2014.

[Skrabanek *et al.*, 2008] Lucy Skrabanek, Harpreet K Saini, Gary D Bader, and Anton J Enright. Computational prediction of protein–protein interactions. *Molecular biotechnology*, 38(1):1–17, 2008.

[Soleymani *et al.*, 2022] Farzan Soleymani, Eric Paquet, Herna Viktor, Wojtek Michalowski, and Davide Spinello. Protein–protein interaction prediction with deep learning: A comprehensive review. *Computational and Structural Biotechnology Journal*, 2022.

[Sun *et al.*, 2017] Tanlin Sun, Bo Zhou, Luhua Lai, and Jianfeng Pei. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC bioinformatics*, 18(1):1–8, 2017.

[Szklarczyk *et al.*, 2019] Damian Szklarczyk, Annika L Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T Doncheva, John H Morris, Peer Bork, et al. String v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic acids research*, 2019.

[Tajbakhsh *et al.*, 2020] Nima Tajbakhsh, Laura Jeyaseelan, Qian Li, Jeffrey N Chiang, Zhihao Wu, and Xiaowei Ding. Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical Image Analysis*, 63:101693, 2020.

[Tang *et al.*, 2015] Yu Tang, Min Li, Jianxin Wang, Yi Pan, and Fang-Xiang Wu. Cytonca: a cytoscape plugin for centrality analysis and evaluation of protein interaction networks. *Biosystems*, 127:67–72, 2015.

[Tarvainen and Valpola, 2017] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 2017.

[Wang *et al.*, 2016] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294, 2016.

[Wang *et al.*, 2020] Ya Wang, Dongliang He, Fu Li, Xiang Long, Zhichao Zhou, Jinwen Ma, and Shilei Wen. Multi-label classification with label graph superimposing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12265–12272, 2020.

[Wong *et al.*, 2015] Leon Wong, Zhu-Hong You, Shuai Li, Yu-An Huang, and Gang Liu. Detection of protein-protein interactions from amino acid sequences using a rotation forest model with a novel pr-lpq descriptor. In *International Conference on Intelligent Computing*, 2015.

[Xu *et al.*, 2019] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[Xu *et al.*, 2022] Kaixin Xu, Liyang Liu, Ziyuan Zhao, Zeng Zeng, and Veeravalli Bharadwaj. Object-aware self-supervised multi-label learning. In *IEEE International Conference on Image Processing*, 2022.

[Yang *et al.*, 2020] Fang Yang, Kunjie Fan, Dandan Song, and Huakang Lin. Graph-based prediction of protein-protein interactions with attributed signed graph embedding. *BMC bioinformatics*, 21(1):1–16, 2020.

[You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33:5812–5823, 2020.

[Zhang *et al.*, 2019] Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. Biowordvec, improving biomedical word embeddings with subword information and mesh. *Scientific data*, 6(1):1–9, 2019.

[Zhang *et al.*, 2021] Yabin Zhang, Haojian Zhang, Bin Deng, Shuai Li, Kui Jia, and Lei Zhang. Semi-supervised models are strong unsupervised domain adaptation learners. *arXiv preprint arXiv:2106.00417*, 2021.

[Zhao *et al.*, 2021] Ziyuan Zhao, Kaixin Xu, Shumeng Li, Zeng Zeng, and Cuntai Guan. Mt-uda: Towards unsupervised cross-modality medical image segmentation with limited source labels. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 293–303. Springer, 2021.

[Zhao *et al.*, 2022] Ziyuan Zhao, Fangcheng Zhou, Kaixin Xu, Zeng Zeng, Cuntai Guan, and S Kevin Zhou. Le-uda: Label-efficient unsupervised domain adaptation for medical image segmentation. *IEEE Transactions on Medical Imaging*, 2022.