

CSGNN: Contrastive Self-Supervised Graph Neural Network for Molecular Interaction Prediction

Chengshuai Zhao, Shuai Liu, Feng Huang, Shichao Liu and Wen Zhang*

College of Informatics, Huazhong Agricultural University, Wuhan, China

{zhaochengshuai, shuailiu, fhuang233}@webmail.hzau.edu.cn,
{scliu, zhangwen}@mail.hzau.edu.cn

Abstract

Molecular interactions are significant resources for analyzing sophisticated biological systems. Identification of multifarious molecular interactions attracts increasing attention in biomedicine, bioinformatics, and human healthcare communities. Recently, a plethora of methods have been proposed to reveal molecular interactions in one specific domain. However, existing methods heavily rely on features or structures involving molecules, which limits the capacity of transferring the models to other tasks. Therefore, generalized models for the multifarious molecular interaction prediction (MIP) are in demand. In this paper, we propose a contrastive self-supervised graph neural network (CSGNN) to predict molecular interactions. CSGNN injects a mix-hop neighborhood aggregator into a graph neural network (GNN) to capture high-order dependency in the molecular interaction networks and leverages a contrastive self-supervised learning task as a regularizer within a multi-task learning paradigm to enhance the generalization ability. Experiments on seven molecular interaction networks show that CSGNN outperforms classic and state-of-the-art models. Comprehensive experiments indicate that the mix-hop aggregator and the self-supervised regularizer can effectively facilitate the link inference in multifarious molecular networks.

1 Introduction

There are various molecular interactions/associations in biological systems, such as drug-target interactions [Cheng *et al.*, 2012], drug-drug interactions [Zhang *et al.*, 2017], protein-protein interactions [Kovács *et al.*, 2019], drug-disease associations [Yu *et al.*, 2020], and miRNA-disease associations [Huang *et al.*, 2020a], and they implicate significant information for sophisticated biological system analysis [Huang *et al.*, 2018]. Although many interactions have been revealed by the wet experiments or clinic reports, a portion of interactions remains undiscovered. However, wet experimental

methods for identifying molecular interactions are extremely costly and time-consuming. The computational methods, in contrast, can efficiently discover potential interactions and guide *in vivo* validation, which provides an economic alternative.

Since molecular interaction discovery is a prerequisite for the downstream analysis (e.g., disease diagnosis and drug discovery), a great number of computational methods have been proposed to reveal molecular interactions in one specific domain (e.g., drug-drug interactions and drug-target interactions). Existing methods heavily depend on features and structures involving the molecules in one specific task, which limits the generalization capacity of transferring the models to other tasks. Thus, the generalized models for multifarious molecular interaction prediction are in demand.

Molecular interactions can be naturally formulated as networks where molecules and interactions are regarded as nodes and edges respectively. Then, network-based algorithms such as random walk and label propagation can be applied for molecular interaction prediction (MIP). Recently, network/graph-based deep learning techniques, e.g., graph neural networks (GNNs) [Wu *et al.*, 2021], have emerged as a dominant class of methods which advance network analysis. In the light of GNNs, many tasks, such as node classification, graph classification, and link prediction, have been successfully handled. In GNNs, one of the most popular categories is graph convolutional networks (GCNs) [Kipf and Welling, 2017] which learn node representation by neighborhood aggregation and feature transformation. GCN provides a flexible and powerful frame for representation learning, which motivates us to resolve molecular interaction prediction problems.

A desired molecular interaction prediction method should have transferable, generalized, and robust performances, whereas several bottlenecks have restricted it. On the one hand, in molecular interaction networks, the degree distribution of nodes links are often nonuniform and sparse [Liu *et al.*, 2020], which challenges many methods. Additionally, in many GNN frameworks, due to the immediate neighborhood aggregator, node representations learned by stacking multiple layers are over smoothed and redundant, which undermines the effect of the indirect nodes. On the other hand, the scarcity of annotated data restricts the predictive capacity of models. Typically, we can only acquire the few molecu-

*Contact Author

lar interactions and features involving the entities, but these specific features are usually not transferable. Therefore, the generalized methods are expected to make prediction merely based on limited interactions.

In this paper, we propose a contrastive self-supervised graph neural network (CSGNN) for molecular interaction prediction (MIP) while tackling the above challenges. CSGNN leverages a mix-hop [Abu-El-Haija *et al.*, 2019] aggregator in the deep graph neural network framework aiming to model and capture the dependent relation between indirect neighbors. Further, a contrastive self-supervised graph neural network, which maximizes mutual information between local patches and the network-level context, is elaborated to assist CSGNN with effective representation learning from scarce data. The contributions can be summarized as:

- The proposed CSGNN can accurately predict interactions between multifarious molecular entities, merely based on known interactions, not depending on any features or structures of entities.
- In molecular interaction networks, we consider the importance of high-order dependency and design a deep mix-hop graph neural network to model the dependency, which is proved to be effective in ablation study.
- We introduce a contrastive self-supervised learning task as a regularizer of molecular interaction prediction within a multi-task learning paradigm, and develop a delicate framework CSGNN, which provides a novel insight for molecular representation learning.

2 Related Works

In molecular interaction networks, the goal of molecular interaction prediction is to predict if a given pair of biomedical entities such as proteins, drugs, or diseases will interact. There are three types of network-based methods: (1) **Similarity-based learning** assumes that there likely exists a link if a pair of non-connected nodes are similar or close in the networks [Liu *et al.*, 2020]. (2) **Matrix completion** transforms the interaction prediction as a matrix completion problem [Zhang *et al.*, 2020b]. (3) **Graph representation learning** acquires representations of entities from networks for downstream link prediction [Zhang *et al.*, 2020a]. Although these pioneers have produced good performances, a neglected deficiency is that most of them are designed for the specific molecular interaction tasks and cannot be transferred to other tasks.

Recently, the application of graph neural networks in multifarious molecular interaction prediction tasks has received increasing attention. For example, MR-GNN [Xu *et al.*, 2019b] utilizes a dual graph neural network to predict structured entity interactions. SkipGNN [Huang *et al.*, 2020b] infers interactions between molecules via a skip graph neural network. Generally, like most deep learning based methods, GNNs require massive amounts of labeled data for model training. But in practice, only few known interactions/associations in the molecular interaction networks are available. Therefore, how to learn powerful representations

from networks with limited data is crucial for the development of high-accuracy predictive models.

A number of graph-specific self-supervised learning pre-text tasks have been designed to learn powerful representations for nodes or graphs. Contrastive learning, an important category of self-supervised methods, which aims to discriminate relative instance pairs and negative-sampled pairs through implicitly serving the intra-pair proximity as supervisory signals, deserves much attention. For examples, a few classic graph embedding methods, such as LINE [Tang *et al.*, 2015], struc2vec [Ribeiro *et al.*, 2017] and GraphSage [Hamilton *et al.*, 2017], define different kinds of node proximity; GCC [Qiu *et al.*, 2020] aims to distinguish between similar subgraph pairs and other dissimilar ones; DGI [Velickovic *et al.*, 2019] seeks to classify node-graph pairs and ‘fake’ counterparts; the method which Hassani and Ahmadi [2020] extend from DGI introduces graph data augmentation to expand contrastive samples; InfoGraph [Sun *et al.*, 2020a] and GraphCL [You *et al.*, 2020] use a similar contrastive learning framework to learn graph-level representations. Another well-designed self-supervised method is explicit in the paradigm of supervised settings (classification or regression) with the targets automatically generated from the data itself. For examples, M3S [Sun *et al.*, 2020b] utilizes a clustering algorithm to yield pseudo labels in a self-training framework for semi-supervised node classification; GPT-GNN [Hu *et al.*, 2020] forces GNN to predict masked nodes and edges; S2GRL [Peng *et al.*, 2020] takes the minimized hop counts between two nodes as the supervisory signals. In this paper, we extend DGI and formulate it as a self-supervised auxiliary task to enhance the performance of link prediction task.

3 The Proposed CSGNN

3.1 Problem Formulation

A molecular interaction network can be viewed as an undirected attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where \mathcal{V} represents the set of nodes that correspond to molecular entities (e.g., drugs, diseases or proteins) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges indicating the existence of interaction between two entities in \mathcal{V} . $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times m}$ is a feature matrix where each node in \mathcal{V} is encoded as a predefined m -dimensionality attribute vector (e.g., a generated graph embedding or a one-hot coding). Specifically, the graph can be further represented as an adjacency matrix $\mathbf{A} \in \{0, 1\}^{|\mathcal{V}| \times |\mathcal{V}|}$ where $\mathbf{A}_{u,v} = 1$ if there exists a link from u to v in the graph (i.e., $(u, v) \in \mathcal{E}$) and $\mathbf{A}_{u,v} = 0$ otherwise. Note that \mathcal{G} is a undirected graph, hence, \mathbf{A} is symmetric. Our goal of molecular interaction prediction is to learn a mapping function $\Theta(\omega) : \mathcal{E} \rightarrow [0, 1]$ from edges to scores, where ω is parameter, such that we can obtain the probability that two arbitrary nodes interact each other [Huang *et al.*, 2020b].

3.2 Overview

Figure 1 illustrates the overview of our proposed contrastive self-supervised graph neural network framework, which contains a deep mix-hop graph neural network undertaking the molecular interaction prediction task and a contrastive graph

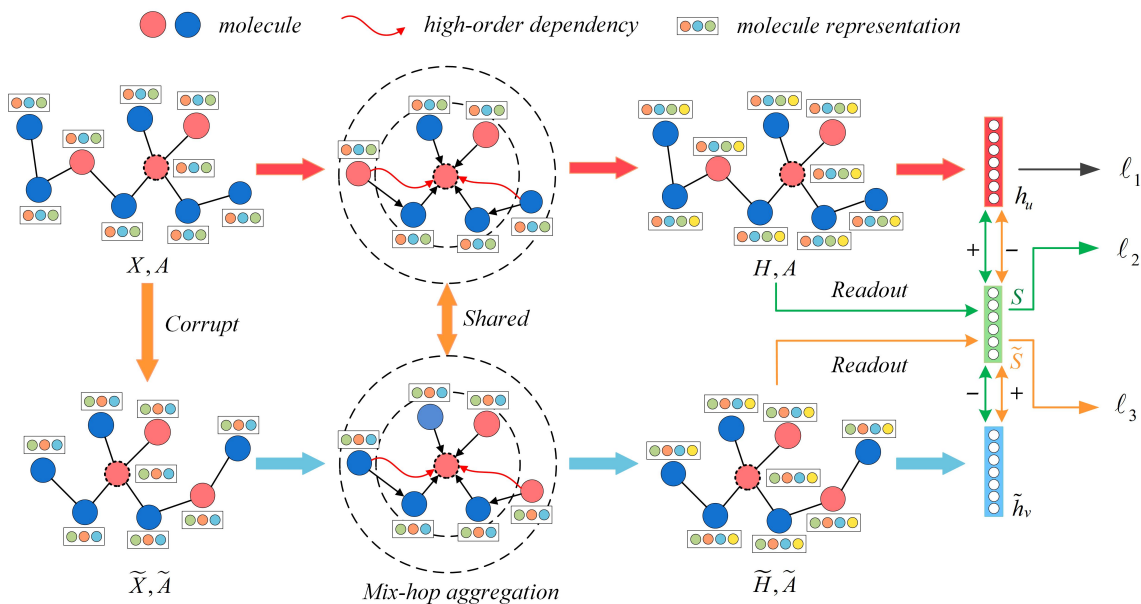


Figure 1: Overview of CSGNN.

neural network implementing the self-supervised learning task. The deep mix-hop graph neural network captures the indirect interactions in the molecular interaction networks; the contrastive graph neural network makes use of the information derived from data itself to enhance the generalization ability of the predictor. Two tasks, incorporated into an objective function, are jointly trained.

In the following, we will describe technical details of the deep mix-hop graph neural network and the contrastive graph neural network.

3.3 Deep Mix-Hop Graph Neural Network

Graph neural networks (GNNs) use the graph structure and node attribute to learn the high-level compact representations of nodes, denoted by $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times d}$ where $\mathbf{h}_v = \mathbf{H}[v, :]$ is the latent representation of the node v . In general, GNNs iteratively update the representation of a node by aggregating representations of its neighbors. After l iterations, the hidden state of node v can be denoted as: $\mathbf{h}_v^{(l)} = \phi^{(l)}(\mathbf{h}_v^{(l-1)}, f^{(l)}(\{\mathbf{h}_u^{(l-1)} : u \in \mathcal{N}(v)\}))$, where $\phi^{(l)}(\cdot)$ is a combination function (e.g., summation, element-wise product, concatenation operation or a neural network) of l -th iterations. $\mathcal{N}(v)$ denotes a set of nodes adjacent to v . Specifically, we initialize $\mathbf{h}_v^{(0)} = \mathbf{X}[v, :]$.

To model the high-order proximity in molecular interaction networks, we design a deep mix-hop graph neural network, which can highlight the role played by indirect neighbors, as shown in Figure 1. The update rule of the hidden states of nodes in our model can be defined as:

$$\mathbf{h}_v^{(l)} = \phi^{(l)}(\mathbf{h}_v^{(l-1)}, \{f_k^{(l)}(\{\mathbf{h}_u^{(l-1)} : u \in \mathcal{N}_k(v)\})\}_{k=1}^K) \quad (1)$$

where $f_k^{(l)}(\cdot)$ is an aggregation function of l -th iteration on k -th hop, and $\mathcal{N}_k(v)$ denotes the set of k -hop neighbors of v .

Later, we decode the representations of node u and node v into the probability of their interaction: $\hat{p}_{uv} = \rho(\mathbf{h}_u, \mathbf{h}_v)$, $\rho(\cdot)$ is a scoring function (e.g., inner product, bilinear function or a neural network). Then the loss of supervised molecular interaction prediction task can be formulated as:

$$\ell_1 = -\frac{1}{|\mathcal{T}|} \sum_{(u,v) \in \mathcal{T}} p_{uv} \log \hat{p}_{uv} + (1 - p_{uv}) \log(1 - \hat{p}_{uv}) \quad (2)$$

where \mathcal{T} is the sampled training set of node pairs and p_{uv} denotes true label for the interaction between nodes u and v .

3.4 Contrastive Graph Neural Network

The proposed CSGNN leverages a self-supervised learning task which conducts contrastive learning between the original networks and corrupted networks by maximizing mutual information.

Following DGI [Velickovic *et al.*, 2019], the self-supervised auxiliary task maximizes mutual information between a local patch (i.e., node-level representation) and its global context (i.e., graph-level representation). One original intention of above design is guiding the node representations to capture the global information of the entire graph. We employ a readout function, $\Gamma : \mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times d} \rightarrow \mathbf{s} \in \mathbb{R}^d$ (i.e., $\mathbf{s} = \Gamma(\mathbf{H})$), to summarize the obtained local patches into the global context \mathbf{s} .

Then, we corrupt the original graph by an explicit (stochastic) corruption pipeline to convert the graph into a "fake" one:

$$\Pi : (\mathbf{X}, \mathbf{A}) \in (\mathbb{R}^{|\mathcal{V}| \times m}, \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}) \rightarrow (\tilde{\mathbf{X}}, \tilde{\mathbf{A}}) \in (\mathbb{R}^{|\mathcal{V}'| \times m}, \mathbb{R}^{|\mathcal{V}'| \times |\mathcal{V}'|}) \quad (3)$$

Next, we utilize a noise-contrastive type objective with a standard binary cross-entropy (BCE) loss between the positive pairs (each pair consists of global information and one

Datasets	Nodes	Edges	Density	Degree
ChG-Miner (DTI)	7,343	15,139	0.06%	4.12
ChCh-Mine (DDI)	1,514	48,514	4.23%	64.09
HuRI-PPI (PPI)	5,604	23,322	0.15%	8.32
DG-AssocMiner (DGI)	7,813	21,357	0.07%	5.47
DD-Miner (DIA)	6,878	6,877	0.03%	2.00
DCh-Miner (DDA)	7,199	466,657	1.80%	129.65
ChSe-Decagon (DSA)	10,823	174,997	0.30%	32.33

Table 1: The statistic of seven widely used datasets.

of local patches) and the negative ones (each pair consists of global information and one of corrupted patches) to model self-supervised leaning task. Naturally, we present the following loss function:

$$\ell_2 = - \frac{1}{|\mathcal{V}| + |\mathcal{V}'|} \left(\sum_{u \in \mathcal{V}} \mathbb{E}_{(\mathbf{X}, \mathbf{A})} [\log \Psi(\mathbf{h}_u, \mathbf{s})] + \sum_{v \in \mathcal{V}'} \mathbb{E}_{(\tilde{\mathbf{X}}, \tilde{\mathbf{A}})} [\log(1 - \Psi(\tilde{\mathbf{h}}_v, \mathbf{s}))] \right) \quad (4)$$

The contrastive learning mechanism maximizing mutual information between the original global context and local patches, to some extent, allows our model to be more "discriminative" for node representation learning. Further, we extend the contrastive learning mechanism from another view: maximizing mutual information between the corrupted global context and local patches, which can be viewed as the "dual problem" of above problem:

$$\ell_3 = - \frac{1}{|\mathcal{V}| + |\mathcal{V}'|} \left(\sum_{v \in \mathcal{V}'} \mathbb{E}_{(\tilde{\mathbf{X}}, \tilde{\mathbf{A}})} [\log \Psi(\tilde{\mathbf{h}}_v, \tilde{\mathbf{s}})] + \sum_{u \in \mathcal{V}} \mathbb{E}_{(\mathbf{X}, \mathbf{A})} [\log(1 - \Psi(\mathbf{h}_u, \tilde{\mathbf{s}}))] \right) \quad (5)$$

where $\tilde{\mathbf{s}}$ denotes the corrupted graph-level representation. This architecture, on the one hand, exploiting more information derived from data itself, enhances the generalization ability. On the other hand, it consider both views of the network (i.e., original and corrupted views) simultaneously, which enables the model more "balanced" and robust.

Intuitively, to train a model by the missing link task and the self-supervised leaning task, we optimize the following objective that combines Eq.2, Eq.4 and Eq.5:

$$\ell = \alpha \ell_1 + \beta \ell_2 + \gamma \ell_3 \quad (6)$$

where α, β, γ are hyperparameters which balance the contribution of different tasks. The pseudo-codes of CSGNN are illustrated in Algorithm 1.

4 Experiment

4.1 Datasets and Settings

We consider seven publicly available network datasets. (1) **ChG-Miner** consists of 5,018 drugs which target 2,325 proteins via 15,139 drug-target interactions (DTIs). (2) **ChCh-Miner** contains 48,514 interactions between 1,514 drugs

Algorithm 1 The Contrastive Self-Supervised Graph Neural Network Algorithm

Input: Molecular interaction network \mathcal{G} ; function $f(\cdot), \phi(\cdot), \rho(\cdot), \Pi(\cdot, \cdot), \Gamma(\cdot), \Psi(\cdot)$; parameters α, β, γ ; training set \mathcal{T}

Output: $\Theta(\omega)$

```

1: while CSGNN not converge do
2:    $\mathbf{H}^{(0)} = \mathbf{X}$ 
3:    $(\tilde{\mathbf{X}}, \tilde{\mathbf{A}}) \leftarrow \Pi(\mathbf{X}, \mathbf{A})$ 
4:    $\tilde{\mathbf{H}}^{(0)} = \tilde{\mathbf{X}}$ 
5:   for  $u \in \mathcal{V}$  do
6:     for  $l = 1, 2, \dots, L$  do
7:       for  $k = 1, 2, \dots, K$  do
8:         Calculate  $f_k^{(l)}(\{\mathbf{h}_v^{(l-1)} : v \in \mathcal{N}_k(u)\})$ 
9:       end for
10:      Update  $\mathbf{h}_u^{(l)}$  via Eq.(1)
11:     end for
12:   end for
13:    $\mathbf{s} \leftarrow \Gamma(\mathbf{H})$ 
14:   Calculate  $\tilde{\mathbf{H}}, \tilde{\mathbf{s}}$  like line 5-13
15:   for  $(u, v) \in \mathcal{T}$  do
16:     Calculate  $p_{uv} \log \hat{p}_{uv} + (1 - p_{uv}) \log(1 - \hat{p}_{uv})$ 
17:   end for
18:   Calculate  $\ell_1$  via Eq.(2)
19:   for  $u \in \mathcal{V}$  do
20:     Calculate  $\mathbb{E}_{(\mathbf{X}, \mathbf{A})} [\log \Psi(\mathbf{h}_u, \mathbf{s})]$ 
21:     Calculate  $\mathbb{E}_{(\mathbf{X}, \mathbf{A})} [\log(1 - \Psi(\mathbf{h}_u, \tilde{\mathbf{s}}))]$ 
22:   end for
23:   for  $v \in \mathcal{V}'$  calculate loss like line 19-22
24:   Calculate  $\ell_2$  and  $\ell_3$  via Eq. (4) and Eq. (5)
25:    $\nabla_{\omega} \alpha \ell_1 + \beta \ell_2 + \gamma \ell_3$ 
26: end while
27: return  $\Theta(\omega)$ 
    
```

(DDIs). (3) **HuRI-PPI** includes 23,322 interactions among 5,604 proteins (PPIs) in HI-III network. (4) **DG-AssocMiner** comprises 519 diseases which associate 7,294 genes through 2,1357 associations (DGIs). (5) **DD-Miner** is a dataset with 6,877 associations between 6,878 diseases (DIAs) (6) **DCh-Miner** contains 5,536 diseases associate 1,663 drugs via 466,657 associations (DDAs). (7) **ChSe-Decagon** includes 639 drugs, 10,184 side-effects and 17,499 associations (DSAs) among them. We download DTIs, DDIs, DGIs, DIAs, DDAs, and DSAs from BioSNAP [Marinka Zitnik and Leskovec, 2018], and collect PPIs from CCSB [Luck *et al.*, 2020]. The detailed information is listed in Table 1.

Baselines. We compare CSGNN with several state-of-the-art methods, which can be categorized as:

- **Matrix Factorization (MF)** decomposes the input data matrix to acquire low-dimension representations. Two representative MF-based methods are selected, including GraRep [Cao *et al.*, 2015] and HOPE [Ou *et al.*, 2016].
- **Random Walk (RW)** utilizes random walk strategy on graphs to generate several node sequences. One most representative RW-based method Struc2Vec [Ribeiro *et al.*

Metrics	Data	MF		RW	NN		GNN				CSGNN		
		HOPE	GraRep	Struc2vec	LINE	SDNE	GCN	GAT	GraphSage	GIN	SkipGNN	-GCN	-GIN
AUC	DTI	0.8064	0.7359	0.7706	0.6599	0.7637	0.9233	0.6512	0.7243	0.9082	0.9096	0.9258	0.9137
	DDI	0.6425	0.6789	0.7615	0.6894	0.7526	0.8569	0.5963	0.8114	0.8793	0.8824	0.9350	0.9590
	PPI	0.6301	0.6635	0.7980	0.7111	0.7500	0.9051	0.7418	0.7980	0.9173	0.9082	0.9214	0.9233
	DGI	0.944	0.9168	0.8907	0.8281	0.8948	0.9918	0.6694	0.9466	0.9917	0.9899	0.9919	0.9917
	DIA	0.4381	0.4588	0.6337	0.4568	0.7375	0.8330	0.5047	0.4219	0.8696	0.7223	0.8844	0.8707
	DDA	0.8958	0.8226	0.8437	0.8509	0.9695	0.9890	0.5000	0.9851	0.9938	0.9941	0.9914	0.9944
	DSA	0.9311	0.8666	0.8997	0.8841	0.9609	0.9918	0.5000	0.9785	0.9925	0.9902	0.9925	0.9926
AUPR	DTI	0.7690	0.6517	0.6291	0.3029	0.4764	0.9253	0.7070	0.7686	0.9165	0.9050	0.9307	0.9237
	DDI	0.2501	0.3695	0.6236	0.3650	0.5374	0.8386	0.5348	0.7520	0.8526	0.8594	0.9210	0.9523
	PPI	0.1983	0.3157	0.6476	0.3391	0.5277	0.9091	0.7338	0.7843	0.9206	0.9136	0.9256	0.9313
	DGI	0.9596	0.9315	0.8290	0.6937	0.8141	0.9885	0.6509	0.9409	0.9890	0.9849	0.9896	0.9895
	DIA	0.1458	0.1437	0.4096	0.1167	0.4098	0.8453	0.5109	0.4375	0.8700	0.7242	0.8730	0.8618
	DDA	0.9068	0.8168	0.7700	0.7857	0.9399	0.9878	0.5000	0.9769	0.9929	0.9931	0.9903	0.9937
	DSA	0.9640	0.9196	0.8903	0.8111	0.9332	0.9878	0.5000	0.9625	0.9894	0.9849	0.9893	0.9899

Table 2: Performances of CSGNN and baselines

al., 2017] is chosen as baseline.

- **Neural Network (NN)** captures hidden patterns underlying the network structure from input data to learn graph embeddings. Two most representative NN-based methods LINE [Tang *et al.*, 2015] and SDNE [Wang *et al.*, 2016] are chosen.
- **Graph Neural Network (GNN)** considers both topological information and node and/or edge information in graph. Several most representative GNN-based methods including GCN [Kipf and Welling, 2017], GAT [Velickovic *et al.*, 2018], GraphSage [Hamilton *et al.*, 2017], GIN [Xu *et al.*, 2019a] and SkipGNN [Huang *et al.*, 2020b] are selected.

Metrics. In all experiments, each dataset is split into training, validation, and test sets as the ratio 7:1:2. Then, we train models on the training sets, select models with the best performance on the validation sets, and evaluate them on the test sets. The area under the receiver operating characteristic curve (AUC) and the area under the Precision-Recall Curve (AUPR) are elected as two representative indicators. Each experiment is conducted ten runs with random splits of dataset and the average metric is computed to avoid bias.

Experimental settings. In the deep mix-hop graph neural network, we employ GCN [Kipf and Welling, 2017] and GIN [Xu *et al.*, 2019a] as message passing frameworks, and concatenate the representation from each hop, therefore, the node representation can be formulated by:

$$\mathbf{h}_v^{(l)} = \parallel \sigma_k^{(l)} \left(\sum_{u \in \mathcal{N}_k(v) \cup \{v\}} \frac{1}{\sqrt{\hat{d}_v \hat{d}_u}} \cdot \mathbf{w}_k^{(l)} \cdot \mathbf{h}_u^{(l-1)} \right) \quad (7)$$

$$\mathbf{h}_v^{(l)} = \parallel MLP_k^{(l)} \left((1 + \epsilon_k^{(l)}) \cdot \mathbf{h}_v^{(l-1)} + \sum_{u \in \mathcal{N}_k(v)} \mathbf{h}_u^{(l-1)} \right) \quad (8)$$

where \parallel is a concatenation operation, σ is a non-linear activation function, $\hat{d}_v = 1 + |\mathcal{N}_k(v)|$, $\mathbf{w}_k^{(l)}$ is a weight matrix, and $\epsilon_k^{(l)}$ is a learnable parameter. Note that GIN can model an injective function mapping initial matrix to node representations, which allows CSGNN to powerfully capture the distinct topological structure of the graph. In this paper, we set

mix-hop $K = 2$ and we use the random distribution ranging from 0 to 1 to initiate the feature matrix \mathbf{X} on 128 dimensions.

We employ MLP with one hidden layer as the scoring function $\rho(\cdot)$. Concretely, the interaction prediction score can be defined as:

$$\hat{p}_{uv} = MLP(\parallel(\mathbf{h}_u + \mathbf{h}_v, \mathbf{h}_u \odot \mathbf{h}_v, \mathbf{h}_u, \mathbf{h}_v)) \quad (9)$$

where \odot is element-wise product. Note that, in above equation, multiple operations are performed to model multifarious relationships between two entities.

In the contrastive graph neural network, empirically, we randomly corrupt permutation of initial feature matrix \mathbf{X} to $\tilde{\mathbf{X}}$ via corruption function $\Pi(\cdot, \cdot)$. Then, we select Mean as the readout function $\Gamma(\cdot)$, which is experimentally efficient for large-scale data. Therefore, the graph-level representation can be denoted by: $s = Mean(\mathbf{H})$. Further, we instantiate the contrastive discriminator $\Psi(\cdot, \cdot)$ as $\sigma(\mathbf{h}^T \mathbf{W} \mathbf{s})$. Here, we use *sigmoid* as the activation function to produce the score that represents probabilities of being a positive sample. Note that the graph neural networks which encode the original and corrupted network share the same parameters.

In the joint training, we set $\alpha = 1$, $\beta = 0.1$ and $\gamma = 0.1$.

We run CSGNN and other compared methods on our workstation with 2 Intel(R) Xeon(R) Gold 6146 3.20GHZ CPUs, 128GB RAM, and 2 NVIDIA 1080 Ti GPUs. For more detailed parameter settings, please refer to the source code¹.

4.2 Results and Analysis

In this section, we conduct experiments to compare CSGNN with all the baselines, where GCN and GIN are respectively used as GNN encoders for CSGNN. The AUC and AUPR scores of all models are shown in Table 2, and we have the following observations from the results. Firstly, CSGNN outperforms all the baselines and produces the most competitive performance on seven benchmark datasets. Considering MF-based methods, RW-based methods, NN-based methods,

¹A reference implementation of CSGNN may be found at <https://github.com/BioMedicalBigDataMiningLab/CSGNN> or <https://github.com/ChengshuaiZhao0/CSGNN>

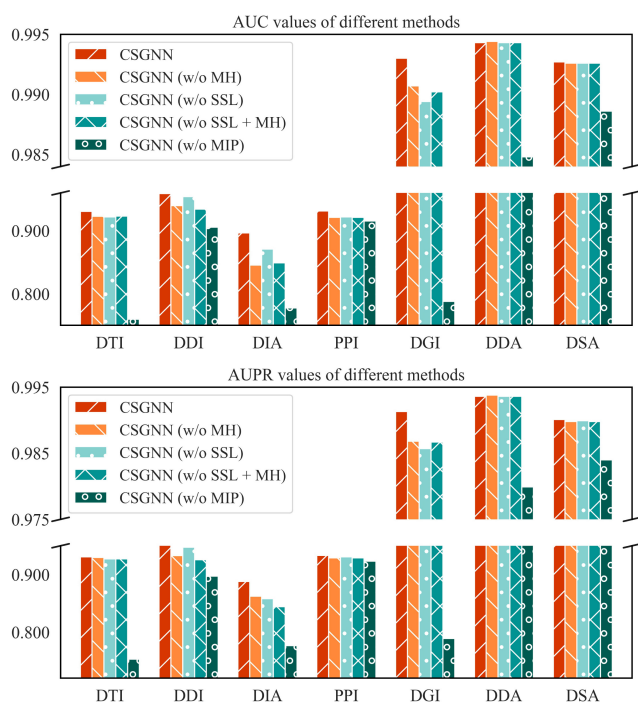


Figure 2: Results of ablation experiments.

and GNN-based methods, CSGNN achieves the average improvement of 20.8%, 15.34%, 17.38%, and 11.68% in AUC values, and 35.56%, 26.59%, 37.63%, and 11.9% in AUPR values, respectively. The results demonstrate the superiority of CSGNN when performing molecular interaction prediction. Secondly, CSGNN outperforms GCN and GIN by 2.58% and 1.7% in AUC values on average, and 2.54% and 1.85% in AUPR values on average, respectively. The results indicate that the consideration of the mix-hop context and the contrastive self-supervised learning strategy boosts the generalization ability of CSGNN, which will be further discussed in **Ablation Study**. Thirdly, CSGNN with different GNN encoders achieves different performances among benchmark datasets. For benchmark datasets with low density ($<0.1\%$), CSGNN with GCN encoder (i.e., CSGNN-GCN) performs better than CSGNN with GIN encoder (i.e., CSGNN-GIN), while for other benchmark datasets, CSGNN-GIN performs better than CSGNN-GCN. At last, we notice that several baseline methods (such as GAT and LINE) work well in node/graph classification do not obtain good performances on molecular interaction prediction, because of the distinct characteristics of biological networks.

4.3 Ablation Study

To further investigate the importance of components: the mix-hop aggregator and the self-supervised learning task, we design the following variants of CSGNN:

- **CSGNN without mix-hop** (w/o MH) only aggregates immediate neighbors instead of mix-hop neighbors in CSGNN.
- **CSGNN without self-supervised learning task** (w/o

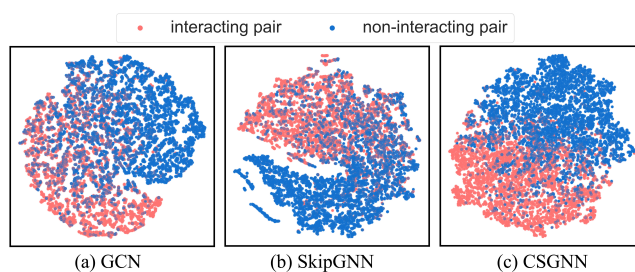


Figure 3: Visualization of DDI network.

SSL) removes the self-supervised learning task, and constructs GNN model based on mix-hop.

- **CSGNN without mix-hop and self-supervised learning task** (w/o MH + SSL) has neither mix-hop nor the self-supervised learning task.
- **CSGNN without molecular interaction prediction task** (w/o MIP) is trained only based on the contrastive self-supervised learning objective. Then, the acquired node representations will be used in downstream link prediction task.

We conduct ablation study on seven datasets, and the results are shown in Figure 2. It is clear that CSGNN with the mix-hop aggregator and the self-supervised learning task achieves superior performances on most settings, and the removal of any will undermine the predictive capacity of CSGNN. Overall, the mix-hop is more efficient than the contrastive self-supervised learning task. Besides, by introducing the contrastive self-supervised learning task, the datasets with few annotations (e.g., DGI and DIA) benefit more significantly compared to the datasets with sufficient ones (e.g., DDA and DSA). It's worth mentioning that only depending on the contrastive self-supervised learning task, CSGNN (w/o MIP) still achieves amazing performances, especially on DSA and PPI datasets. In conclusion, both mix-hop and the self-supervised learning task are necessary for CSGNN.

4.4 Visualization Analysis

In this section, we explore node representations on the molecule level. We extract the learned representations of molecule pairs in the interaction networks from the trained CSGNN model, and project them into 2D space using t-SNE [Maaten and Hinton, 2008]. Here, we take the **ChCh-Miner(DDI)** dataset as an example to compare CSGNN with GCN and SkipGNN. As illustrated in Figure 3, the proposed CSGNN can well distinguish the linked molecules (red) and unlinked molecules (blue); SkipGNN recognizes most of the non-interacting pairs; GCN differentiates part of molecular pairs. To further analyze the quality of learned representations, we calculate the inter-cluster and intra-cluster distances via silhouette scores. CSGNN, SkipGNN, and GCN have silhouette scores of 0.2792, 0.1777, and 0.1705 respectively, indicating that CSGNN allows more delicate representations for molecular interaction prediction.

5 Conclusion

In this paper, we focus on crucial but rarely studied multifarious molecular interaction prediction problem. Most existing methods are dedicated to specific molecular interaction prediction tasks, and cannot be well transferred to other tasks. The scarcity of annotated data restricts the predictive capacity of conventional GNNs. To address above limitations, we propose a novel method CSGNN which leverages the deep mix-hop graph neural network to capture high-order proximity and employs the contrastive graph neural network to enhance the transferable and generalized ability. We intuitively incorporate contrastive learning task into the self-supervised learning framework as auxiliary regularization. Comprehensive experiments on seven public datasets show that CSGNN outperforms classic and state-of-the-art molecular interaction prediction models. CSGNN can be applied to various novel molecular interaction prediction tasks merely based on annotated interactions. In the future, we will explore more powerful representation learning techniques for molecular interaction inference.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.62072206, 61772381); Huazhong Agricultural University Scientific & Technological Self-innovation Foundation; Huazhong Agricultural University Students Research Fund (SRF).

We thank anonymous reviewers for their insightful and helpful comments. Thanks for the discussions with the team member, Ph.D. candidate Haitao Fu, who provides a lot of insights for the work.

References

- [Abu-El-Haija *et al.*, 2019] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *ICML*, pages 21–29, 2019.
- [Cao *et al.*, 2015] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *CIKM*, pages 891–900, 2015.
- [Cheng *et al.*, 2012] Feixiong Cheng, Chuang Liu, Jing Jiang, Weiqiang Lu, Weihua Li, Guixia Liu, Wei-Xing Zhou, Jin Huang, and Yun Tang. Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Comput. Biol.*, 8(5), 2012.
- [Hamilton *et al.*, 2017] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, pages 1024–1034, 2017.
- [Hassani and Ahmadi, 2020] Kaveh Hassani and Amir Hosein Khas Ahmadi. Contrastive multi-view representation learning on graphs. In *ICML*, pages 4116–4126, 2020.
- [Hu *et al.*, 2020] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. GPT-GNN: generative pre-training of graph neural networks. In *KDD*, pages 1857–1867, 2020.
- [Huang *et al.*, 2018] Justin K. Huang, Daniel E. Carlin, Michael Ku Yu, Wei Zhang, Jason F. Kreisberg, Pablo Tamayo, and Trey Ideker. Systematic evaluation of molecular networks for discovery of disease genes. *Cell Systems*, 6(4):484–495.e5, 2018.
- [Huang *et al.*, 2020a] Feng Huang, Xiang Yue, Zhankun Xiong, Zhouxin Yu, Shichao Liu, and Wen Zhang. Tensor decomposition with relational constraints for predicting multiple types of microRNA-disease associations. *Briefings in Bioinformatics*, 2020.
- [Huang *et al.*, 2020b] Kexin Huang, Cao Xiao, Lucas M. Glass, Marinka Zitnik, and Jimeng Sun. SkipGNN: predicting molecular interactions with skip-graph networks. *Scientific Reports*, 10(1):21092, 2020.
- [Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [Kovács *et al.*, 2019] István A. Kovács, Katja Luck, Kerstin Spirohn, Yang Wang, Carl Pollis, Sadie Schlabach, Wenting Bian, Dae Kyum Kim, Nishka Kishore, and Tong Hao. Network-based prediction of protein interactions. *Nature Communications*, 10(1), 2019.
- [Liu *et al.*, 2020] Chuang Liu, Yifang Ma, Jing Zhao, Ruth Nussinov, Yi-Cheng Zhang, Feixiong Cheng, and Zi-Ke Zhang. Computational network biology: Data, models, and applications. *Physics Reports*, 846:1–66, 2020.
- [Luck *et al.*, 2020] Katja Luck, Dae-Kyum Kim, Luke Lambourne, Kerstin Spirohn, Bridget E Begg, Wenting Bian, Ruth Brignall, Tiziana Cafarelli, Francisco J Campos-Laborie, Benoit Charloteaux, et al. A reference map of the human binary protein interactome. *Nature*, 580(7803):402–408, 2020.
- [Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
- [Marinka Zitnik and Leskovec, 2018] Sagar Maheshwari Marinka Zitnik, Rok Sosič and Jure Leskovec. BioSNAP Datasets: Stanford biomedical network dataset collection. <http://snap.stanford.edu/biodata>, 2018. Accessed: 2020-09-25.
- [Ou *et al.*, 2016] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *KDD*, pages 1105–1114, 2016.
- [Peng *et al.*, 2020] Zhen Peng, Yixiang Dong, Minnan Luo, Xiao-Ming Wu, and Qinghua Zheng. Self-supervised graph representation learning via global context prediction. *arXiv preprint arXiv:2003.01604*, 2020.
- [Qiu *et al.*, 2020] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. GCC: graph contrastive coding for graph neural network pre-training. In *KDD*, pages 1150–1160, 2020.

- [Ribeiro *et al.*, 2017] Leonardo Filipe Rodrigues Ribeiro, Pedro H. P. Saverese, and Daniel R. Figueiredo. *struc2vec*: Learning node representations from structural identity. In *KDD*, pages 385–394, 2017.
- [Sun *et al.*, 2020a] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *ICLR*, 2020.
- [Sun *et al.*, 2020b] Ke Sun, Zhouchen Lin, and Zhanxing Zhu. Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes. In *AAAI*, pages 5892–5899, 2020.
- [Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: large-scale information network embedding. In *WWW*, pages 1067–1077, 2015.
- [Velickovic *et al.*, 2018] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.
- [Velickovic *et al.*, 2019] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *ICLR*, 2019.
- [Wang *et al.*, 2016] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *KDD*, pages 1225–1234, 2016.
- [Wu *et al.*, 2021] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.*, 32(1):4–24, 2021.
- [Xu *et al.*, 2019a] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.
- [Xu *et al.*, 2019b] Nuo Xu, Pinghui Wang, Long Chen, Jing Tao, and Junzhou Zhao. MR-GNN: multi-resolution and dual graph neural network for predicting structured entity interactions. In *IJCAI*, pages 3968–3974, 2019.
- [You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *NeurIPS*, 2020.
- [Yu *et al.*, 2020] Zhouxin Yu, Feng Huang, Xiaohan Zhao, Wenjie Xiao, and Wen Zhang. Predicting drug–disease associations through layer attention graph convolutional network. *Briefings in Bioinformatics*, 2020.
- [Zhang *et al.*, 2017] Wen Zhang, Yanlin Chen, Feng Liu, Fei Luo, Gang Tian, and Xiaohong Li. Predicting potential drug-drug interactions by integrating chemical, biological, phenotypic and network data. *Bmc Bioinformatics*, 18(1):18, 2017.
- [Zhang *et al.*, 2020a] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE Trans. Big Data*, 6(1):3–28, 2020.
- [Zhang *et al.*, 2020b] Zi-Chao Zhang, Xiao-Fei Zhang, Min Wu, Le Ou-Yang, Xing-Ming Zhao, and Xiao-Li Li. A graph regularized generalized matrix factorization model for predicting links in biomedical bipartite networks. *Bioinformatics*, 36(11):3474–3481, 2020.