

## Article

# Biomedical Interaction Prediction with Adaptive Line Graph Contrastive Learning

Shilin Sun, Hua Tian, Runze Wang and Zehua Zhang \*

College of Information and Computer, Taiyuan University of Technology, Jinzhong 030600, China

\* Correspondence: zhangzehua@tyut.edu.cn

**Abstract:** Biomedical interaction prediction is essential for the exploration of relationships between biomedical entities. Predicted biomedical interactions can help researchers with drug discovery, disease treatment, and more. In recent years, graph neural networks have taken advantage of their natural structure to achieve great progress in biomedical interaction prediction. However, most of them use node embedding instead of directly using edge embedding, resulting in information loss. Moreover, they predict links based on node similarity correlation assumptions, which have poor generalization. In addition, they do not consider the difference in topological information between negative and positive sample links, which limits their performance. Therefore, in this paper, we propose an adaptive line graph contrastive (ALGC) method to convert negative and positive sample links into two kinds of line graph nodes. By adjusting the number of intra-class line graph edges and inter-class line graph edges, an augmented line graph is generated and, finally, the information of the two views is balanced by contrastive learning. Through experiments on four public datasets, it is proved that the ALGC model outperforms the state-of-the-art methods.

**Keywords:** biomedical interaction; line graph; contrastive learning; adaptive

**MSC:** 05C85



**Citation:** Sun, S.; Tian, H.; Wang, R.; Zhang, Z. Biomedical Interaction Prediction with Adaptive Line Graph Contrastive Learning. *Mathematics* **2023**, *11*, 732. <https://doi.org/10.3390/math11030732>

Academic Editors: Di Jin and Liang Yang

Received: 30 December 2022

Revised: 16 January 2023

Accepted: 28 January 2023

Published: 1 February 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Biomedical interactions, such as drug–target interactions (DTIs) [1], drug–drug interactions (DDIs) [2], protein–protein interactions (PPIs) [3], and disease–disease associations (DDAs) [4], are essential in biological systems. Exploiting biomedical interactions can improve biomedical-related technologies, such as clinical treatments and drug discovery, as traditional wet experiments are inaccurate and require a lot of manpower.

Machine learning drives progress in biomedical interaction prediction. The most popular group of methods used for biomedical interaction prediction incorporates similarity-based features. For instance, Perlman et al. [5] integrated multiple drug–drug and gene–gene similarity measures to facilitate the prediction task using logistic regression to perform the prediction. Kastrin et al. [6] adopted physiological and side effects as features to input into models and established a logistic regression model to predict the interaction between biomedical entities. These functions could also be defined based on the topological properties of a multipartite network of existing entity pairs [7]. The accuracy of prediction can be improved by considering both the entity features and topological properties.

In recent years, numerous computational biomedical interaction prediction methods have been developed. Owing to the development of deep learning, the ability of graph representation learning to learn information about graph topological and node features has been improved [8]. Graph-based deep learning methods have dominated biomedical interaction prediction and can yield effective predictors and many methods have achieved state-of-the-art performance on related datasets. The graph contains a set of objects and relationships as a natural data structure to make full use of both entity features and topological properties [9]. Therefore, the biomedical interaction prediction task is transformed

into a link prediction task by making interactions edges and biomedical entities nodes. Most of the methods are based on GCN or GAT to obtain node embeddings by aggregating neighbor node information and they pass through the pooling layer to predict the link. Decagon [2] constructs a heterogeneous graph to obtain the embedding information of different types of nodes and utilizes the multi-relational link prediction model to predict polypharmacy side effects. However, these methods over-rely on the features of nodes and cause poor generalization. To better exploit the information of the biomedical interaction network structure and enhance the generalization of the model, SkipGNN [10] obtains higher-order neighbor information by constructing skip graphs to enhance the use of topology information. HOGCN [11] proposes a high-order graph convolutional network to collect neighbor node representations at different distances for biomedical interaction prediction. Considering the sparse imbalance of positive and negative sample links in the DTI task, Wang et al. [12] propose the Heterogeneous graph data Augmentation and node Similarity method, which is named HAS. However, GNN methods mainly focus on obtaining effective node embeddings and ignore the information associated with edges that can be beneficial [13].

In this paper, we propose the paradigm of Adaptive Line Graph Contrastive Learning (ALGC) for biomedical interaction prediction. To obtain more information on sparse datasets, ALGC adopts line graphs to convert the edges in the graph into line graph nodes; thus, it enables direct access to edge-to-edge relationships and converts the link prediction task to a node classification task. Therefore, positive and negative sample links can be converted into two kinds of line graph nodes. Considering the effect of intra-class and inter-class line graph edges on model performance, we use an adaptive augmented strategy based on prediction feedback to adjust the edges for an augmented line graph. Based on the theory that nodes of the same category in a graph are more inclined to connect, we increase the number of edges between the line graph nodes of the same class and reduce the number of edges between the line graph nodes of different classes. Finally, to make the line graph nodes of the same class more similar and the differences between the different classes greater, we adopt contrastive learning to maximize the information of double views.

- The proposed ALGC depends mainly on the interaction network topology, rather than the property features and internal structure features of biomedical entities, leading to better generalization and prediction performance.
- By converting the edges in the graph to line graph nodes, positive and negative sample links can be converted into two kinds of line graph nodes. ALGC directly adopts the relationship between edges. Further, the connection through the edges in the original graph is represented by inter- and intra-class line graph edges.
- We propose an adaptive augmented strategy to adjust the number of intra- and inter-class line graph edges to produce an augmented line graph.
- In order to increase the difference between the representations of positive and negative sample links, we introduce contrastive learning to balance the representation of the two views, and experiments on four different biomedical interaction network datasets validate the excellent performance of the model.

## 2. Related Work

### 2.1. Link Prediction Methods

These methods can be divided into three categories:

(1) **Heuristic methods** measure the similarity between entity pairs by the properties of the network such as structural properties. For instance, the common neighbors (CN) [14] method calculates the size of the common neighbors for a given pair of nodes. The Katz index [15] can be considered a variant of the shortest path metric based on global similarity indices. Li et al. [16] propose a motif-based index to capture local structural information and mix it with the similarity index in collaborative filtering to enhance its performance.

(2) **Network embedding methods** optimize an objective function to set up a model that is composed of several parameters [17]. To improve model accuracy, Gul et al. [18]

adopt a hill-climbing strategy to learn various local and quasi-topological features. However, due to the time-consuming calculation process, they are not suitable for real large networks.

(3) **Dimensionality reduction-based methods** mainly include graph embedding and matrix decomposition techniques [10]. Recently, Zhang et al. [19] proposed a new heuristic learning paradigm that unifies all three types of information (local subgraph, embedding, and attribute information) and utilizes the graph convolutional network. Sun et al. [13] represent the edges as subgraphs and line graph nodes separately and balance multi-scale information by contrastive learning.

## 2.2. Biomedical Interaction Prediction

The goal of biomedical interaction prediction is to predict whether pairs of biomedical entities interact. Currently, these methods can be classified into two groups:

(1) **Similarity-based methods** assume that entities with similar interaction patterns are more likely to interact. For example, BGMSDDA [20] developed a bipartite graph diffusion algorithm with multiple similarity integration for drug–disease association prediction. DeepDDI [21] applied a feed-forward neural network to encode structural similarity profiles (SSPs) of drugs to predict DDI. However, using similarity criteria based on known biomedical interactions to model complex biomedical interactions is insufficient, as the number of interactions identified is still sparse.

(2) **Methods based on network topology** utilize the network structure with node information to predict interactions. SkipGNN [10] aggregates higher-order information to predict interactions by constructing skip graphs. HOCGN [11] aggregates neighbor information at different distances and exhibits excellent performance on sparse biomedical interaction networks. However, this method uses node embedding to predict links based on assumptions, cannot directly use edge information, and ignores the difference between positive and negative sample links, resulting in information loss and limiting the generalization of the models.

## 2.3. Graph Contrastive Learning

Contrastive learning learns representations by maximizing the consistency of features across different views. For example, Wang et al. [22] consider the rich semantic relationships between pixels in different images and propose a pixel-based contrastive learning method that makes pixel embeddings belonging to the same semantic class more similar than pixel embeddings from different semantic classes. Scholars have extended the methods applicable to Euclidean spaces to non-Euclidean spaces. A typical graph contrastive learning (GCL) method first builds multiple graph views by random augmentation of the input data and then learns representations by comparing positive and negative samples. Graph contrastive learning can be divided into three categories in terms of contrastive modes, local–local methods, global–local methods, and global–global methods, which are based on the different scales of the views. The local level represents the node level and the global level represents the graph level. For example, HeCo [23] proposes a heterogeneous graph contrastive learning method, which generates two views according to the network scheme and meta-path to maximize the mutual information of the same nodes in different views. GRACE [24], GCA [25], and GROG [26] focus on contrasting views at the node level (local–local). DGI [27] and MVGRL [28] maximize the mutual information between the cross-view representations of nodes and graphs (global–local). GraphCL [29] applies a series of graph augmentations to generate an augmented graph and then learns to predict whether two graphs originate from the same graph (global–global).

## 3. Methods

### 3.1. Problem Formulation

Let  $\mathcal{G}$  be a biomedical interaction network, where biomedical entities form the nodes of the network. Then, the relationships between the entities form the links. For an arbitrary

graph  $g(A, X) \in \mathcal{G}$ ,  $X$  represents the biomedical entities' feature matrix and each node is represented by a one-hot-encoded feature vector, and  $A$  denotes the interaction relations of the biomedical entities. For any biomedical entity  $i, j$ , the purpose of the model is to predict the value of  $A_{ij}$ ; if it is 1, there is an interaction between the two biomedical entities; otherwise, there is not.

The general graph contrastive learning approach enhances model performance by maximizing the agreement of different views [28]. Given a graph  $g(A, X) \in \mathcal{G}$ , it first adopts a transformation method to obtain a line graph  $g_L(X_L, A_L)$ , and thus the line graph nodes represent the edges, as shown in Equation (1).  $X_L, A_L$  represent the line graph node feature matrix and the adjacency matrix of the line graph. The graph encoder  $f_\theta(\cdot)$  is used to obtain the line graph node representation  $H_L$  in Equation (2), and  $\theta$  denotes the parameters of the encoder.

$$g_L(X_L, A_L) = \text{transform}(g(A, X)) \quad (1)$$

$$H_L = f_\theta(g_L(X_L, A_L)) \quad (2)$$

We predict the results  $p$  of the node representation through a predictor composed of two fully connected layers. Then, we adjust the edges of the line graph according to the prediction results using the graph augmentation method  $\mathcal{T}$ . Next, the augmented line graph  $g_A$  is obtained, as shown in Equation (3):

$$g_A(X_A, A_A) = \mathcal{T}_p(g_L) \quad (3)$$

As shown in Equation (4), we input the augmented line graph into the shared graph encoder to obtain the new line graph node representations  $H_A$ . Finally, the objective function is obtained by maximizing the mutual information of the two views, as shown in Equation (5):

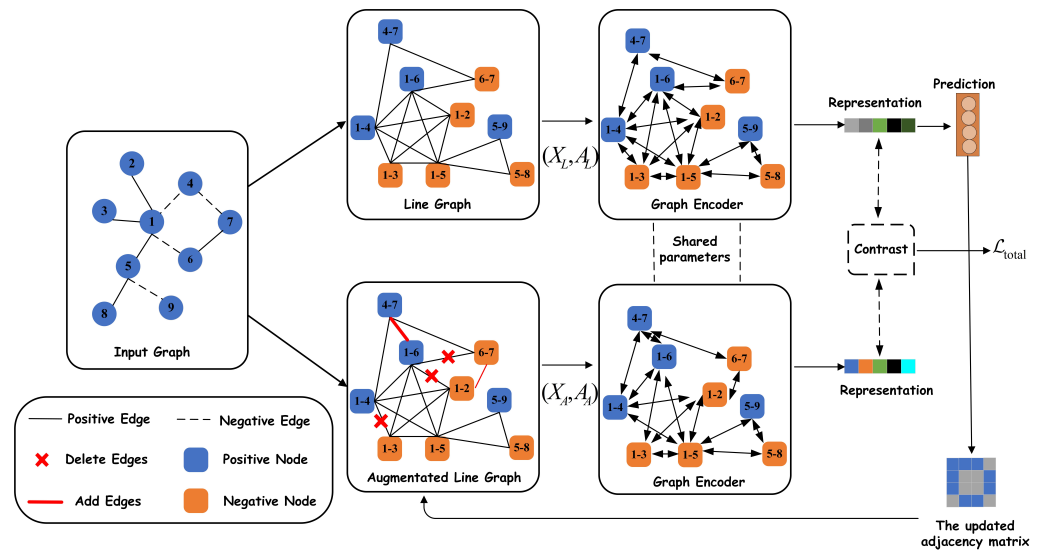
$$H_A = f_\theta(g_A(X_A, A_A)) \quad (4)$$

$$\max_{\theta} \sum_{g \in \mathcal{G}} MI(H_A, H_L) \quad (5)$$

$\theta$  denotes the parameters of the shared graph encoder  $f_\theta(\cdot)$ .

### 3.2. Overview

The model description is illustrated in Figure 1. First, the input biomedical interaction graph is converted into a line graph and the interactions among the biomedical entities are represented by line graph nodes, and second, the line graph is used to obtain the representation of the interactions through a shared graph encoder. Then, the classes of the line graph nodes are predicted, the adjacency matrix of the line graph is obtained according to the prediction results, the probability of connecting nodes of the same class is increased to obtain the enhanced graph of the line graph, the augmented line graph is passed through the graph encoder to obtain the node representation, and finally, the information of the two views is balanced by contrastive learning.



**Figure 1.** Overview of the ALGC. After converting the graph to a line graph, we adopt the prediction results to adjust the number of intra- and inter-class edges in the line graph to produce a new augmented line graph, and finally, we use contrastive learning to balance the information in both views.

### 3.3. Line Graph Construction

For the given graph  $\mathcal{G}$ ,  $h$ -hop subgraphs are first sampled with each edge as the center to obtain each subgraph  $g_i$ , and we adopt *Double-Radius Node Labeling (DRNL)* [19] to assign features to each node. Then, we can obtain the subgraph  $g(V, E, X, A)$  corresponding to any link. The edges of the subgraph are then represented by the nodes of the line graph, as shown in Equation (6). The features of a line graph node are obtained by concatenating the features of two nodes in its corresponding subgraph, as shown in Equation (7), and two line graph nodes are connected if the corresponding edges in the original graph share a node. Finally, the line graph  $g_L(X_L, A_L)$  is obtained.

$$V_L = \{e\}, \forall e \in E \quad (6)$$

$$X_L = \text{concat}(x_a, x_b) | \forall e_{(v_a, v_b)} \in V_L \quad (7)$$

### 3.4. Line Graph Encoder

The link prediction task is converted to a node classification task after we convert the subgraph to a line graph, as described above. Compared to indirect prediction links by learning node embedding, graph neural networks such as Kipf and Welling [30] show superior performance in node classification tasks [31]. Therefore, we create a multi-layer GCN encoder  $f_\theta(\cdot)$  for line graph node classification, as shown in Equation (8).

$$h_L^{l+1} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A}_L \tilde{D}^{-\frac{1}{2}} h_L^l W) \quad (8)$$

$h_L^l$  denotes the line graph node embedding at layer  $l$ ,  $\sigma$  is an activation function and  $\tilde{A}_L = A_L + I$ ,  $I$  denotes the normalized identity matrix.  $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$  represents the normalized degree matrices.

After we obtain the final representation of the node  $H_L$ , the node representation is passed through a fully connected layer of two layers  $FC$  to obtain the probability value of the predicted results  $P$  in Equation (9).

$$P = FC(H_L) \quad (9)$$

Then, the supervised loss  $\mathcal{L}_S$  can be obtained, as shown in Equation (10).

$$\mathcal{L}_S = -\frac{1}{|V_L|} \sum_{v_i \in V_L} y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i) \quad (10)$$

where  $|V_L|$  denotes the total number of line graph nodes or edges in the training dataset,  $p_i$  denotes the predicted probability value for link  $i$ , and  $y_i$  is the true value for link  $i$ .

### 3.5. Adaptive Line Graph Augmentation

In biomedical interaction networks, the known relational edges are usually referred to as positive samples, whereas the unknown edges are referred to as negative samples. After converting the graph to a line graph, the positive and negative samples can be represented by two types of nodes. Nodes with the same label share similar adjacency patterns and GCNs have the potential to achieve good performance [32]. On the other hand, in order to produce a more appropriate contrastive view, based on the “InfoMin principle”, the downstream task-related information is retained as much as possible by removing the noise information [33].

Therefore, we generate new views by adjusting the number of inter- and intra-class line graph edges by a certain ratio. We extend the AdaEdge [34] algorithm to adjust the intra- and inter-class edges of the line graph according to the prediction results. Finally, the adjacency matrix of the new augmented line graph is obtained. The details of the adaptive line graph augmentation algorithm are described in Algorithm 1.

---

#### Algorithm 1: Adaptive Line Graph Augmentation algorithm

---

**Input:** Line graph adjacency matrix  $A_L$ , Ratio  $r$ , Prediction Results  $P$ , Threshold  $t$

**Output:**  $A_A$

```

1 Get Node number  $N$  from  $A_L$ ;
2 Get Edge number  $M$  from  $A_L$ ;
3  $add\_count = M * r$ ;
4  $remove\_count = add\_count$ ;
5  $A_A = A_L$ ;
6 for  $i$  in  $(0, N)$  do
7   for  $j$  in  $(i, N)$  do
8     if  $\max(P[i], P[j]) > t$  and  $\min(P[i], P[j]) < t$  then
9       if  $A_L[i][j] == 1$  then
10         $A_A[i][j] = 0, A_A[j][i] = 0$ ;
11         $remove\_count - = 1$ ;
12        if  $remove\_count \leq 0$  then
13          break
14        end
15      end
16    end
17  else
18    if  $A_L[i][j] == 0$  then
19       $A_A[i][j] = 1, A_A[j][i] = 1$ ;
20       $add\_count - = 1$ ;
21      if  $add\_count \leq 0$  then
22        break;
23      end
24    end
25  end
26 end
27 end
28 Return  $A_A$ ;

```

---



### 3.6. Contrastive Learning

The contrastive learning of our proposed model is mainly a local–local-level contrast. We mainly adopt two views of the line graph and the augmented line graph for the contrast of the line graph nodes. For an arbitrary node  $v$  in a line graph, its positive samples are the corresponding nodes of the augmented line graph and its negative samples are the samples other than the positive samples. Thus, we can obtain the positive sample set  $\mathcal{P}$  and the negative sample set  $\mathcal{N}$ .

Following the overall design of the GRACE [35] model, in this paper, we use a form of the lower bound representation of mutual information, and the goal of InfoNCE [36] is to maximize the scores of the positive sample pairs and minimize the scores of the negative sample pairs, which takes the form shown in Equation (11).

$$\mathcal{L}_{\text{Cons}} = -\frac{1}{|T|} \sum_{v_i \in T} \sum_{p_j \in \mathcal{P}(v_i)} \log \frac{e^{S(v_i, p_j)/\tau}}{e^{S(v_i, p_j)/\tau} + \sum_{q_j \in \mathcal{N}(v_i)} e^{S(v_i, q_j)/\tau}} \quad (11)$$

$$S(\mu, v) = \frac{\mu^T v}{\|\mu\| \|v\|} \quad (12)$$

$S$  denotes the similarity measure function and for the two kinds of representations of nodes  $\mu, v$ , their similarity measures are shown in Equation (12). Moreover,  $T$  represents the training datasets of the line graph nodes and  $|T|$  denotes the number of line graph nodes in  $T$ .

The objective function of our model is obtained by combining the contrastive learning loss  $\mathcal{L}_{\text{Cons}}$  with the supervised loss  $\mathcal{L}_S$ , as shown in Equation (13).

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_S + \beta \mathcal{L}_{\text{Cons}} \quad (13)$$

Here,  $\alpha$  and  $\beta$  are the hyperparameters corresponding to the different loss components. The procedure of our proposed framework is shown in Algorithm 2.

---

#### Algorithm 2: Adaptive Line Graph Contrastive Learning algorithm

---

**input** : Biomedical interaction network  $\mathcal{G}$ , training edges  $T$ , shared graph encoder  $f_\theta(\cdot)$ , augmented function  $\mathcal{T}$   
**output**: Trained encoder  $f_\theta(\cdot)$  and parameters  $\theta$

- 1 **for** biomedical node pair  $(v_a, v_b) \in T$  **do**
- 2     Extract  $h$  – hop subgraph  $g(\mathcal{A}, \mathcal{X})$ ;
- 3     Obtain line graph  $g_L(\mathcal{X}_L, \mathcal{A}_L)$  via Equation (6)(7);
- 4     Put line graph  $g_L(\mathcal{X}_L, \mathcal{A}_L)$  in Line Graph Set  $T_L$ ;
- 5 **end**
- 6 **for** epoch = 1, ...,  $n_{\text{epochs}}$  **do**
- 7     Randomly split  $T_L$  in  $n/b$  batches  $B_i$  of size  $b$ ;
- 8     **foreach** batch  $B_i$  **do**
- 9         Obtain line graph node representation via Equation (8);
- 10        Obtain supervised losses, as shown in Equation (10);
- 11        Obtain the probability value of the predicted result  $\mathbf{P}$  via Equation (9);
- 12        Delete and add edges to obtain augmented line graph via Equation (3);
- 13        Obtain the augmented line graph node representation  $\mathbf{H}_A$  via Equation (4);
- 14        Obtain contrastive learning loss  $\mathcal{L}_{\text{Cons}}$ , as shown in Equation (11);
- 15        Calculate the total loss  $\mathcal{L}_{\text{total}}$ , as shown in Equation (13);
- 16        Update model parameters;
- 17     **end**
- 18 **end**
- 19 **Return** Encoder  $f_\theta(\cdot)$ , The parameters  $\theta$  obtained after training;

---

## 4. Experiment

### 4.1. Datasets and Experiment Setup

We conducted experiments on four datasets, ChCh-Miner (DDI) [37], ChG-Miner (DTI) [37], HuRI-PPI (PPI) [3], and DD-Miner (DDA) [37]. They are diverse biomedical interaction networks and include a drug–drug interaction network, drug–target interaction network, protein–protein interaction network, and disease–disease association network. The descriptions of the different biomedical interaction datasets are shown in Table 1.

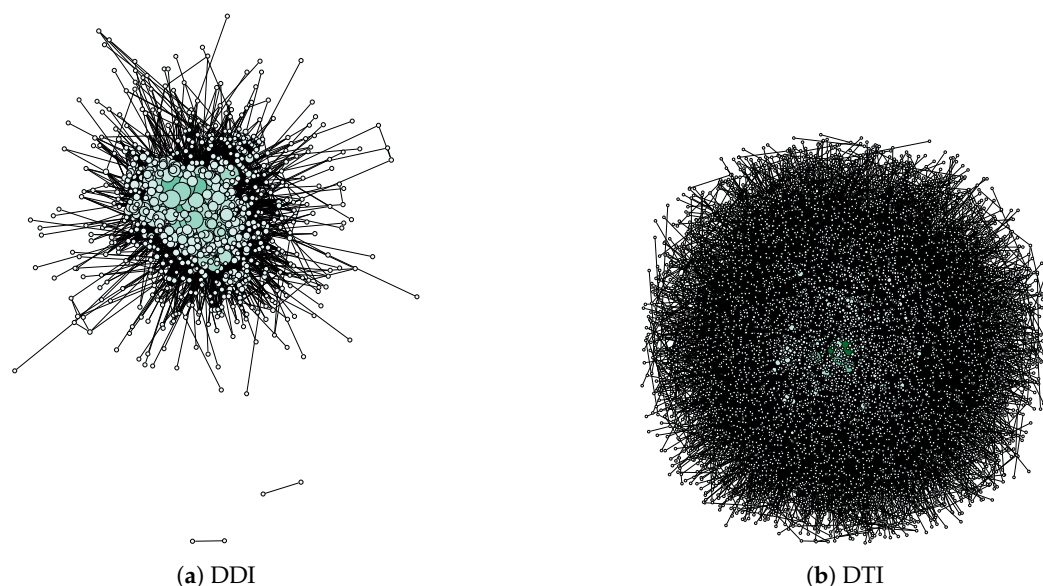
We ran the ALGC model and baselines with the Inspur heterogeneous cluster (GPU: 12\*32G Tesla V100s, memory: 640Gh DDR2). In the  $h$  – hop subgraph sampling method, the default  $h$  was 2, and the parameters were mainly optimized by Optuna [38]. The implementation of the ALGC model was based on PyGCL [39], which is a battery-included toolkit for implementing graph contrastive learning models. As for the evaluation metrics, we adopted the average precision (AP) and area under the receiver operating characteristic (AUROC).

**Table 1.** Summary of datasets used in our experiments.

Datasets	Entities	Nodes	Links	Density
DDI [37]	drug, drug	1514	48,514	4.23%
DTI [37]	drug, target	7343	15,139	0.06%
PPI [3]	protein, protein	5604	23,322	0.15%
DDA [37]	disease, disease	6878	6877	0.03%

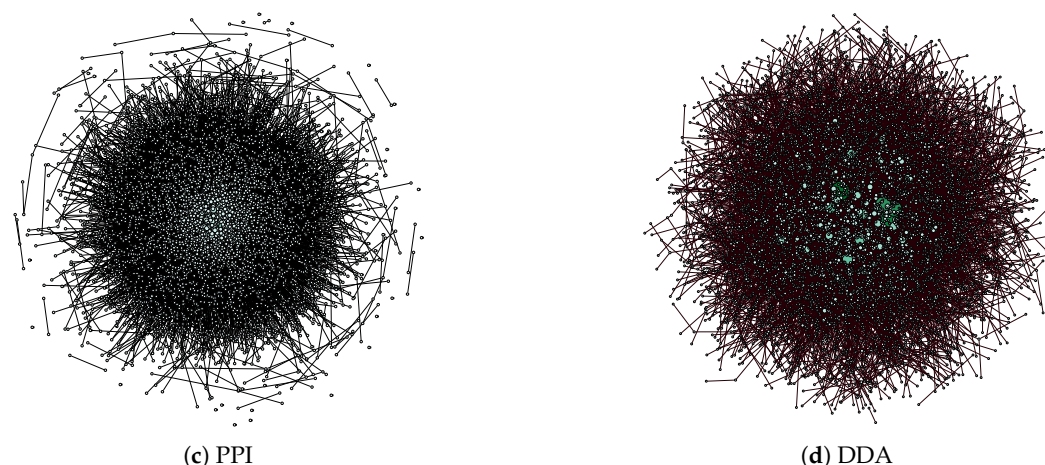
### 4.2. Dataset Visualization and Analysis

To obtain a macroscopic understanding of the datasets of the four different biomedical interaction networks, we visualized them to observe their network topology information. In each dataset, the size of the node and the shade of color were positively correlated with the degree of the node. The topologies of the four biomedical interaction networks were relatively different, as shown in Figure 2. Overall, the drug–drug interaction (DDI) network was dense; the larger the degree, the larger the node and the darker the node color. The drug–target interaction (DTI) network and disease–disease association (DDA) network were relatively sparse. According to the size and color of the nodes, it was found that the distribution of the node degrees in the DTI network was relatively uniform, whereas the distribution of the node degrees in the DDA network was imbalanced.



**Figure 2.** Cont.





**Figure 2.** Dataset visualization for the four datasets. The size and color shade of a node are positively correlated with the degree of the node. The network topologies of the four datasets vary greatly, and the DDI dataset is the densest.

#### 4.3. Baselines

##### Comparison Methods

##### 1. Heuristic methods:

- Common neighbors (CN) [40]: The simplest similarity metric based on node-local information is the common neighbors indicator. This metric is defined as the number of common neighbors between two nodes, that is, if there are more common neighbors between two nodes, they are more inclined to connect.

##### 2. Network embedding methods:

- DeepWalk [41] adopts the node-to-node co-occurrence relationship in the graph to learn the representation of the node.
- LINE [42] directly models the first-order and second-order neighbors of the graph. Unlike DeepWalk, LINE is modeled with the weights on the edges.
- Node2vec(N2V) [43] optimizes the random walk process of Deepwalk. This method comprehensively considers the wandering mode of BFS and DFS and proposes a bias random walk.

##### 3. Graph representation learning methods:

- SkipGNN [10] takes a biomedical interaction network as input to build a skip graph. This second-order network aims to capture skip similarity.
- HOGCN [11] collects neighbor feature representations at different distances to obtain the representation of the biomedical entity.

#### 4.4. Results and Analysis

We compared ALGC with different kinds of baseline methods on four different biomedical interaction datasets including drug–drug interaction (DDI), drug–target interaction (DTI), protein–protein interaction (PPI), and disease–disease association (DDA) datasets.

In order to verify the superiority of the method, we took 70% of the training edges as the training dataset, where the number of samples for the positive and negative edges were the same and the rest were used as the test samples.

As shown in Tables 2 and 3, compared with heuristic methods such as CN, ALGC significantly improved the performance of the network embedding-based methods, such as N2V, DeepWalk, and LINE, across all datasets. Overall, heuristic methods such as CN only consider the common neighbor situation and their performance is not as good as the method based on network embedding methods, which consider both the network topology and global information. N2V and DeepWalk expect nodes with higher second-order proximity

to produce similar low-dimensional representations, whereas LINE maintains both first- and second-order proximity so it has better performance.

**Table 2.** Comparison with baselines on training percentages (70%) (AUROC).

Model	DDI	DTI	PPI	DDA
CN [40]	0.7956	0.7183	0.7695	0.5543
DeepWalk [41]	0.8681	0.8348	0.8472	0.4145
LINE [42]	0.8947	0.8505	0.8493	0.6973
N2V [43]	0.7689	0.8051	0.7751	0.5032
SkipGNN [10]	0.8853	0.9215	0.9142	0.7245
HOGCN [11]	0.9125	0.9321	0.9216	0.7413
ALGC (Ours)	0.9542	0.9530	0.9481	0.8985

**Table 3.** Comparison with baselines on training percentages (70%) (AP).

Model	DDI	DTI	PPI	DDA
CN [40]	0.7603	0.6471	0.7105	0.4133
DeepWalk [41]	0.8390	0.5617	0.7108	0.1053
LINE [42]	0.8802	0.7732	0.7814	0.5556
N2V [43]	0.6075	0.4327	0.3415	0.1029
SkipGNN [10]	0.8587	0.9092	0.9092	0.7325
HOGCN [11]	0.8995	0.9334	0.9324	0.7512
ALGC (Ours)	0.9568	0.9594	0.9536	0.8902

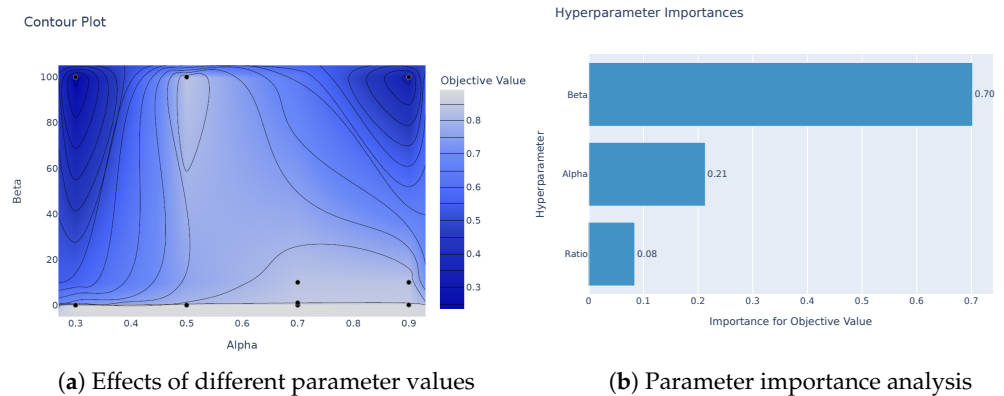
Methods based on graph neural networks not only consider topological information but also effectively learn node representations so their performance is significantly improved compared with other methods. Compared to SkipGNN, which aggregates second-order neighbor information by a skip graph, HOGCN achieves better performance by using the node information of neighbors at different distances. However, none of these methods can directly use the edge information, causing information loss. Additionally, they do not take into account the topological information of the negative samples. ALGC first uses the line graph to convert the edges into nodes, directly uses the edge information, and utilizes the topology information of the negative samples through contrastive learning. Therefore, the ALGC has better performance and robustness.

#### 4.5. Parametric Sensitivity Analysis

In our loss function calculation in Equation (13), there are two main hyperparameters  $\alpha$  and  $\beta$ . In addition, there is an adjustment ratio  $r$  for the intra-class line graph edges and the inter-class edges in Algorithm 1. We explored the effects among the different parameters through experiments on the DDA datasets. We used 70% of the links as the training set and the rest as the test set.

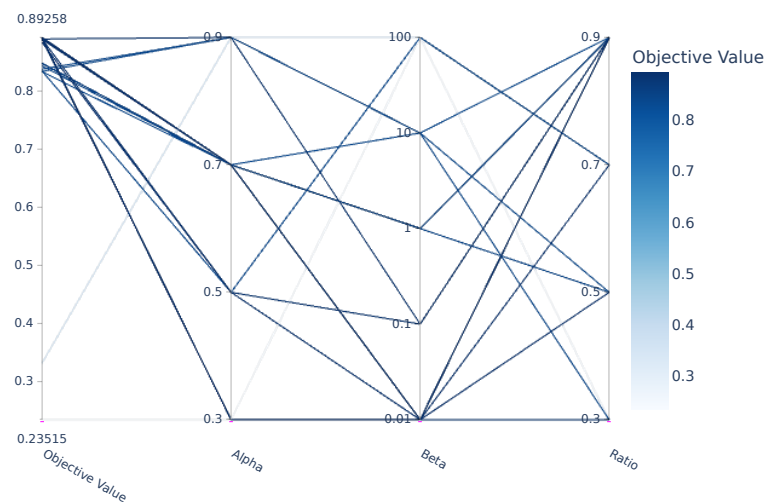
In order to analyze the relationship between  $\alpha$ ,  $\beta$ , and  $r$  in more detail, we used Optuna [38] to optimize the parameter search and set  $\alpha$  from  $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ ,  $\beta$  from  $\{0.01, 0.1, 1, 10, 100\}$ , and  $r$  from  $\{0.3, 0.5, 0.7, 0.9\}$ . Figure 3a shows the effects of  $\alpha$  and  $\beta$  and Figure 3b shows the importance between  $\alpha$ ,  $\beta$ , and  $r$ . We assess the importance of three parameters by an importance evaluation algorithm of Optuna, which is an automatic hyperparameter optimization software framework.

To better demonstrate the effects of the different parameter combinations, we plotted a parallel coordinate plot with Optuna. The figure shows the effects of the different parameter combinations. In Figure 4, we can see that  $\alpha = 0.7$ ,  $\beta = 0.01$ , and  $r = 0.9$  when the model performance is the best, and if Beta is too large or Ratio is too small, it will limit the performance of the model.



**Figure 3.** We adopt the Tree-structured Parzen Estimator algorithm to sample the parameters and optimize the value of the AUROC as the objective value, and denote  $\alpha$ ,  $\beta$ , and  $r$  by Alpha, Beta, and Ratio.

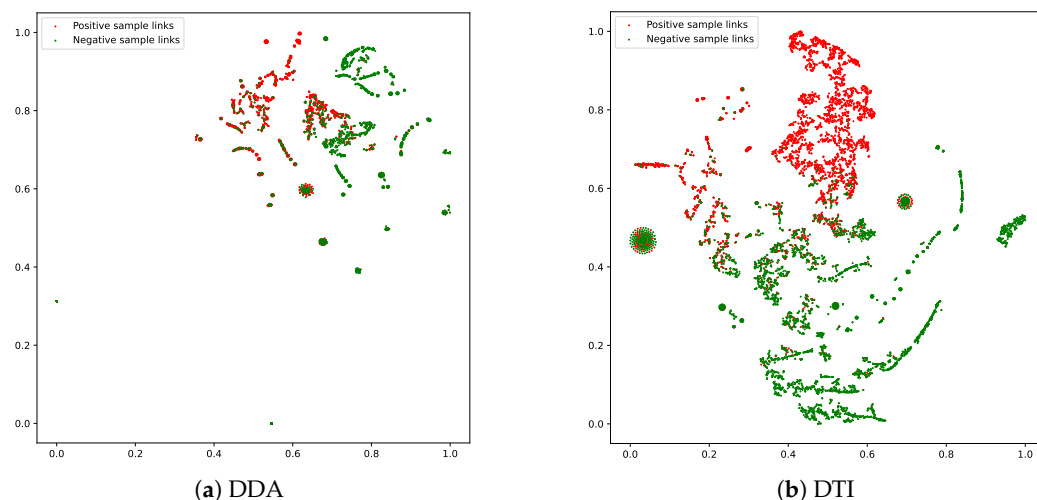
Parallel Coordinate Plot



**Figure 4.** Parallel coordinate plot with Optuna.

#### 4.6. Visualization Analysis

In this section, in the DDA and DTI networks, we visually analyze the learned line graph node representation of the trained model. The learned representation is reduced to a two-dimensional space by t-SNE [44], and finally, is visualized by matplotlib. As shown in Figure 5a,b, the red points represent the positive sample links and the green points represent the negative sample links. ALGC effectively distinguished the different types of edges and achieved perfect results.



**Figure 5.** Visualization of learned representation for links in DDA and DTI networks. The red dots indicate the positive sample links and the green dots indicate the negative sample links.

## 5. Conclusions

In order to directly exploit the relationship between positive and negative sample links in biomedical interaction networks, in this paper, we propose an adaptive line graph contrastive learning method by converting the graph to a line graph. The edges can be converted into line graph nodes to consider the topological relationship between the positive and negative sample links and the influence of the intra- and inter-class edges in the line graph on model performance. ALGC adopts the prediction results of the model to adjust the intra- and inter-class edges in the line graph to produce an augmented line graph. Finally, the information of the two views is balanced using contrastive learning. Comprehensive experiments prove that ALGC outperforms the state-of-the-art models on all four datasets.

We still have a lot of problems to solve and in future work, we will continue to explore the application of line graphs when the number of positive and negative sample links is unbalanced, as well as the application of line graphs in multi-relationship biomedical interaction predictions. We will also take advantage of rich information in the chemical structure of molecules to conduct in-depth exploration in the field of biomedicine. In addition, some works have begun to explore the relationship between hypergraphs and line graphs, using their topological characteristics to achieve information aggregation at different scales [45]. We will also consider integrating line graphs into hypergraphs for biomedical interaction predictions.

**Author Contributions:** Conceptualization, S.S.; methodology, S.S., Z.Z. and R.W.; software, S.S. and R.W.; validation, S.S., H.T. and R.W.; formal analysis, S.S., H.T., R.W. and Z.Z.; investigation, S.S. and Z.Z.; resources, S.S., H.T. and Z.Z.; data curation, S.S. and R.W.; writing—original draft preparation, S.S. and Z.Z.; writing—review and editing, S.S., Z.Z., R.W. and H.T.; visualization, S.S., R.W. and Z.Z.; supervision, S.S., Z.Z. and H.T.; project administration, S.S. and Z.Z.; funding acquisition, Z.Z. and H.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China (61702356, 51901152), the Industry University Cooperation Education Program of the Ministry of Education (2020021680113), and the Shanxi Scholarship Council of China.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Luo, Y.; Zhao, X.; Zhou, J.; Yang, J.; Zhang, Y.; Kuang, W.; Peng, J.; Chen, L.; Zeng, J. A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nat. Commun.* **2017**, *8*, 1–13. [\[CrossRef\]](#)
2. Zitnik, M.; Agrawal, M.; Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* **2018**, *34*, i457–i466. [\[CrossRef\]](#)
3. Luck, K.; Kim, D.K.; Lambourne, L.; Spirohn, K.; Begg, B.E.; Bian, W.; Brignall, R.; Cafarelli, T.; Campos-Laborie, F.J.; Charlotiaux, B.; et al. A reference map of the human binary protein interactome. *Nature* **2020**, *580*, 402–408. [\[CrossRef\]](#)
4. Xiang, J.; Zhang, J.; Zhao, Y.; Wu, F.X.; Li, M. Biomedical data, computational methods and tools for evaluating disease–disease associations. *Briefings Bioinform.* **2022**, *23*, bbac006. [\[CrossRef\]](#)
5. Perlman, L.; Gottlieb, A.; Atias, N.; Rupp, E.; Sharan, R. Combining drug and gene similarity measures for drug-target elucidation. *J. Comput. Biol.* **2011**, *18*, 133–145. [\[CrossRef\]](#)
6. Kastrin, A.; Ferk, P.; Leskošek, B. Predicting potential drug–drug interactions on topological and semantic similarity features using statistical learning. *PLoS ONE* **2018**, *13*, e0196865. [\[CrossRef\]](#)
7. Takarabe, M.; Kotera, M.; Nishimura, Y.; Goto, S.; Yamanishi, Y. Drug target prediction using adverse event report systems: A pharmacogenomic approach. *Bioinformatics* **2012**, *28*, i611–i618. [\[CrossRef\]](#)
8. Du, H.Y.; Wang, W.J. A Clustering Ensemble Framework with Integration of Data Characteristics and Structure Information: A Graph Neural Networks Approach. *Mathematics* **2022**, *10*, 1834. [\[CrossRef\]](#)
9. Yi, H.C.; You, Z.H.; Huang, D.S.; Kwok, C.K. Graph representation learning in bioinformatics: Trends, methods and applications. *Briefings Bioinform.* **2022**, *23*, bbab340. [\[CrossRef\]](#)
10. Huang, K.; Xiao, C.; Glass, L.M.; Zitnik, M.; Sun, J. SkipGNN: Predicting molecular interactions with skip-graph networks. *Sci. Rep.* **2020**, *10*, 1–16. [\[CrossRef\]](#)
11. Kishan, K.; Li, R.; Cui, F.; Haake, A.R. Predicting biomedical interactions with higher-order graph convolutional networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2021**, *19*, 676–687.
12. Wang, R.; Zhang, Z.; Zhang, Y.; Jiang, Z.; Sun, S.; Zhang, C. Sparse Imbalanced Drug-Target Interaction Prediction via Heterogeneous Data Augmentation and Node Similarity. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Chengdu, China, 16–19 May 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 548–561.
13. Sun, S.; Zhang, Z.; Wang, R.; Tian, H. Multi-scale Subgraph Contrastive Learning for Link Prediction. In Proceedings of the International Joint Conference on Rough Sets, Suzhou, China, 11–14 November 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 217–223.
14. Newman, M.E. Clustering and preferential attachment in growing networks. *Phys. Rev. E* **2001**, *64*, 025102. [\[CrossRef\]](#)
15. Katz, L. A new status index derived from sociometric analysis. *Psychometrika* **1953**, *18*, 39–43. [\[CrossRef\]](#)
16. Li, C.; Yang, Q.; Pang, B.; Chen, T.; Cheng, Q.; Liu, J. A Mixed Strategy of Higher-Order Structure for Link Prediction Problem on Bipartite Graphs. *Mathematics* **2021**, *9*, 3195. [\[CrossRef\]](#)
17. Clauset, A.; Moore, C.; Newman, M.E. Hierarchical structure and the prediction of missing links in networks. *Nature* **2008**, *453*, 98–101. [\[CrossRef\]](#)
18. Gul, H.; Al-Obeidat, F.; Amin, A.; Moreira, F.; Huang, K. Hill Climbing-Based Efficient Model for Link Prediction in Undirected Graphs. *Mathematics* **2022**, *10*, 4265. [\[CrossRef\]](#)
19. Zhang, M.; Chen, Y. Link prediction based on graph neural networks. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 2478.
20. Xie, G.; Li, J.; Gu, G.; Sun, Y.; Lin, Z.; Zhu, Y.; Wang, W. BGMSDDA: A bipartite graph diffusion algorithm with multiple similarity integration for drug–disease association prediction. *Mol. Omics* **2021**, *17*, 997–1011. [\[CrossRef\]](#)
21. Kim, H.U.; Ryu, J.; Lee, S. DeepDDI for understanding pharmacological effects of natural products. In Proceedings of the Natural Products-Discovery, Biosynthesis and Application, Copenhagen Bioscience Conferences, Hillerød, Denmark, 5–9 May 2019; Novo Nordisk Foundation: Hellerup, Denmark, 2019.
22. Wang, W.; Zhou, T.; Yu, F.; Dai, J.; Konukoglu, E.; Van Gool, L. Exploring Cross-Image Pixel Contrast for Semantic Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 7303–7313.
23. Wang, X.; Liu, N.; Han, H.; Shi, C. Self-supervised Heterogeneous Graph Neural Network with Co-contrastive Learning. In Proceedings of the KDD’21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, 14–18 August 2021; pp. 1726–1736.
24. Tapley, B.D.; Bettadpur, S.; Ries, J.C.; Thompson, P.F.; Watkins, M.M. GRACE measurements of mass variability in the Earth system. *Science* **2004**, *305*, 503–505. [\[CrossRef\]](#)
25. Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; Wang, L. Graph contrastive learning with adaptive augmentation. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 2069–2080.
26. Jovanović, N.; Meng, Z.; Faber, L.; Wattenhofer, R. Towards robust graph contrastive learning. *arXiv* **2021**, arXiv:2102.13085.
27. Velickovic, P.; Fedus, W.; Hamilton, W.L.; Liò, P.; Bengio, Y.; Hjelm, R.D. Deep Graph Infomax. *ICLR (Poster)* **2019**, *2*, 4.
28. Hassani, K.; Khasahmadi, A.H. Contrastive multi-view representation learning on graphs. In Proceedings of the International Conference on Machine Learning. PMLR, Virtual Event, 13–18 July 2020; pp. 4116–4126.



29. You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; Shen, Y. Graph contrastive learning with augmentations. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 5812–5823.
30. Welling, M.; Kipf, T.N. Semi-supervised classification with graph convolutional networks. In Proceedings of the International Conference on Learning Representations (ICLR 2017), Toulon, France, 24–26 April 2017.
31. Cai, L.; Li, J.; Wang, J.; Ji, S. Line graph neural networks for link prediction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 5103–5113. [[CrossRef](#)]
32. Ma, Y.; Liu, X.; Shah, N.; Tang, J. Is homophily a necessity for graph neural networks? *arXiv* **2021**, arXiv:2106.06134.
33. Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; Isola, P. What makes for good views for contrastive learning? *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6827–6839.
34. Chen, D.; Lin, Y.; Li, W.; Li, P.; Zhou, J.; Sun, X. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 3438–3445.
35. Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; Wang, L. Deep graph contrastive representation learning. *arXiv* **2020**, arXiv:2006.04131.
36. Oord, A.v.d.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748.
37. Zitnik, M.; Sosič, R.; Maheshwari, S.; Leskovec, J. BioSNAP Datasets: Stanford Biomedical Network Dataset Collection. 2018. Available online: <http://snap.stanford.edu/biodata> (accessed on 9 October 2022).
38. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2623–2631.
39. Zhu, Y.; Xu, Y.; Liu, Q.; Wu, S. An Empirical Study of Graph Contrastive Learning. *arXiv* **2021**, arXiv:2109.01116.
40. Liben-Nowell, D.; Kleinberg, J. The link prediction problem for social networks. In Proceedings of the Twelfth International Conference on Information and Knowledge Management, New Orleans, LA, USA, 3–8 November 2003; pp. 556–559.
41. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
42. Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 1067–1077.
43. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
44. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
45. Xia, X.; Yin, H.; Yu, J.; Wang, Q.; Cui, L.; Zhang, X. Self-supervised hypergraph convolutional networks for session-based recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 4503–4511.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.