



# A Graph Representation Learning Framework Predicting Potential Multivariate Interactions

Yanlin Yang<sup>1,2,3</sup> · Zhonglin Ye<sup>1,2,3</sup> · Haixing Zhao<sup>1,2,3</sup> · Lei Meng<sup>1,2,3</sup>

Received: 26 April 2023 / Accepted: 27 August 2023  
© The Author(s) 2023

## Abstract

Link prediction is a widely adopted method for extracting valuable data insights from graphs, primarily aimed at predicting interactions between two nodes. However, there are not only pairwise interactions but also multivariate interactions in real life. For example, reactions between multiple proteins, multiple compounds, and multiple metabolites cannot be mined effectively using link prediction. A hypergraph is a higher-order network composed of nodes and hyperedges, where hyperedges can be composed of multiple nodes, and can be used to depict multivariate interactions. The interactions between multiple nodes can be predicted by hyperlink prediction methods. Since hyperlink prediction requires predicting the interactions between multiple nodes, it makes the study of hyperlink prediction much more complicated than that of other complex networks, thus resulting in relatively limited attention being devoted to this field. The existing hyperlink prediction can only predict potential hyperlinks in uniform hypergraphs, or need to predict hyperlinks based on the candidate hyperlink sets, or only study hyperlink prediction for undirected hypergraphs. Therefore, a hyperlink prediction framework for predicting multivariate interactions based on graph representation learning is proposed to solve the above problems, and then the framework is extended to directed hyperlink prediction (e.g., directed metabolic reaction networks). Furthermore, any size of hyperedges can be predicted by the proposed hyperlink prediction algorithm framework, whose performance is not affected by the number of nodes or the number of hyperedges. Finally, the proposed framework is applied to both the biological metabolic reaction network and the organic chemical reaction network, and experimental analysis has demonstrated that the hyperlinks can be predicted efficiently by the proposed hyperlink prediction framework with relatively low time complexity, and the prediction performance has been improved by up to 40% compared with the baselines.

**Keywords** Hypergraph · Hyperlink prediction · Multivariate interactions · Graph representation learning · Biological metabolic reaction network · Organic chemical reaction network

---

✉ Haixing Zhao  
haixing\_zhao@163.com

Yanlin Yang  
yanlin\_yang@foxmail.com

Zhonglin Ye  
zhonglin\_ye@foxmail.com

Lei Meng  
lei\_meng@foxmail.com

<sup>1</sup> College of Computer, Qinghai Normal University, Xining 810001, Qinghai, China

<sup>2</sup> The State Key Laboratory of Tibetan Intelligent Information Processing and Application, Xining 810008, Qinghai, China

<sup>3</sup> Tibetan Information Processing Engineering Technology and Research Center of Qinghai Province, Xining 810008, Qinghai, China

## 1 Introduction

Link prediction [1–4] is defined as predicting the connection possibility of any two disconnected nodes in an ordinary graph by utilizing the attributes of nodes, the topological information of the graph, and so on. Link prediction has essential applications in real life, such as predicting protein–protein interactions [5], predicting the possibility of people becoming friends in social networks [6, 7], predicting the possibility of two people publishing papers together [8, 9], recommender systems [10–13], drug discovery [14], knowledge acquisition [15], and label classification [16]. However, there are not only pairwise relationships, but also many higher-order relationships beyond pairwise interactions exist in real life. For example, in scientific collaboration networks [17], an article is often published by more

than two authors, and in chemical reaction networks [18, 19], there are not only reactions between two compounds, but also reactions beyond two compounds. Hypergraph [20–24] is composed of nodes and hyperedges that can model the relationships beyond pairwise interactions. Unlike ordinary graphs, where edges can only connect two nodes, but hyperedges in hypergraphs can connect any number of nodes. Hyperlink prediction can be defined as predicting the possibility of any number of nodes forming hyperedges in the hypergraph by utilizing the hyperedge information, node attributes, hypergraph structure information, and so on. Hyperlink prediction can be used to predict the possibility of any number of authors collaborating to publish an article, as well as the possibility of any number of compounds producing chemical reactions.

Although hyperlink prediction is an important research problem, there are some difficulties to research it, for example, the problems of variable cardinality of hyperedges and prediction of directed hyperlink. In an ordinary graph ( $n$  nodes), the number of potential edges is only  $n^2$ , but in a hypergraph ( $n$  nodes), the number of potential hyperedges is exponentially  $2^n$ . If all possible hyperlinks are listed, there will be a total of  $2^n$  possibilities, so it is impractical to list, score and rank all hyperedges.

Currently, some researchers have also proposed hyperlink prediction methods. For example, Xu et al. [25] propose a hyperlink prediction method by using the hidden social feature information, and they solve the problem of predicting the relationship beyond pairwise interactions for the first time. Zhang et al. [26] consider that many potential hyperlinks are not consistent with the actual situation and the candidate hyperlink sets that satisfy the actual situation can be obtained by some special restrictive condition, and then utilize coordination matrix minimization (CMM) to propose a hyperlink prediction algorithm. Sharma et al. [27] propose a clique-closure hypothesis, and then utilize coordination matrix minimization to predict hyperlinks based on this hypothesis. Kumar et al. [28] propose a resource allocation-based hyperlink prediction algorithm with consulting the idea of hypergraph evolution model. Pan et al. [29] propose the loop structures based on nodes and hyperlinks, and then use modified logistic regression to transform the loop structure features into a hyperlink prediction method. Maurya and Ravindran [30] propose a hyperlink prediction algorithm based on  $k$ -uniform hypergraph by using tensor eigenvalue decomposition of hypergraph Laplacian. Fatemi et al. [31] propose two embedding methods for knowledge hypergraphs, named HsimpleE and HypE, and solve the hyperlink prediction problem in knowledge hypergraphs for the first time. In addition, some researchers have proposed negative sampling methods [32, 33] to improve the hyperlink prediction process, and apply these negative sampling methods to

hyperlink prediction algorithms to improve their hyperlink prediction performance.

Graph neural networks (GNNs) [34–38] have been growing at a very fast pace. Zhang and Chen [39] apply graph neural networks to link prediction based on binary relations in ordinary graphs, and then excellent results are obtained. However, there are few studies applying neural networks to hyperlink prediction. Yadati et al. [40] use graph convolutional network (GCN) to extract the node feature information, and then propose a GCN-based hyperlink prediction algorithm, abbreviated as NHP, and the experimental results show that the hyperlink prediction performance of NHP is better than the classic methods such as CMM [26] and SHC [41]. GCN is a model that can perform convolutional neural networks on graph structure data, and it extends convolution operation to irregular or non-Euclidean graph structure data such as metabolic networks, social networks, and telecommunication networks. Multi-layer GCN can be used to propagate the node feature information. However, the single layer GCN equation is stiffly used in the NHP method in the information propagation process, which does not achieve the effect of flexibly using GCN ideas to transfer the feature information of nodes.

Most of the hyperlink prediction methods introduced above focus on predicting hyperlinks in undirected hypergraphs. How to predict directed hyperlinks is a difficult and worthy problem. Especially in chemical reaction and metabolic reaction networks, the choice of reactants and products directly affects whether the chemical reaction will actually occur. For example, in the same hyperedge, the reaction between compounds  $\text{Cu}(\text{OH})_2$  and  $\text{HCl}$  will produce compounds  $\text{CuCl}_2$  and  $\text{H}_2\text{O}$ , but compounds  $\text{CuCl}_2$  and  $\text{HCl}$  will not react and produce compounds  $\text{Cu}(\text{OH})_2$  and  $\text{H}_2\text{O}$ . Therefore, it is urgent to solve the problem of how to predict hyperlink for directed hypergraphs.

Based on the above problems, A hyperlink prediction framework based on graph representation learning is proposed, which can apply different graph representation learning methods to obtain hypergraph structure data and predict the interaction between multiple targets. GCN and GAT focus on the influence of neighboring nodes on the central node, which can enhance the ability of models to perceive the neighborhood, thereby strengthening the joint spatial representation ability of models. The framework is then incubated with two methods, one is hyperlink prediction based on graph convolutional networks (GCN) and the other is hyperlink prediction based on graph attention networks (GAT). Finally, these two methods are extended to directed hyperlink prediction. Since GCN and GAT have strong ability to perceive the neighborhood, they can represent the characteristic attributes of nodes in the hyperedges effectively, which gives them a clear advantage in improving

their performance in hyperlinks prediction. Hence, the main contributions of this article are as follows:

(1) A hyperlink prediction framework based on graph representation learning is proposed, which can be applied not only to hyperlink prediction in undirected hypergraphs but also in directed hypergraphs. (2) The any size of hyperlinks can be predicted by the proposed methods without the need of candidate hyperlink sets, and unseen hyperlinks also can be predicted under the problem of variable cardinality of hyperedges. (3) The proposed hyperlink prediction algorithm framework is applied to both the biological metabolic reaction network and the organic chemical reaction network, and experimental analysis show that the two proposed hyperlink prediction algorithms can effectively predict multivariate interactions in both undirected and directed hypergraphs with relatively low time complexity.

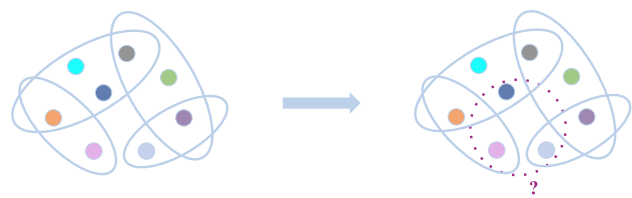
## 2 Preliminaries

### 2.1 Undirected Hypergraphs

The definition of hypergraph [24] was first proposed by Berge in 1970.

Suppose  $V = \{v_1, v_2, \dots, v_n\}$  is a finite set and  $E = \{e_1, e_2, \dots, e_m\}$ ,  $G = (V, E)$  is a hypergraph if (1)  $e_i \neq \emptyset$  ( $i = 1, 2, \dots, m$ ) and 2)  $\bigcup_{i=1}^m e_i = V$  are satisfied. The elements  $v_1, v_2, \dots, v_n$  are the nodes of the undirected hypergraph  $G$ ,  $n = |V|$  denotes the number of nodes in  $G$ , the elements  $e_i = \{v_{i_1}, v_{i_2}, \dots, v_{i_j}\}$  ( $1 \leq j \leq n$ ) consisting of subsets of  $V$ , is usually called the hyperedge of the hypergraph  $G$ ,  $|e_i|$  denotes the number of nodes contained in the hyperedge  $e_i$ , and  $m = |E|$  denotes the number of hyperedges in the hypergraph.

A hypergraph is said to be a uniform hypergraph when the number of nodes in each hyperedge is equal, and a hypergraph is said to be a  $k$ -uniform hypergraph when the number of nodes in each hyperedge is equal to  $k$ . In an undirected hypergraph, each hyperedge contains a different set of nodes, and the total number of potential hyperedges is  $2^n$ , and the set of unobserved hyperedges is  $E' = 2^n - E$ . Assuming that  $G$  is an incomplete undirected hypergraph, the hyperlink prediction of an undirected hypergraph can be defined as predicting the possibility of any number of nodes forming hyperedges in the hypergraph by utilizing the hyperedge information, node attributes, hypergraph structure information, and so on. As shown in Fig. 1, which represents the simplified process of hyperlink prediction for a synthetic undirected hypergraph, we need to predict the hyperlinks that are not currently observed but may be formed with time evolution by some hyperlink prediction methods.



**Fig. 1** Simplified process of hyperlink prediction for a synthetic undirected hypergraph. The hypergraph has 8 nodes, 3 hyperedges, and  $2^8$  potential hyperedges. We need to predict the nodes who may form hyperedges together in the future using some hyperlink prediction methods

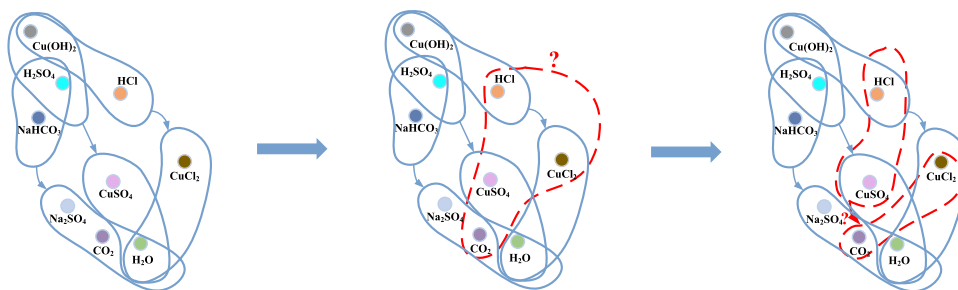
However, it is impractical to predict all possible hyperedges from an exponential hyperedge set. In fact, it is unnecessary to predict the set of all possible hyperedges. When given a particular dataset, many of the predicted hyperedges are not consistent with the actual situation. For example, in metabolic reactions, many predicted hyperedges do not have biological significance.

### 2.2 Directed Hypergraphs

A directed hypergraph [42] is denoted by unordered pair  $\vec{G} = (V, \vec{E})$ , where  $V = \{v_1, v_2, \dots, v_n\}$ ,  $\vec{E} = \{\vec{e}_1, \vec{e}_2, \dots, \vec{e}_m\}$ , the elements  $v_1, v_2, \dots, v_n$  are the nodes of the directed hypergraph  $\vec{G}$ , the elements  $\vec{e}_1, \vec{e}_2, \dots, \vec{e}_m$  are the hyperedges of  $\vec{G}$ . Specifically, the directed hyperedge consists of two parts, the head and tail of the hyperedge, respectively, that is,  $\vec{e}_i = (\vec{e}_i^+, \vec{e}_i^-)$ , where  $\vec{e}_i^+$  is called the tail of the directed hyperedge, and  $\vec{e}_i^-$  is called the head. Then, the hyperlink prediction of a directed hypergraph can be defined as using the observed directed hyperedges, node features, and hypergraph structure information to predict the possibility of any number of nodes in the hypergraph forming hyperedges and to predict the direction of the hyperedges. A simplified process of chemical reaction prediction for a synthetic directed hypergraph is shown in Fig. 2.

## 3 Proposed Methods

Graph Neural Networks (GNNs) are used to obtain graph structure data in this article, and then a GNNs-based hyperlink prediction framework is proposed. The framework is extended to two methods used for undirected hyperlink prediction and directed hyperlink prediction, which are the hyperlink prediction method based on GCN and the hyperlink prediction method based on GAT. These two proposed methods will be introduced in detail in the following.



**Fig. 2** Simplified process for predicting hyperlinks in a synthetic directed chemical reaction hypergraph. The nodes in this hypergraph represent compounds and the hyperedges represent chemical reactions. The hypergraph has 9 nodes, 3 directed hyperedges, and  $2^9-3$  potential hyperedges, which means that there are  $2^9-3$  possibilities for

future chemical reactions. We need to use hyperlink prediction methods to predict the compounds that are likely to react in the future, and then predict the direction of chemical reactions, that is, predict the head and tail of the directed hyperedges

**Table 1** Main symbols comparison

$G$	Hypergraph
$X$	Node feature matrix
$\tilde{A}$	Adjacency matrix after adding self-loops
$\tilde{D}$	Degree matrix after adding self-loops
$W$	Weight matrix
$\bar{A}$	Normalized adjacency matrix
$x_u$	Feature information of node $u$
$h^e$	Representation information of the hyperedge $e$
$s_e$	The scoring function
$b$	Bias
$\sigma$	Sigmoid activation function
$L$	Objective function
$\lambda$	Hyperparameter

### 3.1 Main Symbols Comparison Table

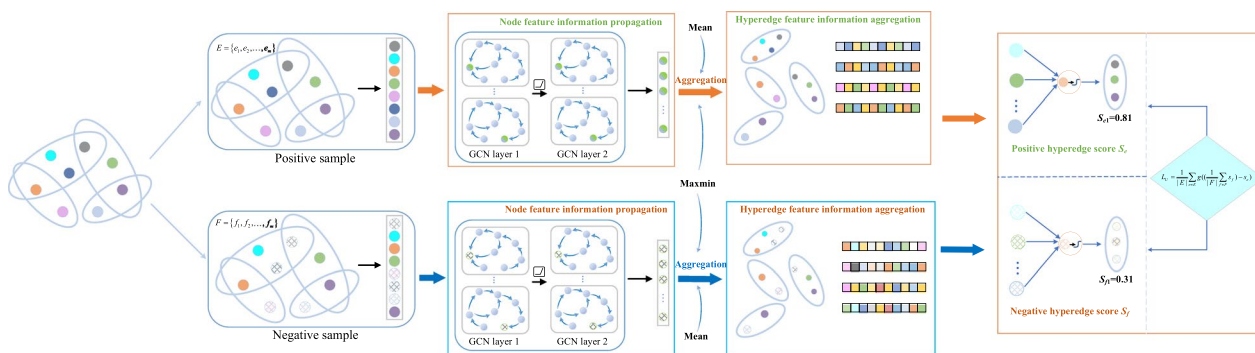
As shown in Table 1, the main comparison table of the symbols is provided for the convenience of the reader.

### 3.2 Hyperlink Prediction for Undirected Hypergraphs: UGCN

Due to the particularity of graph structure data, the number of neighboring nodes of each node is not necessarily equal, it is not feasible to extract the features of nodes directly using convolution neural network model (CNN) [43], because operations such as convolution and pooling can only act on regular grids. GCN [34] is a model that can perform convolutional neural networks on graph structure data, and it extends convolution operation to irregular or non-Euclidean graph structure data such as metabolic networks, social networks, telecommunication networks, which can be described using graphs. Considering that the GCN model can extract graph data information effectively, an undirected hyperlink prediction method based on the GCN model is proposed, named UGCN for short. A schematic description of the UGCN framework is presented in Fig. 3.

The layer-wise information propagation rule of GCN is

$$H^{(l+1)} = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} W^{(l)}). \tag{1}$$



**Fig. 3** Schematic description of UGCN framework

The feature information of the  $(l + 1)$ th layer of each node in the graph is obtained by aggregating the information of the node itself and its neighboring nodes in the  $l$ th layer.  $H^{(l)}$  is the activation matrix in the  $l$ th layer, if  $l = 0$ ,  $H^{(0)}$  represents the initial node feature matrix  $X$  of the graph.  $W^{(l)}$  is the specific weight matrix of the  $l$  layer.  $\tilde{A} = A + I_N$ , where  $A$  is an adjacency matrix which can be used to represent graph data,  $I_N$  is a identify matrix which denotes self-loops of nodes, therefore,  $\tilde{A}$  is the adjacency matrix after adding self-loops to all nodes. In the process of information propagation between layers, the neighboring information of the node can be obtained through the adjacency matrix. However, if we only use the adjacency matrix to transmit information, the information carried by the node itself will be lost. Therefore, it is necessary to retain the information of the nodes themselves by manually adding a self-loop to each node.  $\tilde{D}$  is a diagonal degree matrix, and  $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$  denotes the degree of nodes after adding self-loops. In the process of information propagation between layers, since matrix transformation needs to be continuously done, the node feature information matrix of the next layer obtained by aggregating the current information will continuously become larger. As the number of layers increases, the size of the output node feature information matrix will differ greatly from the initial feature matrix size, so this problem is avoided by normalizing the adjacency matrix in the process of information propagation.  $\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$  is the symmetric normalization of the adjacency matrix.  $\sigma(\cdot)$  is a non-linear transformation activation function, such as ReLU function.

The feature of the node can be continuously propagated by multiple GCN layers. However, Since the receptive field will increase with the update of node features, if the network is deepened continuously and the number of GCN layers is too much, then each node will be affected by irrelevant nodes, and the obtained effect will decrease instead. Therefore, a two-layer GCN is used to propagate the node feature as follows:

$$Z = f(X, A) = (\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} \text{ReLU}(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} X W^{(0)}) W^{(1)}), \tag{2}$$

where  $X = \{x_1, x_2, \dots, x_n\} \in \mathbb{R}^{n \times d}$  is the initial node feature vector,  $n$  is the number of nodes,  $d$  is the dimension of node features,  $A$  is the adjacency matrix, and  $\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$  is the normalized adjacency matrix. Let  $\bar{A} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$ , then Eq. (2) is changed to

$$Z = f(X, A) = (\bar{A} \text{ReLU}(\bar{A} X W^{(0)}) W^{(1)}). \tag{3}$$

Suppose  $u$  is a node in hyperedge  $e$ , according to the above Eq. (3), node  $u$  not only aggregates the information from the neighboring nodes of node  $u$  but also the information originally carried by the node  $u$ , then the updated feature information of node  $u$  is as follows.

$$x_u = (\bar{A} \text{ReLU}(\bar{A} x_u^{(0)} W^{(0)}) W^{(1)}), \tag{4}$$

where  $x_u^{(0)}$  is the initial feature vector of node  $u$ ,  $x_u$  is the updated feature vector of node  $u$  after it passes through two GCN layers. After that, we preserve the higher-order relationships between nodes by aggregating the representation information of hyperedge  $e$ . The representation information of the hyperedge  $e$  is aggregated by using average function, that is the feature information of all nodes in the hyperedge  $e$  as follows:

$$h^e = \frac{1}{|e|} \sum_{u \in e} x_u. \tag{5}$$

Considering that in some networks, the more similar the characteristics of the nodes, the higher the possibility of forming hyperedges. Such as the research collaboration network, if some authors have similar research directions, or are members of the same laboratory, or have the same collaborators, then these authors will be more likely to publish papers together. However, in some networks, the more different the characteristics of the nodes, the higher the possibility of forming hyperedges, such as the chemical reaction network in Fig. 2, the chemical reaction between  $\text{Cu}(\text{OH})_2$  and  $\text{HCl}$ , where  $\text{Cu}(\text{OH})_2$  is a base and  $\text{HCl}$  is an acid, and acids and bases are two kinds chemical substances with opposite properties. Therefore, Yadati et al. [40] propose a maxmin function to aggregate the representation information of the hyperedge  $e$ , that is

$$h^e = \text{maxmin}\{x_u : u \in e\} = (\max_{i \in e} x_{il} - \min_{j \in e} x_{jl}), \tag{6}$$

where  $l = 1, 2, \dots, d$ ,  $d$  is the number of feature dimensions,  $\max_{i \in e} x_{il}$  is the maximum feature value of the nodes within the hyperedge  $e$  at the  $l$ th feature dimension,  $\min_{j \in e} x_{jl}$  is the minimum feature value of the nodes within the hyperedge  $e$  at the  $l$ th feature dimension, and  $\max_{i \in e} x_{il} - \min_{j \in e} x_{jl}$  denotes the difference between the maximum and minimum feature values of the nodes within the hyperedge  $e$  at the  $l$ th feature dimension.

Then, a fully connected layer and an activation layer are introduced to score the hyperedges. If the mean function will be used to aggregate the representation information of hyperedges, the scoring function is as follows:

$$s_e = \sigma(W \cdot h^e + b) = \sigma\left(W \cdot \frac{1}{|e|} \sum_{u \in e} x_u + b\right). \tag{7}$$

If the maxmin function will be used to aggregate the representation information of hyperedges, the scoring function is as follows:

$$s_e = \sigma(W \cdot \text{maxmin}\{x_u : u \in e\} + b), \tag{8}$$

where  $W$  and  $b$  are the weight matrix and bias of the fully connected layer, respectively, which are the hyperparameters need to be learned, and  $\sigma$  is the sigmoid activation function. Without loss of generality, it is possible to score the hyperedges using the above equation.

In order to ensure the accuracy of prediction, we need to ensure that the scores of hyperedges that can be observed in the hypergraph are more likely to be higher than the unobserved hyperedges in the hypergraph. Therefore, we can achieve the goal by optimizing the following objective function:

$$L_U = \frac{1}{|E|} \sum_{e \in E} g \left( \left( \frac{1}{|F|} \sum_{f \in F} s_f \right) - s_e \right), \tag{9}$$

where  $E$  is the set of observed hyperedges in the hypergraph,  $F$  is the set of hyperedges sampled from the set of unobserved hyperedges  $E'$  in the hypergraph,  $|E'| = 2^n - |E|$ , and  $g(x) = \log(1 + e^x)$ .

We continuously train the model and optimize the objective function so that the scores of the positive samples become higher and higher and the scores of the negative samples become lower and lower, thereby maximizing the difference in scores between the positive and negative samples.

### 3.3 Hyperlink Prediction for Undirected Hypergraphs: UGAT

GCN can effectively handle graph data information, but it also has some problems. For example, it cannot allocate the weights of neighboring nodes according to the node characteristic attributes, which lead to the GCN's ability to capture the spatial information correlation is poor. GAT can allocate the weights of neighboring nodes according to the importance of associated nodes. Therefore, the graph attention network framework (GAT) [35] is introduced to propose an undirected hyperlink prediction method based

on the GAT model, UGAT for short. A schematic description of the UGAT framework is presented in Fig. 4.

Suppose the input node feature vector is  $X = \{x_1, x_2, \dots, x_n\} \in \mathbb{R}^{n \times d}$ , where  $n$  is the number of nodes,  $d$  is the number of node feature dimensions, then the updated node feature vector after a GAT layer is  $X' = \{x'_1, x'_2, \dots, x'_n\} \in \mathbb{R}^{n \times d'}$ . Firstly, a weight matrix  $W$  is used to act on each node, and then the self-attention mechanism is used to obtain the importance of neighboring nodes to the current center node, that is

$$e_{ij} = a(Wx_i, Wx_j), \tag{10}$$

where  $e_{ij}$  is the importance of node  $j$  to node  $i$ , and  $a$  is attention mechanism.

There are many ways to replace  $a$ . Here a single-layer feedforward neural network with parameter  $\vec{a} \in \mathbb{R}^{2d'}$  is used to replace  $a$ , and then the LeakyReLU activation function is used to do nonlinearization. Hence  $e_{ij}$  is denoted as

$$e_{ij} = \text{LeakyReLU}(\vec{a}^T [Wx_i || Wx_j]), \tag{11}$$

where  $\vec{a}^T$  is the transposition of  $\vec{a}$ , and  $||$  denotes the concatenation operation.

Then, the attention coefficient is obtained by normalizing the importance of neighboring nodes to the center node through the softmax function, we have

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(\text{LeakyReLU}(\vec{a}^T [Wx_i || Wx_j]))}{\sum_{k \in \tau_i} \exp(\text{LeakyReLU}(\vec{a}^T [Wx_i || Wx_k]))}, \tag{12}$$

where  $j \in \tau_i$ , and  $\tau_i$  is the set of nodes which adjacent to node  $i$ .

Finally, the attention coefficient of each neighbor node of the central node is multiplied with its node feature vector, then the product results are summed, and the final updated feature vector of the central node is

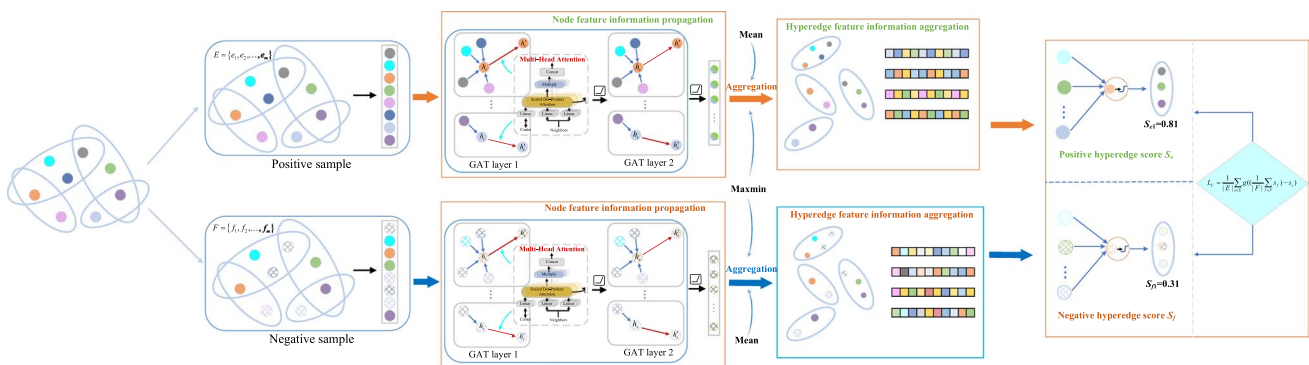


Fig. 4 Schematic description of UGAT framework

$$x'_i = \sigma \left( \sum_{j \in \tau_i} \alpha_{ij} W x_j \right). \tag{13}$$

In order to improve the generalization ability of the attention mechanism, the multi-head graph attention mechanism is proposed, that is,  $K$  sets of independent single-head attention layers are concatenated together, then we have

$$x'_i = || \sigma \left( \sum_{j \in \tau_i} \alpha_{ij}^k W^k x_j \right) \tag{14}$$

where  $||$  denotes the concatenation operation.

A two-layer GAT is used to propagate node feature information, the first layer is a 4-head GAT, and the second layer is a single-head GAT.

The node feature information propagation rule of the first GAT layer is

$$x_i^{(1)} = || \sigma \left( \sum_{j \in \tau_i} \alpha_{ij}^k W^k x_j^{(0)} \right), \tag{15}$$

where  $x_i^{(1)}$  is the updated node feature through the first GAT layer,  $\sigma$  is non-linear activation function and an exponential linear unit (ELU) activation function is used here,  $\alpha_{ij}^k W^k$  is the attention coefficient of the  $k$ th head GAT in the first layer, and  $x_j^{(0)}$  is the initial node feature vector of any neighboring node of node  $i$  in the graph. Note that in order to retain the features of node  $i$  while obtaining the features of neighboring nodes, it is necessary to manually add a self-loop to each node to retain the information of the node itself. Therefore, when aggregating the feature information of nodes of the center node, the information of the center node itself is also included.

The node feature information propagation rule of the second GAT layer is

$$x_i^{(2)} = \sigma \left( \sum_{j \in \tau_i} \alpha_{ij}^{(1)} W^{(1)} x_j^{(1)} \right) = \sigma \left( \sum_{j \in \tau_i} \alpha_{ij}^{(1)} W^{(1)} \left( || \sigma \left( \sum_{j \in \tau_i} \alpha_{ij}^k W^k x_j^{(0)} \right) \right) \right), \tag{16}$$

where  $x_i^{(2)}$  is the updated node feature through the two-layer GAT,  $\sigma$  is the ELU activation function, and  $\alpha_{ij}^{(1)} W^{(1)}$  is the attention coefficient of the second GAT layer.

Then, the higher-order relationships between nodes are preserved by aggregating the representation information of the hyperedge  $e$  in the same way as the aggregation in Sect. 3.2.

The main pseudo-code of the UGCN and UGAT algorithms is provided to show more detail.

---

**Algorithm 1: UGCN and UGAT**

---

Input:  
 Hypergraph datasets:  $G$   
 Initial node feature of  $u$  :  $x_u^{(0)}$   
 Adjacency matrix after adding self-loops:  $A$   
 Degree matrix after adding self-loops:  $D$   
 Weight matrix of layer 1:  $W^{(0)}$   
 Weight matrix of layer 2:  $W^{(1)}$   
 Attention coefficient:  $\alpha$   
 Bias:  $b$   
 The set of neighboring nodes of  $i$ :  $\tau_i$

Output:  
 The score of hyperedge  $e$  :  $s_e$

---

Content:  
 1 Update the node feature  $x_u$  :  
 if model\_name == 'GCN' then  
 $x_u \leftarrow (D^{-1/2} A D^{-1/2} \text{ReLU}(D^{-1/2} A D^{-1/2} x_u^{(0)} W^{(0)})) W^{(1)}$   
 else if model\_name == 'GAT' then  
 $x_u \leftarrow \text{ELU} \left( \sum_{j \in \tau_i} \alpha_{ij}^{(1)} W^{(1)} \left( || \sigma \left( \sum_{j \in \tau_i} \alpha_{ij}^k W^k x_j^{(0)} \right) \right) \right)$   
 end if  
 2 Get the feature information of hyperedge  $e$  :  
 if aggregation\_way == 'mean' then  
 $h^e \leftarrow \frac{1}{|e|} \sum_{u \in e} x_u$   
 else if aggregation\_way == 'maxmin' then  
 $h^e \leftarrow \text{maxmin} \{ x_u : u \in e \} = (\max_{i \in e} x_{i_i} - \min_{j \in e} x_{j_l})$   
 end if  
 3 Get the score of hyperedge  $e$  :  
 $s_e \leftarrow \text{sigmoid}(W \cdot h^e + b)$

---

**3.4 Hyperlink Prediction for Directed Hypergraphs**

Directed hyperlink prediction not only needs to predict unobserved hyperlinks, but also needs to predict the direction of hyperlinks, that is, predict the head and tail of hyperlinks, which is a very difficult research problem. Therefore, the hyperlink prediction for directed hypergraph from the perspective of graph neural networks is explored in this article. Two kinds of directed hyperlink prediction algorithms are proposed, one is directed hyperlink prediction method based on GCN, which is called DGCN for short, and the other is directed hyperlink prediction method based on GAT, which is called DGAT for short. The difference between these two methods is only the pattern of obtaining the updated node feature information. One is to aggregate node feature information by using GCN, and the other is to aggregate node feature information by using GAT. Therefore, when introducing the two

directed hyperlink prediction frameworks, we will introduce them together and do not introduce the two methods separately. As shown in Fig. 5, which is a schematic description of the hyperlink prediction framework for directed hypergraphs.

Firstly, we utilize the proposed UGCN and UGAT in the previous two sections to propagate node features and aggregate hyperedge information, where the scoring function [40] used for a directed hyperlink  $e = (\vec{e}^+, \vec{e}^-)$  is

$$s_e = \sigma(W \cdot \maxmin_D\{x_u : u \in e\} + b), \tag{17}$$

where  $\maxmin_D\{x_u : u \in e\} = (\max_{i \in \vec{e}^+} x_{i_l} - \min_{j \in \vec{e}^-} x_{j_l})$  if  $(\max_{i \in \vec{e}^+} x_{i_l} - \min_{j \in \vec{e}^-} x_{j_l}) > 0$ , else  $\maxmin_D\{x_u : u \in e\} = 0$ ,  $l = 1, 2, \dots, d$ , and  $d$  is the feature vector dimension.

Let  $e_i$  and  $e_j$  are two hyperlinks, this layer is utilized to determine the direction of the hyperlinks. In other words, we need to prediction the head and tail of the directed hyperlinks. Let  $o$  and  $r$  are the aggregated representation information of  $e_i$  and  $e_j$ . The mean function is used to aggregate the feature information of hyperlinks  $e_i$  and  $e_j$  by averaging the feature information of all nodes in hyperlinks  $e_i$  and  $e_j$ , respectively, that is

$$o = \frac{1}{|e_i|} \sum_{u \in e_i} x_u, \quad r = \frac{1}{|e_j|} \sum_{u \in e_j} x_u. \tag{18}$$

Then, a fully connected layer and an activation layer are introduced to score the direction of the hyperedges, that is

$$D_{(o,r)} = \sigma(o^T W_D \cdot r + b_D), \tag{19}$$

where  $\sigma$  is a sigmoid activation function,  $W_D$  and  $b_D$  are the weight matrix and bias of the linear layer, respectively. Without loss of generality, we need to ensure that the scores of the direction of observed directed hyperedges in the hypergraph are more likely to be higher than the direction of the unobserved directed hyperedges in the hypergraph. Therefore, we can achieve the goal by optimizing the following objective function:

$$L_D = \frac{1}{|E|} \sum_{(p,q) \in E} g\left(\left(\frac{1}{|F|} \sum_{(o,r) \in F} D_{(o,r)}\right) - D_{(p,q)}\right), \tag{20}$$

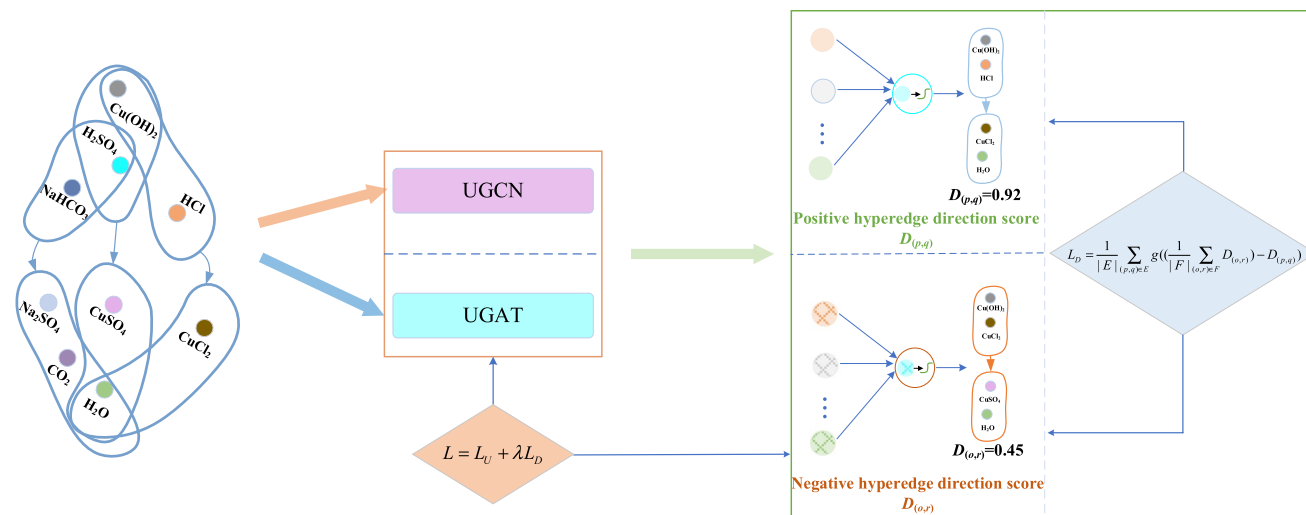
where  $E$  is the set of observed hyperedges in the hypergraph,  $F$  is the set of hyperedges sampled from the set of unobserved hyperedges  $E'$  in the hypergraph,  $|E'| = 2^n - |E|$ , and  $g(x) = \log(1 + e^x)$ .

In the process of predicting directed hyperlinks, we need to optimize not only the objective function (9) of the hyperlink prediction layer, but also the objective function (20) of the direction prediction layer, so that the final objective function of directed hyperlink prediction is

$$L = L_U + \lambda L_D, \tag{21}$$

where  $\lambda$  is the parameter that needs to be learned.

The main pseudo-code of the DGCN and DGAT algorithms is provided to show more detail.



**Fig. 5** A schematic description of the hyperlink prediction framework for directed hypergraphs. The figure shows two hyperlink prediction algorithms for directed hypergraphs, DGCN and DGAT, respectively.

The difference is the way of propagating node feature information, but the purpose is the same, which is to update the features of the nodes within the hyperedges



**Algorithm 2: DGCN and DGAT**

Input:  
 Hypergraph datasets:  $G$   
 Initial node feature of  $u$ :  $x_u^{(0)}$   
 Adjacency matrix after adding self-loops:  $A$   
 Degree matrix after adding self-loops:  $D$   
 Weight matrix of layer 1:  $W^{(0)}$   
 Weight matrix of layer 2:  $W^{(1)}$   
 Weight matrix of hyperedge direction:  $W_D$   
 Attention coefficient:  $\alpha$   
 Bias:  $b, b_D$   
 The set of neighboring nodes of  $i$ :  $\tau_i$

Output:  
 The score of hyperedge  $e$ :  $s_e$   
 The direction score:  $s_D$

---

Content:  
 1 Update the node feature  $x_u$  :  
 if model\_name == 'GCN' then  

$$x_u \leftarrow (D^{-1/2} A D^{-1/2} \text{ReLU}(D^{-1/2} A D^{-1/2} x_u^{(0)} W^{(0)})) W^{(1)}$$
  
 else if model\_name == 'GAT' then  

$$x_u \leftarrow \text{ELU}(\sum_{j \in \tau_i} \alpha_{ij}^{(1)} W^{(1)} (\| \sigma(\sum_{k=1}^4 \sigma(\sum_{j \in \tau_i} \alpha_{ij}^k W^k x_j^{(0)})))$$
  
 end if  
 2 Get the feature information of hyperedge  $e$  :  
 if aggregation\_way == 'mean' then  

$$h^e \leftarrow \frac{1}{|e|} \sum_{u \in e} x_u$$
  
 else if aggregation\_way == 'maxmin' then  

$$h^e \leftarrow \text{maxmin}\{x_u : u \in e\} = (\max_{i \in e} x_{ii} - \min_{j \in e} x_{jj})$$
  
 end if  
 3 Get the score of hyperedge  $e$  :  

$$s_e \leftarrow \text{sigmoid}(W \cdot h^e + b)$$
  
 4 Get the direction score:  

$$o \leftarrow \frac{1}{|e_i|} \sum_{u \in e_i} x_u,$$
  

$$r \leftarrow \frac{1}{|e_j|} \sum_{u \in e_j} x_u$$
  

$$s_D \leftarrow \text{sigmoid}(o^T W_D \cdot r + b_D)$$

---

Hence, hyperlink prediction for undirected hypergraphs and directed hypergraphs can be realized by utilizing the content of this chapter. In addition, we will apply the proposed methods to real datasets to verify the effectiveness of the methods in the next chapter.

## 4 Experimental Results

### 4.1 Datasets

In order to verify the effectiveness of the proposed graph neural network-based undirected and directed hyperlink prediction, experimental validation on five real hypergraph datasets, which are iJO1366, iAF1260b, USPTO, DBLP, and Reverb45k [40], is performed. iJO1366 and iAF1260b are metabolic reaction datasets, USPTO is an organic chemical reaction dataset, DBLP is an undirected scientific collaboration dataset, and Reverb45k is a directed knowledge graph dataset. Note that USPTO, iJO1366 and iAF1260b are used not only for undirected hyperlink prediction, but also for directed hyperlink prediction. The details of the five real hypergraph datasets are presented in Table 2, where td denotes the dataset type, th denotes the hyperedge type, n denotes the number of nodes, m denotes the number of hyperedges, and d denotes the dimension of node features.

### 4.2 Experimental Setup

Except for the experiments on the Reverb45k and USPTO datasets were performed in a workstation equipped with a 24-core Intel Xeon(R) Gold 5118 CPU @2.30 GHz processor and 64 GB RAM, the rest of experiments were run in a desktop pc equipped with an 8-core Intel Xeon(R) E-2244G CPU @3.80 GHz processor and 16 GB RAM. All the program code is run in Pycharm 2021.2.1.

The AUC and Recall@k (k is half of the number of unobserved hyperedges in the dataset) are used for assessing the prediction accuracy of the proposed hyperlink prediction methods. All experimental results are obtained with training rate of 20% and test rate of 80% for the datasets. Due to the large size of the Reverb45k dataset, the results on this dataset are obtained by randomly dividing this dataset

**Table 2** The details of the five real hypergraph datasets

Datasets	td	th	n	m	d
iJO1366	Metabolic reactions	Directed, undirected	1805	2253	26
iAF1260b	Metabolic reactions	Directed, undirected	1668	2084	26
USPTO	Organic reactions	Directed, undirected	16,293	11,433	298
DBLP	Co-authorship	Undirected	20,685	30,956	3763
Reverb45k	Knowledge graph	Directed	28,798	66,914	382

once, and the results on the other datasets are obtained by randomly dividing the datasets three times.

We train the proposed methods for 50 iterations using Adam optimizer [44] with a learning rate of 0.0001 on all datasets. All weights are initialized using Glorot initialization [45]. The hidden size of each layer is 512. In addition, in the two-layer GCN model, dropout [46] = 0.5 is applied to the second layer's inputs. In the two-layer GAT model, dropout = 0.5 is applied to both layers' inputs and the process of normalized attention coefficients. The first layer consists of  $K=4$  attention heads, followed by an exponential linear unit (ELU) [47]. The second layer consists of a single attention head, followed by an ELU.

To avoid results deviation due to different size of the hyperlinks, for each observed hyperlink  $e \in E$ , the same number of nodes are sampled as the negative hyperlink  $f \in F$ , where half of the nodes in the negative hyperlink  $f$  are sampled from the observed hyperlink  $e$ , and the other half are sampled from node set  $V - e$  outside of the hyperlink  $e$ .

### 4.3 Experimental Results on Undirected Hyperlink Prediction

#### 4.3.1 Baselines

To verify the feasibility of the proposed HGCA-UGCN and HGCA-UGAT undirected hyperlink prediction methods, the prediction performance of the following baselines applied to the hyperlink prediction task is compared.

**HGNN [48]:** Hypergraph neural network (HGNN) is a typical model that processes hypergraph data information after transforming hypergraphs into ordinary graphs. When the model is used for undirected hyperlink prediction task, its information propagation equation

$$X^{(l+1)} = \sigma(D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2}) X^{(l)} \Theta^{(l)}, \quad (22)$$

and a linear layer are used to update the node features within the hyperedges.

**Hyper-SAGNN [49]:** The model utilizes the self-attention to aggregate the node features within the hyperedges. It can be directly used for undirected hyperlink prediction.

**Node2Vec-mean:** The Node2Vec [50] method is used to obtain the representation of the nodes in the hypergraph, and then the mean function is used to aggregate the information of the hyperedges.

**Node2Vec-maxmin:** The Node2Vec [50] method is used to obtain the representation of the nodes in the hypergraph, and

then the maxmin function is used to aggregate the information of the hyperedges.

**CMM [26]:** The method predicts hyperlinks by alternately performing nonnegative matrix factorization and least square matching.

**SHC [41]:** The method transforms the hypergraph into a dual hypergraph, where the nodes in the original hypergraph become hyperedges and the hyperedges become nodes in its dual hypergraph, so that the hyperlink prediction becomes a problem of node classification.

**NHP-U-mean [40]:** This baseline simply uses the information propagation equation of GCN, that is Eq. (1), to update the features of the nodes, and then uses the mean function to aggregate the information of hyperedges, and finally scores the hyperedges.

**NHP-U-maxmin [40]:** This baseline simply uses the information propagation equation of GCN, that is Eq. (1), to update the features of the nodes, and then uses the maxmin function to aggregate the information of hyperedges, and finally scores the hyperedges.

#### 4.3.2 Results

Tables 3 and 4 provide the results of the undirected hyperlink prediction of the proposed two models, UGCN and UGAT, with eight baselines. Table 3 presents the comparison of AUC and Recall@k values for all models on iJO1366, iAF1260b and USPTO datasets. Table 4 presents the comparison of AUC and Recall@k values for all models on the DBLP dataset. Note that -mean and -maxmin denote the information of the hyperedges aggregated with the mean and maxmin functions, respectively.

As shown in Table 3, we find that the prediction performance of the two proposed undirected hyperlink prediction methods performs better on all three datasets. Compared with the eight baseline models, the two proposed methods have the best hyperlink prediction performance, where the AUC value increases by a minimum of 10% and a maximum of 40%, and the Recall@k value increases by a minimum of 3% and a maximum of 25%. CMM and SHC are hyperlink prediction methods based on candidate hyperlink sets. We find that the proposed methods can also predict hyperlinks efficiently without using candidate hyperlink sets from the experimental results. NHP-U-mean and NHP-U-maxmin use the information propagation equation of GCN to update the features of nodes, but the single layer GCN equation is stiffly used in the two methods in the information propagation process, which does not achieve the effect of flexibly using GCN ideas to transfer the feature information of nodes and is quite different

**Table 3** Results comparison of undirected hyperlink prediction on iJO1366, iAF1260b and USPTO datasets

Datasets Models	iJO1366		iAF1260b		USPTO	
	AUC	Recall@k	AUC	Recall@k	AUC	Recall@k
HGNN	0.61 ± 0.03	0.32 ± 0.02	0.62 ± 0.02	0.27 ± 0.04	0.69 ± 0.02	0.26 ± 0.01
Hyper-SAGNN	0.56 ± 0.03	0.30 ± 0.04	0.60 ± 0.01	0.26 ± 0.01	0.67 ± 0.01	0.27 ± 0.01
Node2Vec-mean	0.54 ± 0.02	0.27 ± 0.03	0.58 ± 0.03	0.24 ± 0.04	0.58 ± 0.03	0.23 ± 0.02
Node2Vec-maxmin	0.60 ± 0.02	0.29 ± 0.03	0.61 ± 0.05	0.28 ± 0.03	0.68 ± 0.03	0.26 ± 0.01
CMM	0.64 ± 0.03	0.35 ± 0.10	0.64 ± 0.04	0.30 ± 0.14	0.68 ± 0.01	0.37 ± 0.01
SHC	0.64 ± 0.01	0.33 ± 0.02	0.65 ± 0.01	0.31 ± 0.02	0.56 ± 0.01	0.22 ± 0.01
NHP-U-mean	0.61 ± 0.02	0.29 ± 0.02	0.60 ± 0.04	0.28 ± 0.06	0.65 ± 0.02	0.33 ± 0.05
NHP-U-maxmin	0.63 ± 0.02	0.32 ± 0.02	0.64 ± 0.03	0.31 ± 0.03	0.74 ± 0.02	0.37 ± 0.02
UGCN-mean	0.96 ± 0.02	0.49 ± 0.01	0.94 ± 0.03	0.48 ± 0.02	0.95 ± 0.01	0.48 ± 0.02
UGCN-maxmin	0.84 ± 0.02	0.425 ± 0.028	0.82 ± 0.04	0.43 ± 0.04	0.99 ± 0.01	0.49 ± 0.01
UGAT-mean	0.96 ± 0.03	0.48 ± 0.02	0.96 ± 0.03	0.49 ± 0.01	0.84 ± 0.02	0.46 ± 0.01
UGAT-maxmin	0.82 ± 0.02	0.40 ± 0.04	0.79 ± 0.03	0.39 ± 0.01	0.99 ± 0.01	0.49 ± 0.01

**Table 4** Results comparison of undirected hyperlink prediction on DBLP dataset

Datasets Models	DBLP	
	AUC	Recall@k
HGNN	0.65 ± 0.04	0.33 ± 0.03
Hyper-SAGNN	0.65 ± 0.01	0.34 ± 0.04
Node2Vec-mean	0.58 ± 0.01	0.31 ± 0.03
Node2Vec-maxmin	0.63 ± 0.04	0.35 ± 0.06
CMM	\	\
SHC	\	\
NHP-U-mean	0.56 ± 0.02	0.28 ± 0.04
NHP-U-maxmin	0.69 ± 0.02	0.38 ± 0.05
UGCN-mean	0.99 ± 0.01	0.49 ± 0.01
UGCN-maxmin	0.98 ± 0.01	0.49 ± 0.01
UGAT-mean	0.99 ± 0.01	0.49 ± 0.01
UGAT-maxmin	0.76 ± 0.021	0.39 ± 0.02

from the convolution idea of GCN in essence. The convolution idea of GCN is used flexibly, so the performance of hyperlink prediction is greatly improved in this article. As shown in Table 4, it indicates that the hyperlink prediction performance of the proposed two methods is also better in the scientific collaboration hypergraph, which indicates that the proposed two methods can predict unseen hyperlinks effectively, where the AUC value increases by a minimum of 7% and a maximum of 43%, and the Recall@k value increases by a minimum of 1% and a maximum of 21%.

## 4.4 Experimental Results on Directed Hyperlink Prediction

### 4.4.1 Baselines

To verify the feasibility of the proposed DGCN and DGAT directed hyperlink prediction methods, we compare the prediction performance of the four baselines introduced in Sect. 4.4.1, which are HGNN, Hyper-SAGNN, Node2Vec-mean, and Node2Vec-maxmin, and the following two baselines.

**NHP-D-mean** [40]: This baseline simply uses the information propagation equation of GCN, that is Eq. (1), to update the features of the nodes, and then uses the mean function to aggregate the information of hyperedges, and finally scores the hyperedges and the direction of the hyperedges.

**NHP-D-maxmin<sub>D</sub>** [40]: This baseline simply uses the information propagation equation of GCN, that is Eq. (1), to update the features of the nodes, and then uses the maxmin function to aggregate the information of hyperedges, and finally scores the hyperedges and the direction of the hyperedges.

### 4.4.2 Results

Tables 5 and 6 provide the results of the two proposed methods, DGCN and DGAT, with six Baselines for directed hyperlink prediction. Table 5 presents the comparison of AUC and Recall@k values of all methods on iJO1366, iAF1260b and USPTO datasets. Table 6 presents the comparison of AUC and Recall@k values of all methods on the Reverb45k dataset.

As shown in Table 5, we find that the two proposed directed hyperlink prediction methods have better prediction

**Table 5** Results comparison of directed hyperlink prediction on iJO1366, iAF1260b and USPTO datasets

Datasets Models	iJO1366		iAF1260b		USPTO	
	AUC	Recall@k	AUC	Recall@k	AUC	Recall@k
HGNN	0.53 ± 0.02	0.25 ± 0.01	0.55 ± 0.02	0.23 ± 0.05	0.59 ± 0.01	0.22 ± 0.04
Hyper-SAGNN	0.51 ± 0.02	0.23 ± 0.02	0.53 ± 0.02	0.20 ± 0.03	0.56 ± 0.02	0.21 ± 0.03
Node2Vec-mean	0.51 ± 0.03	0.22 ± 0.04	0.52 ± 0.03	0.18 ± 0.03	0.56 ± 0.04	0.17 ± 0.04
Node2Vec-maxmin <sub>D</sub>	0.52 ± 0.01	0.24 ± 0.01	0.53 ± 0.01	0.21 ± 0.01	0.58 ± 0.02	0.24 ± 0.03
NHP-D-mean	0.54 ± 0.02	0.26 ± 0.02	0.55 ± 0.01	0.23 ± 0.05	0.60 ± 0.03	0.18 ± 0.03
NHP-D-maxmin <sub>D</sub>	0.56 ± 0.01	0.28 ± 0.03	0.58 ± 0.02	0.26 ± 0.04	0.63 ± 0.02	0.25 ± 0.04
DGCN-mean	0.93 ± 0.03	0.48 ± 0.02	0.91 ± 0.01	0.48 ± 0.02	0.96 ± 0.02	0.49 ± 0.01
DGCN-maxmin <sub>D</sub>	0.93 ± 0.03	0.48 ± 0.02	0.92 ± 0.01	0.48 ± 0.02	0.96 ± 0.02	0.49 ± 0.01
DGAT-mean	0.97 ± 0.02	0.49 ± 0.01	0.96 ± 0.02	0.49 ± 0.01	0.92 ± 0.01	0.48 ± 0.02
DGAT-maxmin <sub>D</sub>	0.97 ± 0.02	0.49 ± 0.01	0.96 ± 0.02	0.49 ± 0.01	0.93 ± 0.01	0.48 ± 0.02

**Table 6** Results comparison of directed hyperlink prediction on Reverb45k dataset

Datasets Models	Reverb45k	
	AUC	Recall@k
HGNN	0.76 ± 0.03	0.40 ± 0.05
Hyper-SAGNN	0.74 ± 0.03	0.39 ± 0.03
Node2Vec-mean	0.65 ± 0.02	0.41 ± 0.05
Node2Vec-maxmin <sub>D</sub>	0.75 ± 0.01	0.40 ± 0.04
NHP-D-mean	0.83 ± 0.03	0.40 ± 0.03
NHP-D-maxmin <sub>D</sub>	0.85 ± 0.04	0.43 ± 0.03
DGCN-mean	1	0.5
DGCN-maxmin <sub>D</sub>	1	0.5

performance on all three datasets. Compared with the six baseline models, the two proposed methods have the best directed hyperlink prediction performance, where the AUC value increases by a minimum of 29% and a maximum of 46%, and the Recall@k value increases by a minimum of 2% and a maximum of 32%. On all three datasets, most of the AUC values of these six baseline models are lower than 0.6, and the Recall@k values are lower than 0.3, while the AUC values of the proposed directed hyperlink prediction are more than 0.9, and the Recall@k values are more than 0.4. As shown in Table 6, it indicates that the hyperlink prediction performance of the two proposed directed hyperlink prediction methods also performs better in knowledge hypergraphs, where the AUC value increases by a minimum of 15% and a maximum of 35%, and the Recall@k value increases by a minimum of 7% and a maximum of 11%.

It can be found from Tables 2, 3, 4, 5 and 6 that the prediction results of the two proposed graph neural network-based hyperlink prediction are compared with those of other hyperlink prediction methods [26, 40, 41, 48–50], and it can be found that the prediction results of the proposed methods are better than those of the baselines on the undirected hyperlink prediction task and the directed hyperlink prediction task.

GCN is a spectral domain-based GNNs, GAT is a spatial domain-based GNNs, and it can adjust neighbor weights according to the characteristic attributes of nodes. In order to compare the performance between the spectral domain-based and spatial domain-based GNNs on the hyperlink prediction task, a hyperlink prediction method based on GAT is proposed. Through the experimental results, it can be found that to some extent, GCN performs better on the undirected hyperlink prediction task, and GAT performs better on the directed hyperlink prediction task.

#### 4.5 Hyperparameter Sensitivity Analysis

This section analyzes the influence of the hidden-layer size and the number of GAT heads on the experimental results in this article.

As shown in the line chart of Fig. 6, the influence of the hidden-layer size on the results of the proposed undirected and directed hyperlink prediction methods is shown on the iJO1366 and USPTO datasets, where the abscissa indicates the hidden-layer size and the ordinate indicates the AUC values. From Fig. 6, it can be found that although the AUC values decrease for a short duration when hidden size = 16, the overall trend is upward. In particular, when the hidden size = 512, the AUC values of DGCN-maxmin<sub>D</sub> and DGAT-maxmin<sub>D</sub> on the iJO1366 dataset reach steady state, and UGCN-maxmin, UGAT-maxmin and DGCN-maxmin<sub>D</sub> show a moderate uptrend on the USPTO dataset. Therefore, the experimental results when hidden size = 512 are selected as the final experimental results of the proposed methods.

As shown in the thermodynamic chart of Fig. 7, the influence of the number of GAT heads on the results of the proposed UGAT and DGAT methods on the iJO1366 and iAF1260b datasets are presented. As shown in Fig. 7a, on the iJO1366 dataset, except for UGAT-maxmin which has different AUC results when the number of heads is different, the AUC results of the other three methods are basically equal. As shown in Fig. 7b, it can be found that these

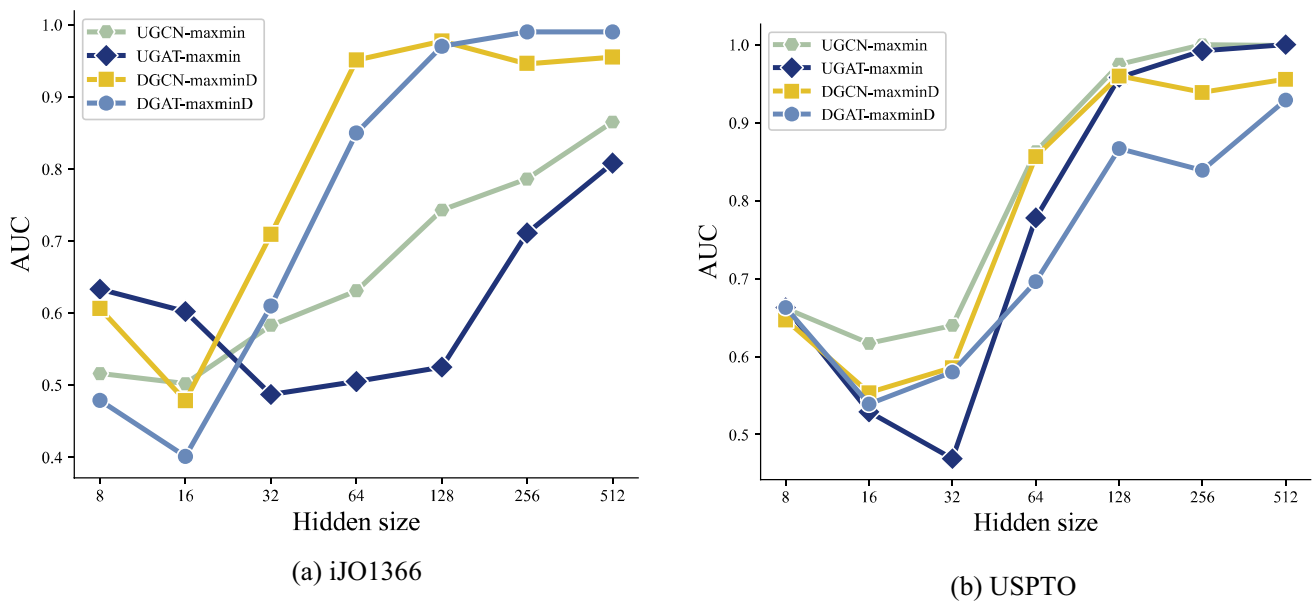


Fig. 6 Influence of the hidden-layer size on the experimental results

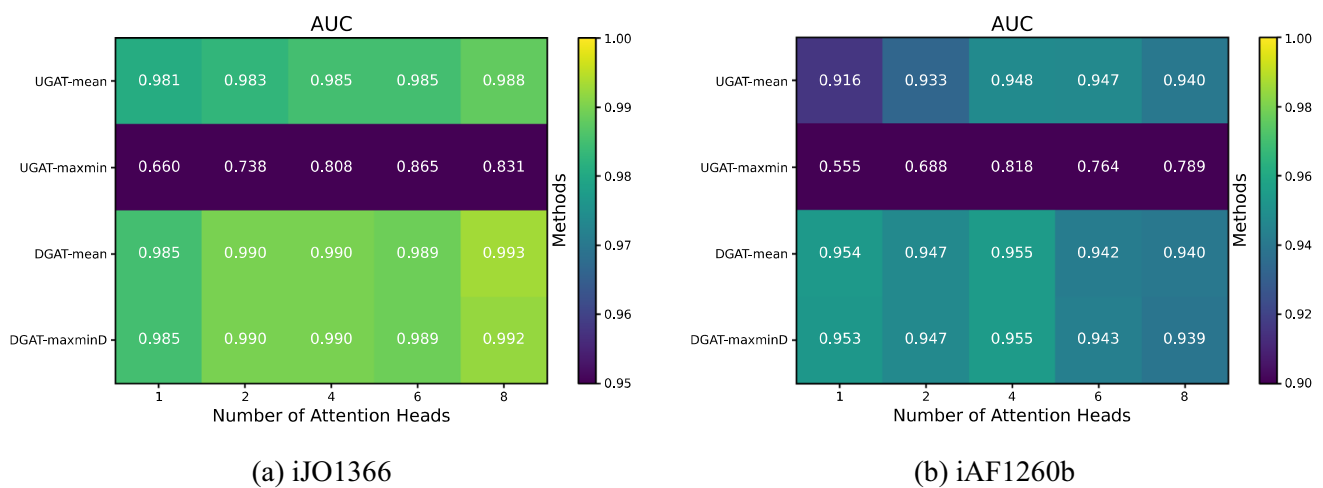


Fig. 7 Influence of the number of GAT heads on the results of the proposed UGAT and DGAT methods

methods have the best AUC results when the number of GAT heads is equal to 4. Therefore, based on the above comparison, the results when GAT heads are equal to 4 are comprehensively selected as the final results.

### 4.6 Time Complexity Analysis

As shown in Tables 7, 8, 9 and 10, the running time of the proposed methods are compared with the NHP algorithm on the iJO1366, iAF1260b and USPTO datasets. It can be found that the running time of the proposed UGCN and DGCN is lower than that of the NHP-U and NHP-D methods, whether using mean function or maxmin function to

Table 7 Comparison of running time of undirected hyperlink prediction using the mean function to aggregate hyperedge features

Datasets	iJO1366	iAF1260b	USPTO
NHP-U-mean	12.06	10.72	59.38
UGCN-mean	10.76	10.23	35.34
UGAT-mean	170.97	149.79	1756.82

aggregate hyperedge feature information, and by comparing the experimental results in Tables 3, 4, 5 and 6, the prediction results of the proposed UGCN and DGCN are better than other algorithms, which indicates that the proposed

**Table 8** Comparison of running time of undirected hyperlink prediction using the maxmin function to aggregate hyperedge features

Datasets	iJO1366	iAF1260b	USPTO
NHP-U-maxmin	179.84	161.42	724.81
UGCN-maxmin	149.54	137.52	819.86
UGAT-maxmin	317.57	278.29	2827.71

**Table 9** Comparison of running time of directed hyperlink prediction using the mean function to aggregate hyperedge features

Datasets	iJO1366	iAF1260b	USPTO
NHP-D-mean	23.88	22.45	120.51
DGCN-mean	13.54	12.69	34.65
DGAT-mean	178.87	162.68	1790.82

**Table 10** Comparison of running time of undirected hyperlink prediction using the maxmin<sub>D</sub> function to aggregate hyperedge features

Datasets	iJO1366	iAF1260b	USPTO
NHP-D-maxmin <sub>D</sub>	225.94	203.68	892.79
DGCN-maxmin <sub>D</sub>	181.77	168.58	987.13
DGAT-maxmin <sub>D</sub>	349.19	308.86	3040.89

UGCN and DGCN can predict hyperlinks efficiency with relatively low time complexity. Because the two-layer GAT is used in this article, and the first layer of GAT is 4-head, the running time of the proposed UGAT and DGAT methods is relatively higher than that of the NHP-U and NHP-D methods. However, compared with the NHP-U and NHP-D methods, the hyperlink prediction performance of the UGAT and DGAT methods has got a maximum improvement of 33%. Hence, it can be shown that the proposed methods can predict hyperlinks effectively.

## 5 Conclusion

Hyperlink prediction is an important topic to explore the evolution of higher-order structures. GNNs can effectively extract graph data information, so a GNNs-based hyperlink prediction framework is proposed from the perspective of graph neural networks. We utilize this framework to propose two hyperlink prediction methods, GCN-based hyperlink prediction and GAT-based hyperlink prediction, respectively, and extend these two methods to predict directed hyperlinks. In this article, firstly, two-layer GCN and two-layer GAT are used to propagate the feature vector information of nodes in hyperlinks. Secondly, the hyperedge information is aggregated utilizing the updated

node feature vector information. And then the scoring function is used to score the hyperlinks and the direction of the hyperlinks, Finally, through extensive experimental analysis, the any size of hyperlinks can be predicted by the proposed methods without the need of candidate hyperlink sets, and the hyperlinks can be efficiently predicted with relatively low time complexity. GCN is a spectral domain-based GNNs, GAT is a spatial domain-based GNNs. GCN and GAT focus on the influence of neighboring nodes on the central node, which can enhance the ability of models to perceive the neighborhood, thereby strengthening the joint spatial representation ability of models. It can also be found from the experimental results that GCN performs better on the undirected hyperlink prediction task, and GAT performs better on the directed hyperlink prediction task to some extent.

However, there are two problems with the two proposed methods. On the one hand, although the prediction performance of these two proposed methods is excellent, they lack interpretability, on the other hand, they do not consider the temporal information. Causal inference is a strong and effective tool in terms of interpretability. The hyperedges formed at different times are different in terms of the time series. Therefore, in the next work, we will tend to do research on explainable hyperlink prediction research while considering the factor of time.

**Author Contributions** YY: methodology, formal analysis, writing—original draft, writing—review and editing. ZY: formal analysis, writing—review and editing. HZ: funding acquisition, writing—review and editing. LM: writing—review and editing.

**Funding** This work is partially supported by the National Key Research and Development Program of China under Grant No.2020YFC1523300, the Construction of Innovation Platform Program of Qinghai Province of China under Grant No.2022-ZJ-T02.

**Data Availability** The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

## Declarations

**Conflict of Interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will

need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**(1), 47–97 (2002). <https://doi.org/10.1103/revmodphys.74.47>
- Getoor, L., Diehl, C.P.: Link mining. *ACM SIGKDD Explor. Newsl.* **7**(2), 3–12 (2005). <https://doi.org/10.1145/1117454.1117456>
- Lü, L.Y., Zhou, T.: Link prediction in complex networks: A survey. *Phys. A: Stat. Mech. Appl.* **390**(6), 1150–1170 (2011). <https://doi.org/10.1016/j.physa.2010.11.027>
- Zhou, T., Lee, Y.L., Wang, G.N.: Experimental analyses on 2-hop-based and 3-hop-based link prediction algorithms. *Phys. A: Stat. Mech. Appl.* **564**, 125532 (2021). <https://doi.org/10.1016/j.physa.2020.125532>
- Nasiri, E., Berahmand, K., Rostami, M., Dabiri, M.: A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding. *Comput. Biol. Med.* **137**, 104772 (2021). <https://doi.org/10.1016/j.combiomed.2021.104772>
- Zareie, A., Sakellariou, R.: Similarity-based link prediction in social networks using latent relationships between the users. *Sci. Rep.* **10**(1), 20137 (2020). <https://doi.org/10.1038/s41598-020-76799-4>
- Xie, X., Li, Y., Zhang, Z., Han, S., Pan, H.: A joint link prediction method for social network. In: ICYCSEE 2015. Springer, pp. 56–64 (2015). [https://doi.org/10.1007/978-3-662-46248-5\\_8](https://doi.org/10.1007/978-3-662-46248-5_8)
- Chuan, P.M., Son, L.H., Ali, M., Khang, T.D., Huong, L.T., Dey, N.: Link prediction in co-authorship networks based on hybrid content similarity metric. *Appl. Intell.* **48**(8), 2470–2486 (2017). <https://doi.org/10.1007/s10489-017-1086-x>
- Lande, D., Fu, M.L., Guo, W., Balagura, I., Gorbov, I., Yang, H.B.: Link prediction of scientific collaboration networks based on information retrieval. *World Wide Web.* **23**(4), 2239–2257 (2020). <https://doi.org/10.1007/s11280-019-00768-9>
- Zhou, T., Lü, L.Y., Zhang, Y.C.: Predicting missing links via local information. *Eur. Phys. J. B.* **71**(4), 623–630 (2009). <https://doi.org/10.1140/epjbe2009-00335-8>
- Wang, H.L., Yang, D., Nie, T.Z., Kou, Y.: Attributed heterogeneous information network embedding with self-attention mechanism for product recommendation. *J. Comput. Res. Dev.* **59**(7), 1509–1521 (2022). <https://doi.org/10.7544/issn1000-1239.2021016>
- Zhang, Z.P., Dong, M.X., Ota, K.R., Zhang, Y., Ren, Y.G.: LBCF: a link-based collaborative filtering for overfitting problem in recommender system. *IEEE Trans. Comput. Social Syst.* **8**(6), 1450–1464 (2021). <https://doi.org/10.1109/tcss.2021.3081424>
- Tao, H.W., Niu, X.X., Fu, L.Y., Yuan, S.Z., Wang, X., Zhang, J.X., Hu, Y.H.: DeepRS: A library of recommendation algorithms based on deep learning. *Int. J. Comput. Syst.* **15**, 45 (2022). <https://doi.org/10.1007/s44196-022-00102-8>
- Abbas, K., Abbasi, A., Dong, S., Niu, L., Yu, L.H., Chen, B.L., Cai, S.M., Hasan, Q.: Application of network link prediction in drug discovery. *BMC Bioinform.* **22**(1), 187 (2021). <https://doi.org/10.1186/s12859-021-04082-y>
- Liu, S., Tan, N.N., Yang, H., Lukač, N.: An intelligent question answering system of the liao dynasty based on knowledge graph. *Int. J. Comput. Intell. Syst.* **14**, 170 (2021). <https://doi.org/10.1007/s44196-021-00010-3>
- Kumar, R., Novak, Tomkins, J., A.: Structure and evolution of online social networks. In: KDD'06. Association for Computing Machinery, pp. 611–617 (2006). <https://doi.org/10.1145/1150402.1150476>
- Hu, F., Zhao, H.X., He, J.B., Li, F.X., Li, S.L., Zhang, Z.K.: An evolving model for hypergraph-structure-based scientific collaboration networks. *Acta Phys. Sin-Ch. Ed.* **62**(19), 198901 (2013). <https://doi.org/10.7498/aps.62.198901>
- Coley, C.W., Jin, W.G., Rogers, L., Jamison, T.F., Jaakkola, T.S., Green, W.H., Barzilay, R., Jensen, K.F.: A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem. Sci.* **10**(2), 370–377 (2019). <https://doi.org/10.1039/c8sc04228d>
- Harada, S., Akita, H., Tsubaki, M., Baba, Y., Takigawa, I., Yamashita, Y., Kashima, H.: Dual graph convolutional neural network for predicting chemical networks. *BMC Bioinform.* **21**(S3), 94 (2020). <https://doi.org/10.1186/s12859-020-3378-0>
- Nagurney, A., Dong, J.: *Supernetworks*. Edward Elgar Pub, Cheltenham (2002)
- Berge, C.: *Hypergraphs*. Elsevier, Amsterdam (1984)
- Estrada, E., Rodríguez-Velázquez, J.A.: Subgraph centrality and clustering in complex hyper-networks. *Phys. A: Stat. Mech. Appl.* **364**, 581–594 (2006). <https://doi.org/10.1016/j.physa.2005.12.002>
- Estrada, E., Rodríguez-Velázquez, J.A.: Subgraph centrality in complex networks. *Phys. Rev. E* **71**(5), 056103 (2005). <https://doi.org/10.1103/physreve.71.056103>
- Berge, C.: *Graphs and Hypergraphs*. American Elsevier Pub. Co, New Jersey (1973)
- Xu, Y., Rockmore, D., Kleinbaum, A.M.: Hyperlink prediction in hypernetworks using latent social features. In: DS 2013. Springer, pp. 324–339 (2013). [https://doi.org/10.1007/978-3-642-40897-7\\_22](https://doi.org/10.1007/978-3-642-40897-7_22)
- Zhang, M.H., Cui, Z.C., Jiang, S.L., Chen, Y.X.: Beyond link prediction: predicting hyperlinks in adjacency space. In: AAAI'18. AAAI Press, pp. 4430–4437 (2018). <https://doi.org/10.1609/aaai.v32i1.11780>
- Sharma, G., Patil, P., Murty, M.N.: C3MM: Clique-Closure based hyperlink prediction. In: IJCAI'20. pp. 3364–3370 (2020). <https://doi.org/10.24963/ijcai.2020/465>
- Kumar, T., Darwin, K., Parthasarathy, S., Ravindran, B.: HPRA: hyperedge prediction using resource allocation. In: WebSci'20. Association for Computing Machinery, pp. 135–143 (2020). <https://doi.org/10.1145/3394231.3397903>
- Pan, L.M., Shang, H.J., Li, P.Y., Dai, H.X., Wang, W., Tian, L.X.: Predicting hyperlinks via hypernetwork loop structure. *Europhys. Lett.* **135**(4), 48005 (2021). <https://doi.org/10.1209/0295-5075/ac1a22>
- Maurya, D., Ravindran, B.: Hyperedge prediction using tensor eigenvalue decomposition. *J. Indian Inst. Sci.* **101**(3), 443–453 (2021). <https://doi.org/10.1007/s41745-021-00225-5>
- Fatemi, B., Taslakian, P., Vazquez, D., Poole, D.: Knowledge hypergraphs: prediction beyond binary relations. In: IJCAI'20. pp. 2191–2197 (2020). <https://doi.org/10.24963/ijcai.2020/303>
- Patil, P., Sharma, G., Murty, M.N.: Negative sampling for hyperlink prediction in networks. In: PAKDD 2020. Springer-Verlag, pp. 607–619 (2020). [https://doi.org/10.1007/978-3-030-47436-2\\_46](https://doi.org/10.1007/978-3-030-47436-2_46)
- Hwang, H., Lee, S., Park, C., Shin, K.: AHP: Learning to negative sample for hyperedge prediction. In: SIGIR'22. Association for Computing Machinery, pp. 2237–2242 (2022). <https://doi.org/10.1145/3477495.3531836>

34. Kipf, T.N., Welling, M.: Semi-Supervised classification with graph convolutional networks. In: ICLR 2017 (2017). <https://doi.org/10.48550/arXiv.1609.02907>
35. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: ICLR 2018 (2018). <https://doi.org/10.48550/arXiv.1710.10903>
36. Xue, J.W., Jiang, N., Liang, S.W., Pang, Q.Y., Yabe, T., Ukkusuri, S.V., Ma, J.Z.: Quantifying the spatial homogeneity of urban road networks via graph neural networks. *Nat. Mach. Intell.* **4**(3), 246–257 (2022). <https://doi.org/10.1038/s42256-022-00462-y>
37. Schulte-Sasse, R., Budach, S., Hnisz, D., Marsico, A.: Integration of multiomics data with graph convolutional networks to identify new cancer genes and their associated molecular mechanisms. *Nat. Mach. Intell.* **3**(6), 513–526 (2021). <https://doi.org/10.1038/s42256-021-00325-y>
38. Wang, Y.Y., Wang, J.R., Cao, Z.L., Farimani, A.B.: Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.* **4**(3), 279–287 (2022). <https://doi.org/10.1038/s42256-022-00447-x>
39. Zhang, M.H., Chen, Y.X.: Link prediction based on graph neural networks. In: NIPS'18. Curran Associates Inc., pp. 5171–5181 (2018). <https://doi.org/10.48550/arXiv.1802.09691>
40. Yadati, N., Nitin, V., Nimishakavi, M., Yadav, P., Louis A., Talukdar, P.: NHP: neural hypergraph link prediction. In: CIKM'20. Association for Computing Machinery, pp. 1705–1714 (2020). <https://doi.org/10.1145/3340531.3411870>
41. Zhou, D.Y., Huang, J.Y., Schölkopf, B.: Learning with hypergraphs: clustering, classification, and embedding. In: NIPS'06. MIT Press, pp. 1601–1608 (2006)
42. Gallo, G., Longo, G., Pallottino, S., Nguyen, S.: Directed hypergraphs and applications. *Discrete Appl. Math.* **42**(2–3), 177–201 (1993). [https://doi.org/10.1016/0166-218x\(93\)90045-p](https://doi.org/10.1016/0166-218x(93)90045-p)
43. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2012). <https://doi.org/10.1145/3065386>
44. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR 2015 (2015). <https://doi.org/10.48550/arXiv.1412.6980>
45. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS 2010. pp. 249–256 (2010)
46. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014). <https://doi.org/10.5555/2627435.2670313>
47. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs). In: ICLR 2016 (2015). <https://doi.org/10.48550/arXiv.1511.07289>
48. Feng, Y.F., You, H.X., Zhang, Z.Z., Ji, R.R., Gao, Y.: Hypergraph neural networks. In: AAAI'2019. AAAI Press, pp. 3558–3565 (2019). <https://doi.org/10.1609/aaai.v33i01.33013558>
49. Zhang, R.C., Zou, Y.S., Ma, J.: Hyper-SAGNN: A self-attention based graph neural network for hypergraphs. In: ICLR'2020 (2020). <https://doi.org/10.48550/arXiv.1911.02613>
50. Grover, A., Leskovec, J.: Node2vec: Scalable feature learning for networks. In: KDD'16. Association for Computing Machinery, pp. 855–864 (2016). <https://doi.org/10.1145/2939672.2939754>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.