Pattern Recognition 146 (2024) 109995

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Collaborative contrastive learning for hypergraph node classification

Hanrui Wu [a,e], Nuosi Li [a], Jia Zhang [a,e], Sentao Chen [b], Michael K. Ng [c], Jinyi Long [a,d,e,*]

[a] Jinan University, Guangzhou, China
[b] Shantou University, Shantou, China
[c] Hong Kong Baptist University, Hong Kong, China
[d] Guangdong Key Lab of Traditional Chinese Medicine Information Technology, Guangzhou, China
[e] Pazhou Lab, Guangzhou, China

ARTICLE INFO

Keywords:
Hypergraph
Hypergraph convolution
Contrastive learning
Graph convolution
Node classification

ABSTRACT

Plenty of models have been presented to handle the hypergraph node classification. However, very few of these methods consider contrastive learning, which is popular due to its great power to represent instances. This paper makes an attempt to leverage contrastive learning to hypergraph representation learning. Specifically, we propose a novel method called Collaborative Contrastive Learning (CCL), which incorporates a generated standard graph with the hypergraph. The main technical contribution here is that we develop a collaborative contrastive schema, which performs contrast between the node views obtained from the standard graph and hypergraph in each network layer, thus making the contrast collaborative. To be precise, in the first layer, the view from the standard graph is used to augment that from the hypergraph. Then, in the next layer, the augmented features are adopted to train a new representation to augment the view from the standard graph conversely. With this setting, the learning procedure is alternated between the standard graph and hypergraph. As a result, the learning on the standard graph and hypergraph is collaborative and leads to the final informative node representation. Experimental results on several widely used datasets validate the effectiveness of the proposed model.

1. Introduction

Graph neural networks (GNNs) [1] have obtained increasing attention in recent years in applications where data are generated from non-Euclidean domains and have complex relationships between samples. GNN techniques have been successfully applied to many areas, like graph classification [2,3], recommendation systems [4,5], and image classification [6]. Recent studies, such as [7,8], showed that a standard graph only presents the pair-wise relations between samples such that it may lose valuable information. To address this issue, some researchers tend to exploit hypergraphs [9–11]. Mathematically, a hypergraph is a generalization of a graph where an edge can join any number of nodes. Learning on hypergraphs, i.e., hypergraph learning [8,10], has shown to be an effective tool for studying complex data structures. Compared with a standard graph, a hypergraph is more capable and flexible in modeling high-order relations since one hyperedge in a hypergraph connects more than two nodes [8]. As a result, hypergraph learning is gaining increasing attention and has been applied to many research fields, such as computer vision [12,13] and recommendation systems [14,15].

Recently, contrastive learning [16] has achieved state-of-the-art performance in the area of self-supervised representation learning [17]. Contrastive learning is becoming popular due to its superior ability to bring together two views of the same instance while pushing apart two views of different instances. With the advance in contrastive learning methods, the power of representing instances has significantly improved. Consequently, various applications [16,18] take advantage of contrastive learning and receive great success. Especially, applying contrastive learning to graphs, i.e., graph contrastive learning [19,20], has become a prominent strategy for graph representation learning. However, most related models, such as [19,21], focus on the standard graphs, and contrastive learning on hypergraphs is still an open and under-researched topic.

To fill the vacancy and enrich the study of hypergraph contrastive learning, we propose to investigate the usage of contrastive learning in hypergraph representation learning for the node classification task. On the one hand, although a few works, such as [22,23], have proven that contrastive learning is effective in hypergraph learning, these models are proposed for recommendation systems. Little work is proposed to face the hypergraph node classification problem, which may be attributed to the difficulty of discovering another representation of nodes. Particularly, Hypergraph Structure Learning (HSL) [24] is

* Corresponding author at: Jinan University, Guangzhou, China.
E-mail address: jinyil@jnu.edu.cn (J. Long).

https://doi.org/10.1016/j.patcog.2023.109995
Received 25 January 2023; Received in revised form 27 August 2023; Accepted 22 September 2023
Available online 26 September 2023

the first work to perform contrastive learning on hypergraphs for the node classification task. HSL aims to maintain the consistency between the learned structure and the original structure using intra-hyperedge contrast, thus making predictions for hypergraph nodes. However, HSL does not seek to conduct contrast in each network layer and make the contrast collaborative to aggregate information. On the other hand, current hypergraph learning models attempt to learn latent representations based on either nodes [9] or the combination of nodes and hyperedges [8,25,26]. It is usually ignored that a standard graph could be easily generated based on a hypergraph. For example, given a hypergraph represented by an incidence matrix that indicates the relations between nodes and hyperedges, we are able to present the relations between nodes by multiplying the incidence matrix by its transpose.

Motivated by the above, in this paper, we propose a novel model called Collaborative Contrastive Learning (CCL) to handle the hypergraph node classification problem. Specifically, CCL incorporates a generated standard graph with a hypergraph and employs contrastive learning to make the standard graph and hypergraph collaborative. To achieve this goal, we propose a collaborative contrastive schema to make use of the convolution networks on the standard graph and hypergraph, i.e., graph convolutional network (GCN) [27] and hypergraph convolutional network (HGCN) [9]. Different from previous contrastive learning works, we aim to discover the contrast in each network layer so as to aggregate useful information collaboratively. Particularly, in each layer, we have two kinds of node views by applying GCN to the standard graph and HGCN to the hypergraph. In the first layer, the view from the standard graph is adopted to augment the view from the hypergraph using contrastive learning. Specifically, the views of the same node are treated as positive pairs, while the ones of any different nodes are regarded as negative pairs. In the second layer, the augmented view of the first layer is used to train a new hypergraph representation for augmenting the view from the standard graph conversely. Consequently, as the learning progress proceeds, the consistency between different views of the same node and the divergence across different nodes are both ensured. Besides, the standard graph and hypergraph are collaboratively contrastive, and the learning on them is alternated to obtain the final node representation. We highlight the main contributions of this paper as follows:

- We propose a hypergraph node classification model named Collaborative Contrastive Learning (CCL), which fully takes contrast in each layer from GCN and HGCN into consideration. We show that the contrast can be done not only on the final network output as most previous works did but also in each layer, which makes our paper a decent contribution to the hypergraph learning field.
- We develop a collaborative contrastive schema that utilizes the convolutional networks on the standard graph and hypergraph, making the node representation alternately contrastive and augmented.
- We perform comprehensive experiments on several famous datasets to demonstrate that the performance of the proposed model is better than that of baseline methods.

The remainder of the paper is organized as follows. We review important studies about hypergraph learning and contrastive learning in Section 2. Subsequently, we introduce the proposed method in Section 3. We conduct comprehensive experiments and discuss the experimental results in Section 4. Section 5 concludes the whole paper and discusses potential future work.

2. Related works

In this section, we study some relevant works about hypergraph learning and contrastive learning. We also emphasize the difference between the proposed model and the models related to it.

2.1. Hypergraph learning

Standard graphs have limitations for indicating high-order complex relationships, which can be represented by hyperedges connecting an arbitrary number of nodes in a hypergraph. Hypergraph learning has received widespread attention due to its excellent ability to represent high-order correlations. In [7], Zhou et al. employed hypergraphs to model high-order relationships for semi-supervised node classification and clustering tasks. In recent years, researchers have sought to introduce hypergraphs into neural networks and achieved successful performance improvement in various areas, such as citation networks [26], social network analysis [28], and recommendation systems [14,29]. For instance, Yang et al. represented location-based social network data via hypergraphs and proposed a hypergraph-based scheme to sample user check-ins and social relationships for learning node representations [28]. Besides, Wang et al. proposed an end-to-end hypergraph framework to capture correlations between items for next-item recommendation [29]. Generally, existing research has shown that hypergraph-based models usually perform better than standard graph-based ones on many learning tasks, like classification [9] and link prediction [30]. Below, we discuss primary works related to node classification. HyperGraph Neural Network (HGNN) [12] is proposed as the first neural network method on hypergraph structure, which introduces hypergraph Laplacian and truncated Chebyshev polynomials to the convolution operation. In [25], Hypergraph Networks with Hyperedge Neurons (HNHN) proposes a flexible normalization scheme and applies nonlinear activation functions to update the representations of nodes and hyperedges alternately. To further exploit the hyperedge information, Hypergraph Convolution on Nodes-Hyperedges (HCNH) [26] performs convolution operations on both nodes and hyperedges. It then seeks to recover the hypergraph to enhance the learning of the node representation. In addition, Hypergraph Collaborative Network (HCoN) [8] considers information from previous nodes and hyperedges and treats the hypergraph reconstruction error as a regularizer to assist the training of the classifier.

2.2. Contrastive learning

Contrastive learning is a representative method of self-supervised learning, which has gained extensive discussions in the fields of computer vision [31] and natural language processing [32]. The main intuition of contrastive learning is constructing positive and negative sample pairs, closing the distance between positive sample pairs while pushing the negative samples apart. In recent years, this idea has attracted researchers to explore the effectiveness of contrastive learning in graph representation learning [20,23,33,34]. For example, Hassani et al. introduced a self-supervised method for learning graph and node representations by comparing two structural views from the graph, including first-order neighbors and graph diffusion [33]. Besides, You et al. proposed Graph Contrastive Learning (GraphCL), which designs different types of graph augmentations, i.e., node dropping and edge perturbation, and investigates the impacts of different combinations of graph augmentations [20]. In [34], Yu et al. proposed a simple contrastive learning method that discards graph augmentation and adds uniform noise in the feature space to create contrastive views.

Contrastive learning on hypergraphs is the main topic of this paper. There are a few related tailored approaches [23,35] to real-world applications. Nevertheless, these works have demonstrated little regarding node classification. There are two relevant works to our method, i.e., HSL [24] and Tri-directional Contrastive Learning (TriCL) [36]. In HSL [24], Cai et al. proposed treating the redundant hyperedges and nodes in the original hypergraph as the augmented graph. To maintain the consistency between the optimized hypergraph structure and the original hypergraph, HSL develops intra-hyperedge contrastive learning to maximize the mutual information between the node representations

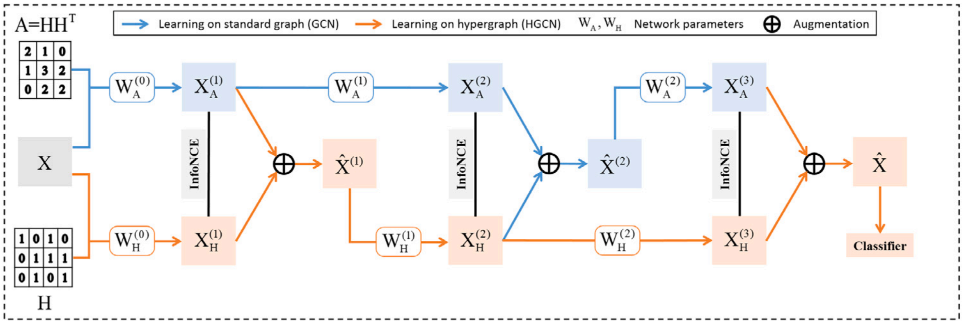


Fig. 1. An illustration of the proposed CCL model. $\mathbf{W}_{\mathrm{A}}$ and $\mathbf{W}_{\mathrm{H}}$ are the network parameters for GCN and HGCN, respectively, and the blue and red branches indicate GCN and HGCN, respectively. In the first layer, the view from GCN, i.e., $\mathbf{X}_{\mathrm{A}}^{(1)}$, is used to augment that from HGCN, i.e., $\mathbf{X}_{\mathrm{H}}^{(1)}$, and achieve an augmented representation, i.e., $\hat{\mathbf{X}}^{(1)}$, which is then put into HGCN to learn a new representation, i.e., $\mathbf{X}_{\mathrm{H}}^{(2)}$. In the second layer, the view from HGCN, i.e., $\mathbf{X}_{\mathrm{H}}^{(2)}$, is adopted conversely to augment that from GCN, i.e., $\mathbf{X}_{\mathrm{A}}^{(2)}$, and obtain another augmented representation, i.e., $\hat{\mathbf{X}}^{(2)}$. The final layer works similarly to the first layer to learn the final representation, i.e., $\hat{\mathbf{X}}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

from different hypergraph structures. In TriCL [36], Lee et al. developed tri-directional contrast to maximize in two augmented views the agreement between the same node, between the same group of nodes, as well as between each group and its members. Although HSL and TriCL adopt contrastive learning on hypergraphs and handle the hypergraph node classification problem as ours does, they conduct contrast on the final network output. Unlike HSL and TriCL, the proposed model employs the views in every network layer for contrast to aggregate valuable information collaboratively.

3. Methodology

In this section, we first define the notations and their descriptions and make a statement about our learning problem. Subsequently, we provide detailed information on the proposed CCL method.

3.1. Notations and problem statement

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{\Omega})$ be a hypergraph that comprises a set of nodes $\mathcal{V}$ with $N = |\mathcal{V}|$, a set of hyperedges $\mathcal{E}$ with $M = |\mathcal{E}|$, and a diagonal matrix of hyperedge weights $\mathbf{\Omega} \in \mathbb{R}^{M\times M}$. If no further information about the importance of hyperedges is provided, each hyperedge $e \in \mathcal{E}$ is simply assigned with value 1, which indicates that the diagonal values in $\mathbf{\Omega}$ are set to be 1. Denote the node features by $\mathbf{X} \in \mathbb{R}^{N\times D}$, where D represents the feature dimension of nodes. Let $\mathbf{Y} \in \mathbb{R}^{N\times C}$ be the corresponding label matrix of the nodes, where C is the number of classes of nodes. Specifically, the hypergraph structure can be represented by an incidence matrix $\mathbf{H} \in \{0,1\}^{N\times M}$, where each entry $\mathbf{H}(v,e)$ means whether the node v is contained in the hyperedge e, i.e., 1 means contained and 0 indicates not contained. In this case, the degrees of nodes and hyperedges can be represented by matrices $\mathbf{D}_v \in \mathbb{R}^{N\times N}$ and $\mathbf{D}_e \in \mathbb{R}^{M\times M}$, respectively, where $\mathbf{D}_v$ and $\mathbf{D}_e$ are diagonal matrices with values being $\sum_{e\in\mathcal{E}} \mathbf{\Omega}(e,e)\mathbf{H}(v,e)$ and $\sum_{v\in\mathcal{V}} \mathbf{H}(v,e)$, respectively.

Next, we describe the problem statement. Given a hypergraph $\mathcal{G}$ with its node features $\mathbf{X}$ and several training nodes that have labels, the goal of this paper is to find a classifier $f : \mathbf{X} \to \mathbb{R}^C$ that assigns a label vector $f(v) \in \mathbb{R}^C$ to a node $v \in \mathcal{V}$. Particularly, the label of a unlabeled node is predicted as the index of the largest confidence score in its label vector.

3.2. Collaborative contrastive learning

3.2.1. Overview

In this paper, we study the problem of hypergraph node classification. Specifically, considering that existing models usually do not employ the standard graph representing relations between nodes derived from the hypergraph, we propose taking advantage of this standard graph to help improve the classification performance. As a result, each sample has two views by applying GCN and HGCN to the standard graph and hypergraph, respectively. Fig. 1 illustrates the main idea of the proposed CCL model. Particularly, in each layer of CCL, we utilize contrastive learning with the InfoNCE [37] to make the standard graph and hypergraph collaborative. As shown in Fig. 1, in the first layer, the view from GCN is used to augment that from HGCN and achieve an augmented representation, which is then put into HGCN to learn a new representation. Subsequently, in the second layer, the view from HGCN is adopted conversely to augment that from GCN and obtain another augmented representation, which is then employed to help the augmentation of HGCN in the final layer. In consequence, the learning progress is collaborative and alternated to learn the final informative representation.

3.2.2. Learning on hypergraph

Given a hypergraph $\mathcal{G}$ and its incidence matrix $\mathbf{H}$, we formulate the hypergraph convolutional network (HGCN) on the nodes as follows:

$$\mathbf{X}_{\mathrm{H}}^{(l+1)} = \sigma(\mathbf{H}\mathbf{\Omega}\mathbf{H}^{\top}\mathbf{X}_{\mathrm{H}}^{(l)}\mathbf{W}_{\mathrm{H}}^{(l)}), \tag{1}$$

where the superscript (l) means the (l)th layer, $\mathbf{W}_{\mathrm{H}}^{(l)}$ is the trainable weight parameter between the (l)th and $(l+1)$th layers, $\mathbf{X}_{\mathrm{H}}^{(l)}$ and $\mathbf{X}_{\mathrm{H}}^{(l+1)}$ are node representations in the (l)th and $(l+1)$th layers, respectively, and $\mathbf{X}_{\mathrm{H}}^{(0)} = \mathbf{X}$. $\sigma(\cdot)$ is a nonlinear activation function, such as ReLU. Following [8,9], we further adopt the symmetric normalization to avoid the possibility of vanishing gradients and numerical instabilities as follows:

$$\mathbf{X}_{\mathrm{H}}^{(l+1)} = \sigma(\mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}\mathbf{\Omega}\mathbf{D}_e^{-1}\mathbf{H}^{\top}\mathbf{D}_v^{-\frac{1}{2}}\mathbf{X}_{\mathrm{H}}^{(l)}\mathbf{W}_{\mathrm{H}}^{(l)}). \tag{2}$$

3.2.3. Learning on standard graph

Given a hypergraph indicating the relations between nodes and hyperedges, it is easy to obtain a standard graph representing the relations between nodes by multiplying the incidence matrix by its transpose. Therefore, by defining $\mathbf{A}$ as the adjacency matrix of the standard graph, we have $\mathbf{A} = \mathbf{H}\mathbf{H}^{\top}$. By making use of graph convolutional network (GCN) [27], we have the convolution operation on the nodes as follows:

$$\mathbf{X}_{\mathrm{A}}^{(l+1)} = \sigma(\mathbf{A}\mathbf{X}_{\mathrm{A}}^{(l)}\mathbf{W}_{\mathrm{A}}^{(l)}), \tag{3}$$

where $\mathbf{W}_{\mathrm{A}}^{(l)}$ is the trainable weight parameter between the (l)th and $(l+1)$th layers, $\mathbf{X}_{\mathrm{A}}^{(l)}$ and $\mathbf{X}_{\mathrm{A}}^{(l+1)}$ are node representations in the (l)th and $(l+1)$th layers, respectively, and $\mathbf{X}_{\mathrm{A}}^{(0)} = \mathbf{X}$. Similar to [27], we impose a normalization to obtain the formulation as follows:

$$\mathbf{X}_{\mathrm{A}}^{(l+1)} = \sigma(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{X}_{\mathrm{A}}^{(l)}\mathbf{W}_{\mathrm{A}}^{(l)}), \tag{4}$$

where $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$ is the degree matrix of $\mathbf{A}$.

3.2.4. Collaborative contrastive schema

Until now, we have discussed the learning on the hypergraph and standard graph. Next, we present the proposed collaborative contrastive schema. Specifically, we design a three-layer network architecture, which alternately takes the views from every layer of GCN and HGCN to augment the representations of nodes. We treat the views of the same node as the positive pairs and those of any different nodes as the negative pairs. As a consequence, the learning on positive pairs ensures the consistency between different representations of the same node, while the learning on negative pairs guarantees the divergence among different nodes. Specifically, we use the InfoNCE [37] for contrastive learning.

Let $\mathbf{x}_{\mathrm{H},i}^{(l)}$ and $\mathbf{x}_{\mathrm{A},i}^{(l)}$ be the views of ith node from $\mathbf{X}_{\mathrm{H}}^{(l)}$ and $\mathbf{X}_{\mathrm{A}}^{(l)}$, respectively, in the (l)th layer. In the first layer of the proposed CCL model, we obtain the (1)-st layer views, i.e., $\mathbf{X}_{\mathrm{A}}^{(1)}$ and $\mathbf{X}_{\mathrm{H}}^{(1)}$, based on Eqs. (2) and (4), respectively, where $l = 1$. Then, we use the view from GCN to augment that from HGCN and formulate the InfoNCE as follows:

$$\mathcal{L}_{\mathrm{cl}}^{(1)} = \sum_{i\in\mathcal{V}} -\log \frac{\exp(s(\mathbf{x}_{\mathrm{H},i}^{(1)},\ \mathbf{x}_{\mathrm{A},i}^{(1)})/\tau)}{\sum_{j\in\mathcal{V}} \exp(s(\mathbf{x}_{\mathrm{H},i}^{(1)},\ \mathbf{x}_{\mathrm{A},j}^{(1)})/\tau)}, \tag{5}$$

where $i \neq j$, i and j indicate the ith and jth nodes, respectively, τ is the temperature parameter, and $s(\cdot,\cdot)$ is the function calculating the cosine similarity between two vectors. In this layer, $\{(\mathbf{x}_{\mathrm{H},i}^{(1)}, \mathbf{x}_{\mathrm{A},i}^{(1)})|i \in \mathcal{V}\}$ is the positive pairs and $\{(\mathbf{x}_{\mathrm{H},i}^{(1)}, \mathbf{x}_{\mathrm{A},j}^{(1)})|i, j \in \mathcal{V}, i \neq j\}$ is the negative pairs. By contrasting $\mathbf{X}_{\mathrm{H}}^{(1)}$ with $\mathbf{X}_{\mathrm{A}}^{(1)}$, we obtain the augmented representation, denoted by $\hat{\mathbf{X}}^{(1)}$.

In the second layer, the view of the first layer in GCN, i.e., $\mathbf{X}_{\mathrm{A}}^{(1)}$, will be put into Eq. (4), while the augmented representation based on GCN and HGCN, i.e., $\hat{\mathbf{X}}^{(1)}$, will be the input of Eq. (2), as shown in Fig. 1. In this way, we achieve the (2)-nd layer views, i.e., $\mathbf{X}_{\mathrm{A}}^{(2)}$ and $\mathbf{X}_{\mathrm{H}}^{(2)}$. Then, we attempt to make the learning between the standard graph and hypergraph collaborative to achieve informative representations by applying the view from HGCN to augment that from GCN. Thus, we formulate the InfoNCE in the second layer as follows:

$$\mathcal{L}_{\mathrm{cl}}^{(2)} = \sum_{i\in\mathcal{V}} -\log \frac{\exp(s(\mathbf{x}_{\mathrm{A},i}^{(2)},\ \mathbf{x}_{\mathrm{H},i}^{(2)})/\tau)}{\sum_{j\in\mathcal{V}} \exp(s(\mathbf{x}_{\mathrm{A},i}^{(2)},\ \mathbf{x}_{\mathrm{H},j}^{(2)})/\tau)}, \tag{6}$$

where $\{(\mathbf{x}_{\mathrm{A},i}^{(2)}, \mathbf{x}_{\mathrm{H},i}^{(2)})|i \in \mathcal{V}\}$ and $\{(\mathbf{x}_{\mathrm{A},i}^{(2)}, \mathbf{x}_{\mathrm{H},j}^{(2)})|i, j \in \mathcal{V}, i \neq j\}$ are the positive and negative sample pairs, respectively. Eq. (6) leads to the new augmented representation, denoted by $\hat{\mathbf{X}}^{(2)}$.

In the last layer, the view of the second layer in HGCN, i.e., $\mathbf{X}_{\mathrm{H}}^{(2)}$, will be stacked into Eq. (2), while the augmented representation learned from GCN and HGCN, i.e., $\hat{\mathbf{X}}^{(2)}$, will be the input of Eq. (4). Therefore, we obtain the (3)-rd layer views, i.e., $\mathbf{X}_{\mathrm{A}}^{(3)}$ and $\mathbf{X}_{\mathrm{H}}^{(3)}$. In this layer, the view from GCN is adopted to augment that from HGCN. Similar to the first layer, we have the InfoNCE as follows:

$$\mathcal{L}_{\mathrm{cl}}^{(3)} = \sum_{i\in\mathcal{V}} -\log \frac{\exp(s(\mathbf{x}_{\mathrm{H},i}^{(3)},\ \mathbf{x}_{\mathrm{A},i}^{(3)})/\tau)}{\sum_{j\in\mathcal{V}} \exp(s(\mathbf{x}_{\mathrm{H},i}^{(3)},\ \mathbf{x}_{\mathrm{A},j}^{(3)})/\tau)}, \tag{7}$$

where $\{(\mathbf{x}_{\mathrm{H},i}^{(3)}, \mathbf{x}_{\mathrm{A},i}^{(3)})|i \in \mathcal{V}\}$ and $\{(\mathbf{x}_{\mathrm{H},i}^{(3)}, \mathbf{x}_{\mathrm{A},j}^{(3)})|i, j \in \mathcal{V}, i \neq j\}$ are the positive and negative sample pairs, respectively.

In summary, CCL alternately takes the view of each layer from GCN and HGCN for contrast. Particularly, the use of contrastive learning in the proposed CCL method makes the standard graph and hypergraph collaborative to learn the final informative representation.

3.2.5. Model objective function

Based on Eqs. (5), (6), and (7), we obtain the final node representation, denoted by $\hat{\mathbf{X}}$. Then, we apply a softmax function to it to make predictions. Specifically, we evaluate the cross-entropy error over the labeled nodes as follows:

$$\mathcal{L}_{\mathrm{ce}} = -\sum_{i\in\mathcal{Y}_L} \sum_{c=1}^{C} \mathbf{Y}_{ic} \ln \hat{\mathbf{X}}_{ic}, \tag{8}$$

where $\mathcal{Y}_L$ is the set of node indices that have labels. By incorporating Eqs. (5), (6), (7), and (8), we have the overall model objective function as follows:

$$\mathcal{L} = \mathcal{L}_{\mathrm{ce}} + \lambda \sum_{l=1}^{n} \mathcal{L}_{\mathrm{cl}}^{(l)}, \tag{9}$$

where λ is the trade-off parameter and n is the number of layers, we have $n = 3$ here.

4. Experiment

We perform extensive experiments to estimate the performance of the proposed model in this section. Specifically, we first give the descriptions of the datasets used in our experiments and list several baseline methods. Subsequently, we present the experimental results and discussions.

4.1. Datasets

We adopt several widely used benchmark datasets, i.e., Citeseer [38], Pubmed [39], Cora [40], and DBLP [41], to evaluate the effectiveness of the proposed method. Particularly, we focus on the hypergraph node classification task and employ the setting proposed in [8,25] to make fair comparisons with the baseline methods. Therefore, we follow [8,25] and use two kinds of datasets, i.e., co-citation and co-authorship, where the labels are subject topics of papers. Table 1 also presents the statistical information of the used datasets.

- **Co-citation data**
  The Citeseer, Cora, and Pubmed datasets are constructed as co-citation data, which consist of a set of documents and their citation links. To obtain a co-citation hypergraph, we treat documents as nodes and the citation links as hyperedges. According to [8,25,36], the initial node features of the Citeseer and Cora datasets are bag-of-words vector representations and those of the Pubmed dataset are the term frequency-inverse document frequency (TF-IDF).
- **Co-authorship data**
  The Cora and DBLP datasets are built as co-authorship data, which contain a set of documents with their authors. The hypergraph is created with each node representing a document and each hyperedge connecting all documents co-authored by one author. For the Cora co-authorship data collected from the website,[1] we remove the hyperedges that contain only one node and the nodes that are not connected to any hyperedges. To obtain the original features of the nodes, we select the most frequent one- and two-grams from all abstracts to construct a vocabulary with the size of 1000 and use a stopword removal mechanism, i.e., discard any term that appears in more than 20% of the documents. Subsequently, TF-IDF is executed to obtain the initial feature of the node. DBLP co-authorship data are collected and processed in [42]. Similar to the Cora co-authorship dataset, we delete the nodes connecting to no hyperedges.

4.2. Compared models

We compare the proposed CCL method with several models, including 1-Nearest Neighbor (NN), Support Vector Machine (SVM), GCN [27], HGNN [12], HyperGCN [42], FastHyperGCN [42], HNHN [25], UniGCN [43], UniGAT [43], HCNH [26], HCoN [8], and TriCL [36]. Among the baseline models, NN and SVM are methods without considering the graph structure information. Specifically, we

[1] https://people.cs.umass.edu/~mccallum/data.html

Table 1
Statistical information of the used datasets.

Dataset	#nodes	#edges	#classes	#features	Label rate
Citeseer co-citation	1498	1107	6	3703	0.150
Cora co-citation	1434	1579	7	1433	0.100
Pubmed co-citation	3840	7963	3	500	0.020
Cora co-authorship	16313	7389	10	1000	0.052
DBLP co-authorship	41302	22363	6	1425	0.042

Table 2
Accuracy (%, mean ± standard deviation) of different methods on the co-citation and co-authorship datasets.

Method	Co-citation			Co-authorship	
	Citeseer	Cora	Pubmed	Cora	DBLP
NN	27.0 ± 9.0	24.5 ± 5.7	41.2 ± 2.7	28.9 ± 8.1	20.6 ± 6.2
SVM	21.0 ± 1.3	32.6 ± 5.2	41.3 ± 4.1	52.1 ± 1.1	27.2 ± 0.1
GCN [27]	63.0 ± 1.9	72.1 ± 1.9	69.6 ± 6.3	40.0 ± 3.3	83.2 ± 2.6
FastHyperGCN [42]	56.1 ± 11.2	50.4 ± 11.7	54.4 ± 10.0	45.2 ± 12.9	70.5 ± 14.3
HyperGCN [42]	54.7 ± 9.8	52.2 ± 11.4	60.0 ± 10.7	55.0 ± 0.9	71.3 ± 1.2
HGNN [12]	61.1 ± 2.2	76.9 ± 1.8	63.3 ± 2.2	58.2 ± 0.3	77.6 ± 0.4
HNHN [25]	64.8 ± 1.6	71.3 ± 1.9	75.9 ± 1.5	63.9 ± 0.8	85.1 ± 0.2
UniGCN [43]	70.9 ± 1.0	78.3 ± 1.7	78.8 ± 1.7	61.4 ± 0.9	88.3 ± 0.2
UniGAT [43]	70.7 ± 1.0	78.5 ± 1.8	78.7 ± 1.7	60.6 ± 0.9	87.8 ± 0.3
HCNH [26]	71.4 ± 1.2	78.5 ± 1.7	77.1 ± 3.6	65.9 ± 0.5	–
HCoN [8]	71.2 ± 2.7	79.0 ± 1.5	80.4 ± 1.1	**66.5 ± 0.5**	88.0 ± 0.1
TriCL [36]	71.1 ± 0.8	**81.0 ± 1.0**	80.3 ± 1.6	64.2 ± 0.5	88.5 ± 0.1
CCL (ours)	**71.9 ± 0.8**	79.7 ± 1.1	**81.9 ± 0.9**	64.6 ± 0.5	**88.6 ± 0.1**

use the tools provided by sklearn[2] to run NN and SVM. GCN and HGNN (HGCN and HGNN have similar convolution operations, we refer HGNN to HGCN) are basic compared approaches to the proposed model, and the remaining models can be treated as state-of-the-art methods.[3] In particular, for the baseline models, we use the default settings provided by the original papers or codes.

4.3. Implementation details

We implement the proposed model in the Pytorch[4] platform. Specifically, we develop a three-layer graph network architecture with the output dimension being $512 \rightarrow 256 \rightarrow C$. Additionally, we train the model for 200 epochs using the Adam optimizer with a learning rate of 0.05 and early stopping with a window size of 100. Moreover, we adopt ReLU as the activation function and tune the L2 regularization factor in a range space {0.001, 0.003, 0.005}. Following [8], we use one trial to tune the value of the L2 regularization factor and apply the selected value to the remaining trials. The proposed CCL model has a trade-off parameter, i.e., λ, which is empirically set to be 0.1 for all the learning tasks. Besides, for the temperature parameter τ contained in the InfoNCE for contrastive learning, we set it to be 0.2.

4.4. Results

Table 2 lists the results in terms of the accuracy of the test models on the co-citation and co-authorship datasets. For the datasets except Cora co-citation, we directly quote the results from [8] except UniGCN, UniGAT, and TriCL to make the comparisons with the baseline methods fair. Specifically, we follow [8] and run 100 trials to obtain the mean and standard deviation results. We observe that the proposed CCL model obtains better or competitive results compared with the baseline methods, which verifies the effectiveness of the proposed model. We have several interesting observations as follows.

- NN and SVM obtain poorer performance than graph-based models in general, which means that the graph structure information cannot be omitted, and specific models for graph-structured data are needed.
- GCN and HGNN are two basic compared models to CCL as CCL involves the convolutional operations from GCN and HGNN. Compared with GCN and HGNN, CCL performs better, demonstrating that exploiting contrastive learning to make the convolutions of the standard graph and hypergraph collaborative is helpful in enhancing the ability to represent instances.
- HNHN, HCNH, and HCoN are state-of-the-art methods incorporating the hyperedge information, CCL does not involve such information and generally outperforms them. Besides, TriCL and CCL are hypergraph learning models based on contrastive learning, and CCL outperforms TriCL on most tasks. These phenomena verify that conducting contrast in every network layer to aggregate valuable information leads to more informative representations.
- Note that the proposed model is not a panacea for all the datasets. CCL does not achieve the best results on the Cora co-citation and Cora co-authorship datasets, indicating the *No Free Lunch* [44] theorem still applies. Specifically, CCL performs worse than HCNH and HCoN on the Cora co-authorship dataset, the reason may be that GCN works poorly (even worse than SVM), and CCL is a GCN-based model.

4.5. Model variants

CCL is composed of two basic networks, i.e., GCN and HGNN, for the standard graph and hypergraph, respectively. To investigate the effectiveness of the proposed collaborative contrastive schema, we develop several variants of the proposed CCL model. Specifically, we replace the basic networks with other approaches and generate three variants, i.e., GCN+UniGCN, GAT+HGNN, and GAT+UniGCN, where GCN and GAT [45] are for standard graphs and HGNN and UniGCN are for hypergraphs. The results on the Cora co-citation dataset are exhibited in Fig. 2, in which the average accuracies of GCN+UniGCN, GAT+HGNN, and GAT+UniGCN are 64.2 ± 16.0, 77.2 ± 2.0, and 73.7 ± 7.0, respectively. Fig. 2 also presents the maximum mean discrepancy (MMD) [6,46] values between the views generated by two basic networks in different layers of different variants. Here, MMD is

[2] https://scikit-learn.org/

[3] Note that we do not compare with HSL [24] for two reasons: first, the code is not publicly available, and it is not easy for us to reproduce. Second, the detailed data settings are different from ours, so we cannot directly use their results.

[4] https://pytorch.org/

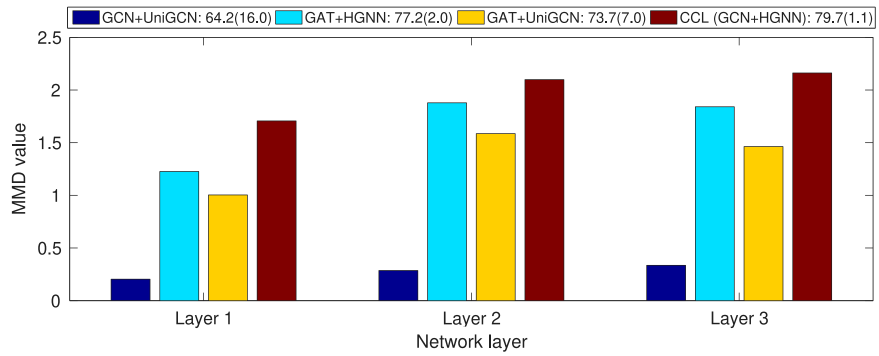


Fig. 2. Diversity estimation on the variants of CCL on the Cora co-citation dataset.

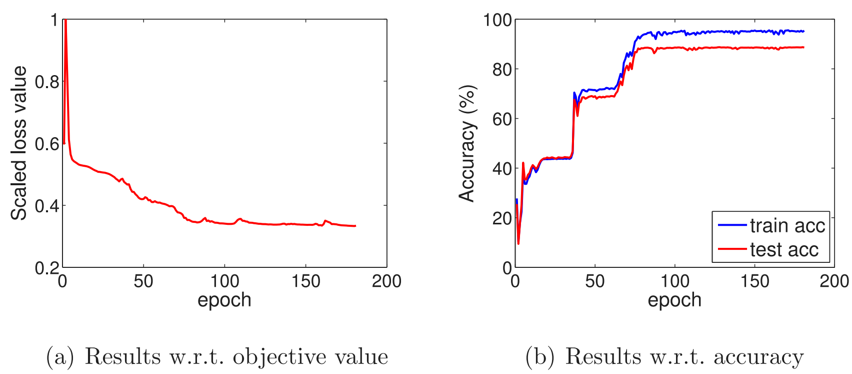


Fig. 3. Learning curves of the proposed CCL method on the DBLP co-authorship dataset.

used to estimate the diversity of two different views, i.e., higher MMD means higher diversity. We observe that high diversity leads to better classification performance, i.e., the ranks of both diversity and accuracy are CCL > GAT+HGNN > GAT+UniGCN > GCN+UniGCN. This phenomenon is reasonable because the collaborative contrastive schema performs contrast between two views, whose diversity is expected to be high to indicate different perspectives of the same node, thus learning effective representations.

4.6. Model learning curves

We study the model learning curves w.r.t. objective value and accuracy by taking the DBLP co-authorship dataset as an example. The experimental results are exhibited in Fig. 3, where Figs. 3(a) and 3(b) plot the learning curves for the loss and accuracy, respectively. Specifically, we rescale the objective values to the range [0, 1] by dividing them by their maximum value. It can be observed from Fig. 3 that as the number of epochs increases, the scaled objective values decrease and flatten out and the training and test accuracies improve and reach stable values, meaning the learning algorithm is converged. This phenomenon further demonstrates the effectiveness of the proposed method.

4.7. Effect of parameters

The proposed CCL model involves a trade-off parameter λ to control the strengths of contrastive loss, which is critical to classification performance. To discover the effect of λ, we adopt the Citeseer and Pubmed co-citation datasets as examples and tune λ in a range space {0.001, 0.01, 0.1, 1, 10}. The experimental results are plotted in Fig. 4(a), from which we observe that a reasonable value of λ helps improve the performance. As the value of λ increases, the model performance first increases and then decreases, and the best value of λ is 0.1. Specifically, setting λ to be a large value could break the balance between the classification loss and contrastive loss, leading to performance degradation.

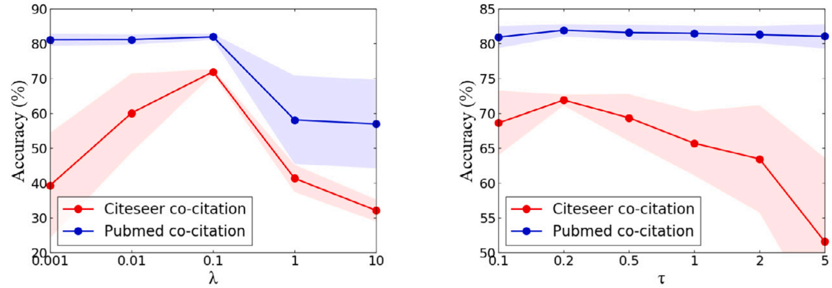
The proposed contrastive loss contains a temperature parameter τ. Similarly, we run experiments on the Citeseer and Pubmed co-citation datasets to investigate the impacts of τ. Specifically, we vary τ in a range space {0.1, 0.2, 0.5, 1, 2, 5} and report the results in Fig. 4(b), where we see that the performance on the Citeseer co-citation dataset is more sensitive to τ than that on the Pubmed co-citation dataset. Generally, the performance improves and then drops as the value of τ increases.

4.8. Effect of contrasting each layer

In this paper, we set the number of layers n in Eq. (9) to be 3. Here, we evaluate the influence of model depth (number of layers), which also reveals the importance of contrasting each layer in the proposed model. Specifically, we set n to be 1, 2, 3, 4, and 5 and exhibit the results of 100 trials in Figs. 5(a) and 5(b), which indicate the average training and test accuracy on the Citeseer and Pubmed co-citation datasets, respectively. For the learning tasks considered here, the best performance is achieved with a three-layer model, which is significantly better than the one-layer model, indicating the effectiveness of performing contrast in each layer. The performance drops for models deeper than three layers, which we conjecture that with model depth, the training becomes difficult, and the increasing number of parameters may lead to overfitting. This phenomenon also occurs in other graph neural networks, such as GCN [27].

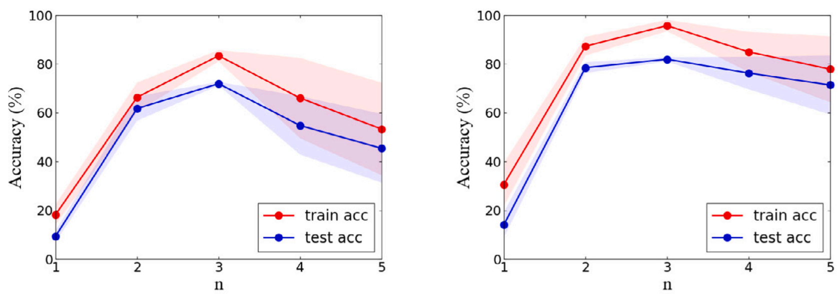
4.9. Efficiency evaluation

We evaluate the efficiency of the proposed method on the Pubmed co-citation dataset and run the experiment on an Ubuntu server with NVIDIA GeForce RTX 3080 Ti GPU. The test methods, including GCN, HGNN, HNHN, UniGCN, UniGAT, HCNH, HCoN, TriCL, and CCL, are



(a) Results w.r.t. λ (b) Results w.r.t. τ

Fig. 4. Parameter sensitivity study of the proposed model. The shaded areas denote standard error.



(a) Results w.r.t. n on Citeseer (b) Results w.r.t. n on Pubmed

Fig. 5. Influence of model depth of the proposed model. The shaded areas denote standard error.

Table 3
Wall-clock time (s) of different methods on the Pubmed co-citation dataset.

Method	Time	Time per epoch
GCN [27]	0.8663	0.0043 ± 0.0416
HGNN [12]	1.2611	0.0063 ± 0.0429
HNHN [25]	2.0174	0.0101 ± 0.0447
UniGCN [43]	1.1939	0.0060 ± 0.0496
UniGAT [43]	1.5041	0.0075 ± 0.0448
HCNH [26]	4.2108	0.0211 ± 0.0654
HCoN [8]	7.8374	0.0392 ± 0.0425
TriCL [36]	114.9478	0.5747 ± 0.0451
CCL (ours)	4.3299	0.0216 ± 0.0424

run for 200 epochs without considering early stopping. We estimate the results in seconds wall-clock time and report them in Table 3. Following [8], the results w.r.t. the training time of all epochs and the average training time per epoch, including forward pass, objective loss calculation, and backward pass, are listed. The proposed CCL model spends more time than GCN and HGNN since CCL performs contrast between views from GCN and HGCN (HGNN). Besides, CCL costs less time than HCoN because HCoN needs to incorporate the information of nodes and hyperedges in each layer. TriCL requires the most time to finish the task as it jointly considers node-, group-, and membership-level contrast. Taking the learning performance into account, the proposed model consumes reasonable training time and obtains competitive classification accuracy.

4.10. Visualization

We estimate the effectiveness of the proposed CCL method qualitatively by inputting the node representations to the Uniform Manifold Approximation and Projection (UMAP) [47] tool for visualization. Specifically, we compare the features learned by CCL with the original one and those learned by GCN and HCoN. We take the Pubmed co-citation and DBLP co-authorship datasets as examples and depict the results in Fig. 6, where different colors represent different labels. We observe that the original node features are chaotically distributed and the node features learned by graph learning methods are easier to classify. Compared with GCN, both HCoN and CCL cluster better the samples of the same category and separate those of different classes. However, CCL still shows an advantage over HCoN, especially on the DBLP dataset as presented in Figs. 6(g) and 6(h), where the blue labels are mixed up with cyan ones in HCoN, while the blue and cyan labels are separated in CCL, explaining why the proposed approach is capable of achieving better performance.

5. Conclusion

In this paper, we study the problem of hypergraph node classification. To this end, we introduce contrastive learning into hypergraph learning. Specifically, we first build a standard graph that represents the relationship between nodes based on the given hypergraph. Then, we propose a collaborative contrastive schema, which considers the node views learned by applying convolutional operations on the standard graph and hypergraph in each layer. Consequently, the views of the standard graph and hypergraph are collaboratively contrastive and are alternately augmented to achieve the final informative representation. We perform extensive experiments on several widely used datasets and compare the proposed model with well-known baseline methods. Experimental results demonstrate the effectiveness of the proposed method. In general, we show in this paper that one can perform contrast not only on the final network output but also in each layer, and this strategy improves the performance of hypergraph learning.

Note that the proposed model works in a transductive manner and is not able to make predictions for hypergraph nodes that are unseen during training. Therefore, in the future, we are going to extend the proposed model to the inductive learning schema using some neural message-passing techniques like [48]. Besides, as shown in Figs. 5(a)

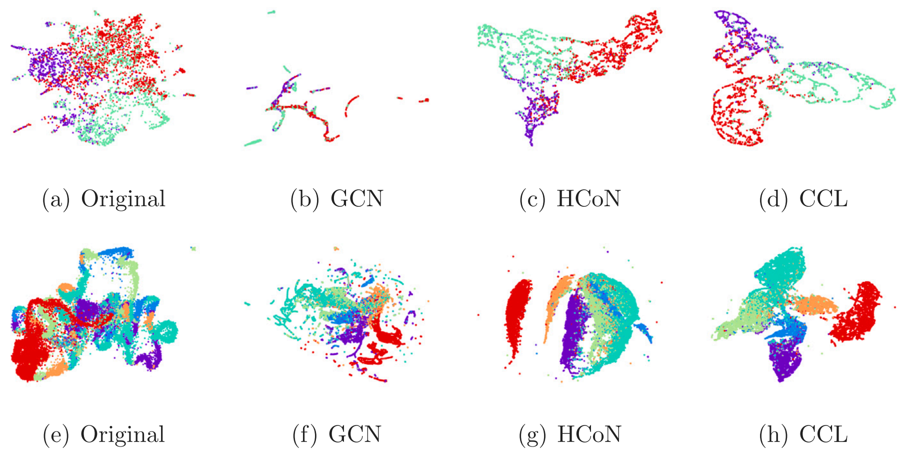


Fig. 6. Visualization of node samples represented by original features and different graph learning features on the Pubmed co-citation ((a)–(d)) and DBLP co-authorship ((e)–(h)) datasets. The three colors in (a)–(d) mean the three labels on the Pubmed co-citation dataset, and the six colors in (e)–(h) are the six labels on the DBLP co-authorship dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and 5(b), the overfitting issue occurs as the depth of the proposed model increases, which we plan to address in our future work so as to perform deep graph convolutional operations based on [49]. In addition, since CCL performs contrast in each layer, the efficiency may be low when working on large-scale datasets. To address this issue, we will consider conducting contrast based on prototypes, i.e., the central of samples with the same category. Furthermore, theoretical analyses like the generalization gap of the proposed model are not presented currently. As a result, we will work on the theoretical part to guarantee the performance of the proposed model.

Declaration of competing interest

All authors declare that No conflict of interest exists.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62206111, 62206061, 62106084, 61773179), the Young Talent Support Project of Guangzhou Association for Science and Technology (No. QT-2023-017), the Guangzhou Basic and Applied Basic Research Foundation (No. 2023A04J1058), the China Postdoctoral Science Foundation (No. 2022M721343, 2022M711331), the Outstanding Youth Project of Guangdong Natural Science Foundation of China (No. 2021B1515020076), the Natural Science Foundation of Guangdong (No. 2022A1515010468, 2023A1515012954), the Guangdong Provincial Key Laboratory of Traditional Chinese Medicine Informatization (2021B1212040007), the Science and Technology Planning Project of Guangdong Province (2023A0505050092), and the Fundamental Research Funds for Central Universities (No. 21622326). M. Ng's work was supported by Hong Kong Research Grant Council GRF 17201020, 17300021, CRF C7004-21GF and Joint NSFC-RGC N-HKU76921.

References

[1] X. Lin, C. Zhou, J. Wu, H. Yang, H. Wang, Y. Cao, B. Wang, Exploratory adversarial attacks on graph neural networks for semi-supervised node classification, Pattern Recognit. 133 (2023) 109042.
[2] L. Bai, L. Cui, Y. Jiao, L. Rossi, E.R. Hancock, Learning backtrackless aligned-spatial graph convolutional networks for graph classification, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2) (2022) 783–798.
[3] L. Bai, Y. Jiao, L. Cui, L. Rossi, Y. Wang, P.S. Yu, E.R. Hancock, Learning graph convolutional networks based on quantum vertex information propagation, IEEE Trans. Knowl. Data Eng. 35 (2) (2023) 1747–1760.
[4] S. Zhao, W. Wei, D. Zou, X. Mao, Multi-view intent disentangle graph networks for bundle recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, no. 4, 2022, pp. 4379–4387.
[5] D. Zou, W. Wei, X.-L. Mao, Z. Wang, M. Qiu, F. Zhu, X. Cao, Multi-level cross-view contrastive learning for knowledge-aware recommender system, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 1358–1368.
[6] H. Wu, Y. Yan, Y. Ye, M.K. Ng, Q. Wu, Geometric knowledge embedding for unsupervised domain adaptation, Knowl.-Based Syst. 191 (2020) 105155.
[7] D. Zhou, J. Huang, B. Schölkopf, Learning with hypergraphs: Clustering, classification, and embedding, Adv. Neural Inf. Process. Syst. 19 (2006).
[8] H. Wu, Y. Yan, M.K.-P. Ng, Hypergraph collaborative network on vertices and hyperedges, IEEE Trans. Pattern Anal. Mach. Intell. 45 (3) (2023) 3245–3258.
[9] S. Bai, F. Zhang, P.H. Torr, Hypergraph convolution and hypergraph attention, Pattern Recognit. 110 (2021) 107637.
[10] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du, C. Zou, Hypergraph learning: Methods and practices, IEEE Trans. Pattern Anal. Mach. Intell. 44 (5) (2022) 2548–2566.
[11] J. Hou, M. Pelillo, H. Yuan, Hypergraph matching via game-theoretic hypergraph clustering, Pattern Recognit. 125 (2022) 108526.
[12] Y. Feng, H. You, Z. Zhang, R. Ji, Y. Gao, Hypergraph neural networks, in: AAAI Conference on Artificial Intelligence, Vol. 33, no. 01, 2019, pp. 3558–3565.
[13] J. Hou, H. Yuan, M. Pelillo, Game-theoretic hypergraph matching with density enhancement, Pattern Recognit. 133 (2023) 109035.
[14] H. Wu, J. Long, N. Li, D. Yu, M.K. Ng, Adversarial auto-encoder domain adaptation for cold-start recommendation with positive and negative hypergraphs, ACM Trans. Inf. Syst. 41 (2) (2023) 1–25.
[15] H. Wu, C.W. Wong, J. Zhang, Y. Yan, D. Yu, J. Long, M.K. Ng, Cold-start next-item recommendation by user-item matching and auto-encoders, IEEE Trans. Serv. Comput. 16 (4) (2023) 2477–2489.
[16] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, P. Isola, What makes for good views for contrastive learning? Adv. Neural Inf. Process. Syst. 33 (2020) 6827–6839.
[17] L. Ericsson, H. Gouk, C.C. Loy, T.M. Hospedales, Self-supervised representation learning: Introduction, advances, and challenges, IEEE Signal Process. Mag. 39 (3) (2022) 42–62.
[18] Y. Wei, X. Wang, Q. Li, L. Nie, Y. Li, X. Li, T.-S. Chua, Contrastive learning for cold-start recommendation, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 5382–5390.
[19] U. Fang, J. Li, X. Lu, A. Mian, Z. Gu, Robust image clustering via context-aware contrastive graph learning, Pattern Recognit. (2023) 109340.
[20] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, Y. Shen, Graph contrastive learning with augmentations, Adv. Neural Inf. Process. Syst. 33 (2020) 5812–5823.
[21] J. Xia, L. Wu, G. Wang, J. Chen, S.Z. Li, Progcl: Rethinking hard negative mining in graph contrastive learning, in: International Conference on Machine Learning, PMLR, 2022, pp. 24332–24346.
[22] X. Xia, H. Yin, J. Yu, Q. Wang, L. Cui, X. Zhang, Self-supervised hypergraph convolutional networks for session-based recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, no. 5, 2021, pp. 4503–4511.
[23] L. Xia, C. Huang, Y. Xu, J. Zhao, D. Yin, J. Huang, Hypergraph contrastive collaborative filtering, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 70–79.

[24] D. Cai, M. Song, C. Sun, B. Zhang, S. Hong, H. Li, Hypergraph structure learning for hypergraph neural networks, in: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, 2022, pp. 1923–1929.
[25] Y. Dong, W. Sawin, Y. Bengio, HNHN: Hypergraph networks with hyperedge neurons, in: Graph Representations and beyond Workshop At International Conference on Machine Learning, 2020.
[26] H. Wu, M.K. Ng, Hypergraph convolution on nodes-hyperedges network for semi-supervised node classification, ACM Trans Knowl Discovery Data (TKDD) 16 (4) (2022) 1–19.
[27] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, 2017.
[28] D. Yang, B. Qu, J. Yang, P. Cudre-Mauroux, Revisiting user mobility and social relationships in lbsns: A hypergraph embedding approach, in: The World Wide Web Conference, 2019, pp. 2147–2157.
[29] J. Wang, K. Ding, L. Hong, H. Liu, J. Caverlee, Next-item recommendation with sequential hypergraphs, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1101–1110.
[30] H. Fan, F. Zhang, Y. Wei, Z. Li, C. Zou, Y. Gao, Q. Dai, Heterogeneous hypergraph variational autoencoder for link prediction, IEEE Trans. Pattern Anal. Mach. Intell. 44 (8) (2022) 4125–4138.
[31] T. Badamdorj, M. Rochan, Y. Wang, L. Cheng, Contrastive learning for unsupervised video highlight detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14042–14052.
[32] H. Cao, W. Yang, H.T. Ng, Grammatical error correction with contrastive learning in low error density domains, in: Findings of the Association for Computational Linguistics: EMNLP 2021, 2021, pp. 4867–4874.
[33] K. Hassani, A.H. Khasahmadi, Contrastive multi-view representation learning on graphs, in: International Conference on Machine Learning, PMLR, 2020, pp. 4116–4126.
[34] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, Q.V.H. Nguyen, Are graph augmentations necessary? simple graph contrastive learning for recommendation, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 1294–1303.
[35] D. Cai, C. Sun, M. Song, B. Zhang, S. Hong, H. Li, Hypergraph contrastive learning for electronic health records, in: Proceedings of the 2022 SIAM International Conference on Data Mining, SDM, SIAM, 2022, pp. 127–135.
[36] D. Lee, K. Shin, I'm me, we're us, and i'm us: Tri-directional contrastive learning on hypergraphs, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, no. 7, 2023, pp. 8456–8464.
[37] M. Gutmann, A. Hyvärinen, Noise-contrastive estimation: A new estimation principle for unnormalized statistical models, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 297–304.
[38] I. Bhattacharya, L. Getoor, Collective entity resolution in relational data, ACM Trans. Knowl. Discov. Data 1 (1) (2007) 5–es.
[39] G. Namata, B. London, L. Getoor, B. Huang, U. EDU, Query-driven active surveying for collective classification, in: International Workshop on Mining and Learning with Graphs, Vol. 8, 2012.
[40] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, Collective classification in network data, AI Mag. 29 (3) (2008) 93.
[41] R. Rossi, N. Ahmed, The network data repository with interactive graph analytics and visualization, in: AAAI Conference on Artificial Intelligence, Vol. 29, no. 1, 2015.
[42] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, P. Talukdar, HyperGCN: A new method of training graph convolutional networks on hypergraphs, in: Advances in Neural Information Processing Systems, 2019.
[43] J. Huang, J. Yang, UniGNN: A unified framework for graph and hypergraph neural networks, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, 2021, pp. 2563–2569.
[44] D.H. Wolpert, The lack of a priori distinctions between learning algorithms, Neural Comput. 8 (7) (1996) 1341–1390.
[45] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: International Conference on Learning Representations, 2018.
[46] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, A. Smola, A kernel two-sample test, J. Mach. Learn. Res. 13 (1) (2012) 723–773.
[47] L. McInnes, J. Healy, N. Saul, L. Grossberger, UMAP: Uniform manifold approximation and projection, J. Open Source Softw. 3 (29) (2018) 861.
[48] D. Arya, D.K. Gupta, S. Rudinac, M. Worring, Hypersage: Generalizing inductive representation learning on hypergraphs, 2020, arXiv preprint arXiv:2010.04558.
[49] K. Wang, G. Li, S. Wang, G. Zhang, K. Wang, Y. You, X. Peng, Y. Liang, Y. Wang, The snowflake hypothesis: Training deep GNN with one node one receptive field, 2023, arXiv preprint arXiv:2308.10051.

Hanrui Wu is currently an Associate Professor with the Department of Computer Science, Jinan University, Guangzhou, China. Before that, he was a Postdoctoral Research Fellow with the Department of Mathematics, The University of Hong Kong, from 2020 to 2021. He received the B.S. and Ph.D. degrees in the School of Software Engineering from South China University of Technology, China, in 2013 and 2020, respectively. His research interests include transfer learning, hypergraph learning, and their applications in recommendation systems and brain–computer interactions.

Nuosi Li is currently a graduate student with the College of Information Science and Technology, Jinan University, Guangzhou, China. His research interests are recommendation systems and graph convolution networks.

Jia Zhang received the Ph.D. degree from the Department of Artificial Intelligence, Xiamen University, Xiamen, China, in 2020. He is currently a Lecturer at the College of Information Science and Technology, Jinan University, Guangzhou, China. He is broadly interested in machine learning and data mining. He is currently working on multi-label learning, data fusion, feature selection, and weakly supervised learning.

Sentao Chen received the Ph.D. degree in Software Engineering from South China University of Technology, Guangzhou, China, in 2020. He is currently a Lecturer with the Department of Computer Science, Shantou University, Shantou, China. His research interests include statistical machine learning, domain adaptation, and domain generalization.

Michael K. Ng received the B.Sc. and M.Phil. degrees from the University of Hong Kong in 1990 and 1992, respectively, and the Ph.D. degree from the Chinese University of Hong Kong in 1995. He was a Research Fellow of Computer Sciences Laboratory at Australian National University from 1995 to 1997, and an Assistant/Associate Professor of the University of Hong Kong from 1997 to 2005. He was a Professor/Chair Professor in Department of Mathematics at Hong Kong Baptist University from 2006 to 2019. He was a Chair Professor in Research Division of Mathematical and Statistical Science at The University of Hong Kong from 2019 to 2023. He is currently a Chair Professor in Mathematics and a Chair Professor in Data Science at Hong Kong Baptist University. His research interests include bioinformatics, image processing, scientific computing, and data mining. He is selected for the 2017 Class of Fellows of the Society for Industrial and Applied Mathematics. He obtained the Feng Kang Prize for his significant contributions in scientific computing. He serves on the Editorial Board members of several international journals.

Jinyi Long received the Ph.D. degree at the South China University of Technology, Guangzhou, China. From 2012 to 2013, he worked with South China University of Technology as lecturer. From 2014 to 2016, he was with the Systems Neuroscience Institute, University of Pittsburgh and the Miami Project to Cure Paralysis Lois Pope Life Center, University of Miami, USA, as a Research Fellow. Since 2017, he has been with the Department of Computer Science, Jinan University, Guangzhou, China, as a full Professor. He works in the field of machine learning, brain signal processing, brain–computer interactions, and neural engineering.